

論 文

## 타임 페트리 넷를 이용한 비동기식 병행 시스템의 성능 평가

正會員 李 富 寧\* 正會員 宋 榮 宰\*

### Performance Evaluation of Asynchronous Concurrent System Using Timed Petri Nets

Boo Young LEE,\* Young Jae SONG\* *Regular Members*

**要 約** 지능 링크는 "전송과 처리"라는 두 가지 개념을 융합한 그림이며, 종래의 통신 회선이나 정보 전송뿐만 아니라 처리를 실시할 수 있는 기능을 가진 새로운 전송 처리 시스템이다. 이 지능 링크는 각 세그먼트의 기능을 소프트웨어에 의하여 주어지는 범용 파이프라인 프로세싱 시스템이라고 간주할 수 있으며, 각 세그먼트에 대한 처리시간이 다르므로 비동기적 파이프라인으로 동작한다. 본 논문은 도형 표현에 의한 시각적 이해가 용이하고 실행 규칙이 단순한 페트리 넷를 이용하여 지능링크와 파이프라인을 모델화하였으며, 계산기상에서 시뮬레이션하는 것으로서 그 성능을 평가하였다.

**ABSTRACT** Generally, Intelligent link is made of drawing picture that is amalgated in two conception of communication and processing. This is a new system of information transmission having fuction being able to handle even though the former information trasmission can not be possible. That Intelligent link is to say that function of each segment by working software can be available at general purpose pipeline processing system, general purpose pipeline have a different working time in function of segment, work asynchronizing pipeline. in the papers, graphic design is easy to understand the sense of sight model by Petri Net that is simple regulation. It can translate into working of asynchronizing pipeline by working of calculator of simulation.

#### I. 서 론

최근 소프트웨어와 하드웨어의 공용, 시스템의 신뢰성 향상등을 목적으로한 컴퓨터 네트워크의

구축이 여러 가지로 시험하고 있다. 이런 동작중에서 컴퓨터간의 통신 회선으로 정보 전송 처리를 할 수 있는 새로운 전송 처리 시스템으로써 지능 링크가 제안되고 있다.<sup>(1)</sup>

오늘날, 통신 선로는 금속 회선으로 구성되어 그 기능은 단지 데이터의 전송을 위한 것이다. 그러나, 소프트웨어에 의해 공급된 기능을 가진 지능링크는 데이터 전송과 처리를 하기 위해 제한된 것이다.<sup>(2)</sup> 그러므로 그 기능은 데이터 처리의

\*慶熙大學校 工科大學 電子工學科  
Dept. of Electronics Eng. Kung Hee University Seoul,  
131-701, Korea.  
論文番號 : 89-49(接受. 6. 7)

내용에 따라서 변화할 수 있다.

컴퓨터 네트워크의 발전은 컴퓨터 기술과 통신 시스템, 특히 컴퓨터 통신의 배후에 추진력이 되는 경향이 있다.<sup>(8-9)</sup> 그러한 단계는 처리 능력에 따라서 물리적 제약으로 제한되는데 도달되고 있다. 그러므로 컴퓨터 네트워크 설계의 점진적 진보는 더욱더 중요하게 된다.

따라서 이 지능 링크는 각 세그먼트의 기능을 소프트웨어에 의하여 주어진 범용 파이프라인 처리 시스템을 가정할 수 있으며, 각 세그먼트에 대한 처리 시간이 다르며 비동기적 파이프라인으로 동작한다.

한편, 페트리 넷은 도형 표현에 의한 시각적 이해가 용이하고, 해석수단의 명세화, 계층적 기술의 가능성에 의하여 최근에 들어 비동기 시스템을 표현하는 우수한 방법으로 주목되고 있다.

더우기 페트리 넷에 시간의 개념을 도입한 타임 페트리 넷이 있다. 통신 제어 순서에 있어서 시간 페트리 넷 시간의 설정 문제등이 이용되고 있다. 본 논문에서는 타임 페트리 넷에 의한 모델화로서 그 동작을 계산기상에서 시뮬레이션하는 것으로 지능 링크의 비동기 파이프라인의 동작을 해석했다. 특히 처리 시간의 방법과 세그먼트수가 처리율, 지연시간에 주는 영향 및 세그먼트간에 버퍼(buffer)를 도입했을 경우 그것에 의한 변화를 검토한다.

## II. 지능 링크와 페트리 넷

### II-1. 지능 링크

지능 링크는 호스트 계산기 A,B간에 전송로와 데이터의 처리를 행하는 프로세서(Processor)가 프로세스 요구를 일 차원인 세그먼트로 나란히 구성한 것이다. 임의의 데이터는 파이프라인 처리로부터 전송되고 있지만, 그때 각 데이터의 처리는 데이터에 관련되어 공급되는 프로그램에 의해 처리되어진다. 지능 링크는 소프트웨어에 의하여 기능을 공급하는 범용 파이프라인 시스템이다.

이와같이, 지능 링크는 각 세그먼트의 기능이 고정되어 있는 것은 아니고 소프트웨어에 의하여

처리를 주어진다. 각 세그먼트의 처리 시간은 고정되어 있지 않으며, 비동기형 파이프라인으로 동작하는 것이다.

그것을 위하여, 그림 1에 도시한 세그먼트 사이에 블록킹(blocking)이 생기므로, 시스템의 성능이 저하되는 것이 예상된다. 그림 1에서는 각 세그먼트에서 처리와 전송이 병렬로 처리되어지고 있지만, 다음의 세그먼트에서 전송이 완료되고 있지 않다. 앞의 세그먼트에서 전송이 완료되고 있지 않는 등의 원인 때문에 프로세서의 휴무 상태가 사선부분으로 나타낸다.

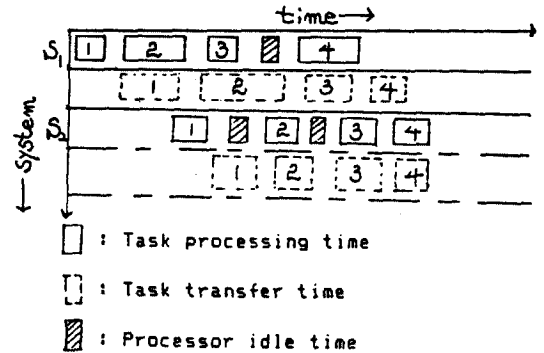


그림 1. 지능 링크의 타스크 천이  
Transition of tasks within an intelligent link.

이러한 프로세서의 휴무시간은 세그먼트수가 증가함으로써 증가함을 이것에 의한 처리율이 향상됨을 예상할 수 있다. 지능 링크의 비동기 파이프라인으로 이러한 동작을, 다음장 이후에 타임 페트리 넷을 이용하여 해석하고 있다.

### II-2. 페트리 넷(Petri Net)

페트리 넷은 1962년에 Petri에 의하여 사용된 이래 정보의 흐름을 추상적, 형식적 모델로서 알려지고 있다.<sup>(9)</sup> 최근에는, 비동기 시스템을 표현하는 우수한 방법으로 주목되고, 각 방면에 대한 응용이 되고 있다.

페트리 넷은 다음 세개의 집합으로 구성되어 있다.

$n=(P,T,A)$

- 여기서 P: 유한 개의 장소 집합
- T: 유한 개의 천이 집합
- A: 유한 개의 아크(arc) 집합

각 아크는 천이에서 장소로 또는 장소에서 천이 방향으로 도시할 때는 장소는 ○으로, 천이는 | (봉)으로, 아크는(→)로 표시한다. 더우기 장소에 토큰(Token)이라 불리는 표적물을 주므로써 페트리 넷에 표시된 시스템의 상태를 표시하고 있는 것이다. 장소에 토큰을 주어진 것을 마킹(Marking)이라 부르며, 마킹된 페트리 넷을 마크된(Marked) 페트리 넷이라고 한다. 만일 각 입력 장소가 적어도 각 장소에서 천이에 이르는 아크의 수 만큼 토큰을 포함한다면 천이는 점화가능(enable) 상태가 된다.

천이의 점화를 가능하게 하는 입력 장소에서의

토큰을 천이의 점화 가능 토큰이라 한다. 점화가능 천이는 점화될 수 있다. 천이의 점화는 입력장소로부터 모든 천이 점화 가능 토큰을 제거하고 천이로부터 각 아크를 통하여 토큰을 이동하여 출력 장소에 도달하게 한다. 천이의 점화로 장소에서 토큰들을 제거하고 할당함으로써, 넷의 상태가 변화되어 새로운 상태로 된다. 한 페트리 넷의 도달가능 집합과 초기 상태가 주어지면 초기 상태에서 천이의 연속적인 점화에 의해 모든 상태에 도달가능하게 된다. 도달가능 집합에 의한 도달가능 그래프는 다음과 같이 구성될 수 있다. 만일 상태 S1 이 상태 S2로 어떤 천이의 점화에 의해 발생한다면 넷의 각 상태는 노드 S1으로 부터 노드 S2까지 방향을 가진 간선과 노드로 표시된다.

이와 같은 페트리 넷의 동작은 그림 2에서 살펴보면,

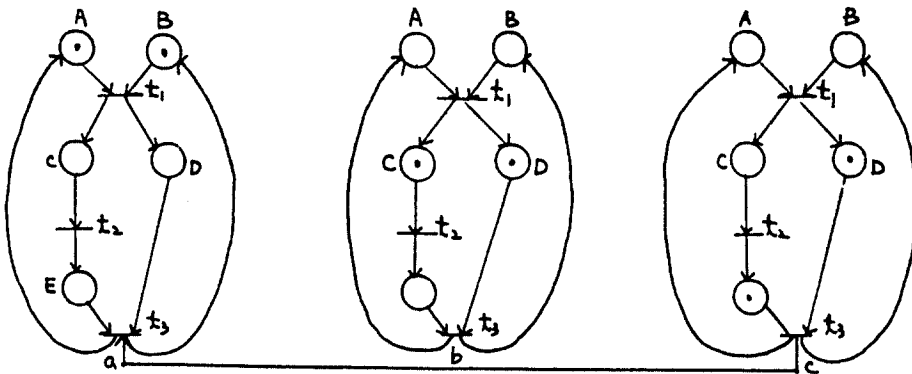


그림 2. 페트리 넷의 동작  
Execution of Petri net

그림 2의 a에서 천이  $t_1$ 은 입력 장소 A,B 양쪽에 하나의 토큰을 가지고 있기 때문에 동작 가능하다. 천이  $t_1$ 이 점화하면 장소 A,B로부터 하나의 토큰은 제거되고, 각 출력 장소 C,D에 하나의 토큰을 배열한다.

이런 점에서 천이  $t_3$ 는 입력 장소중 하나인 E에 토큰이 없기때문에 동작 불가능한 상태이다. 그러나, 천이  $t_2$ 는 동작가능하다.  $t_2$ 가 점화하므로

장소 C로부터 하나의 토큰을 제거하고 장소 E에 하나의 토큰을 배치한다. 천이  $t_3$ 가 동작가능하면 천이  $t_3$ 가 점화되어 장소 D와 E에서 하나의 토큰을 A와 B에 배치하여 시스템은 초기 배열로 되돌아간다.

### II -3, 타임 페트리 넷(Time Petri Net)

페트리 넷은 시간에 관련된 기술을 내포하고

있지 않다. 그러나 일반적으로, 시스템을 모델화 할 때는 시간의 개념을 도입한 것으로 Merlin에 의하여 제안된 타임 페트리 넷이 있다. 타임 페트리 넷은 페트리 넷의 각 천이(\$t\_i\$)에 점화 시간 허용 범위[\$\tau\_i^\*\$, \$\tau\_i^{\*\*}\$]을 주는 것으로써 천이 \$t\$는 토큰을 갖추므로써 \$\tau\_i^\*\$을 경과하면 점화 가능한 것이다. 즉시 토큰이 갖추어지면 지연과 함께 \$\tau\_i^{\*\*}\$을 경과함으로써 점화 되어야 한다. 그리고 이 점화는 순간적으로 실행된다. 타임 페트리 넷은 그 점화 시간 허용 범위의 차이로써 점화의 우선권을 표시하므로써 얻는다. 점화 시간 허용 범위를 갖는 경우에는 자유 선택 장소에서 분기에 의한 버스(bus)의 변경이라는 형태로 결과적으로 똑같은 형태로 실현될 수 있다.

여기서 천이 \$t\_i\$의 사이클 타임 \$C\_i\$를 다음과 같이 정의 한다.

$$C_i = \lim_{n_i \rightarrow \infty} \frac{S(n_i)}{n_i}$$

단, \$S(n\_i)\$는 \$t\_i\$의 \$n\_i\$회 반복하는 점화 시간

각각의 장소에 한 개의 입력 아크와 한 개의 출력 아크만이 존재하고, 더구나 밀접하게 연결된 그래프로 표시되고 있는 타임 페트리 넷의 각 천이의 사이클 타임은 모두 똑같은 상태로 다음과 같이 된다.

$$C = \max \left\{ \frac{T_k}{N_k} : K=1,2,\dots,q \right\}$$

단 \$T\_k = \sum f(t\_i)\$: 계로 \$L\_k\$내의 천이의 합계 점화 시간

\$N\_k = \sum M(P\_i)\$: 계로 \$L\_k\$내의 장소의 합계 토큰 수

\$q\$: 넷내의 장소 수

\$M(P\_i)\$: 장소 \$p\$의 초기 토큰 수

그러나 이상의 결과는 적용된 시스템이 어느정도 제한된다. 그래서 이하에서는 일반적으로 안정한 타임 페트리 넷을 해석하기 위해서 타임 페트리 넷의 시점 \$A\$에서 시점 표시 (ID: Instantaneous Description)d를 다음과 같이 두개의 함수 조로 정의된다.

$$d_a = (m_a, w_a)$$

단, \$m\$: 장소에 토큰의 유무를 대응하는 마크(marking)의 함수

\$w\$: 천이에서 점화될 때 부터 대기하는 최소의 대기 시간을 대응하는 점화 대기 시간 함수

$$0 \leq w_a(t) \leq f(t)$$

천이 \$t\$는 점화될 때 부터 최저 \$w(t)\$를 대기해야만 한다. \$w(t)\$를 경과할 때까지 어디에서 점화될 것인가는 제한되며 그 사이에 연속적으로 토큰을 갖추고 있는 경우에 볼 수 있는 점화이다. 이하의 것이 성립할 경우는 \$d\_a = (m\_a, w\_a)\$에서 \$d\_b = (m\_b, w\_b)\$로 직접 도달가능하게 된다.

(1) \$\forall p \in P\$

$$m_b(p) = m_a(p) + \sum_{t \in \text{In}(p)} e_a(t) - \sum_{t \in \text{Out}(p)} e_a(t)$$

(2) \$\forall t \in T\$

$$w_b(t) = \begin{cases} f(t) & t \in S(m_a) \cup S(m_b) \cup S'(m_a) \\ w_a(t) - T_a & \end{cases}$$

단

$$T = \min_{t \in S(m_a)} w_a(t)$$

\$S(m\_a) = \{ m\_a \text{에 토큰이 갖추어진 천이} \}\$

\$S(m\_a) = \{ S(m\_a) \ni t, w\_a(t) = T\_a \text{인 } t \}\$

단 충돌이 있는 경우는 그 내에 한 개

$$e_a(t) = \begin{cases} 0 & t \in S'(m_a) \\ 1 & t \in S(m_a) \end{cases}$$

\$\text{In}(p) = p\$의 입력 천이, \$\text{Out}(p) = p\$의 출력 천이  
다음 장에서 이 타임 페트리 넷에 의한 지능 링크를 모델화한다.

### III. 타임 페트리 넷에 의한 지능 링크의 모델화

#### III-1. 버퍼가 없는 경우

두개의 세그먼트로 구성된 지능 링크의 타임 페트리 넷 모델은 그림 3에 표시했다.

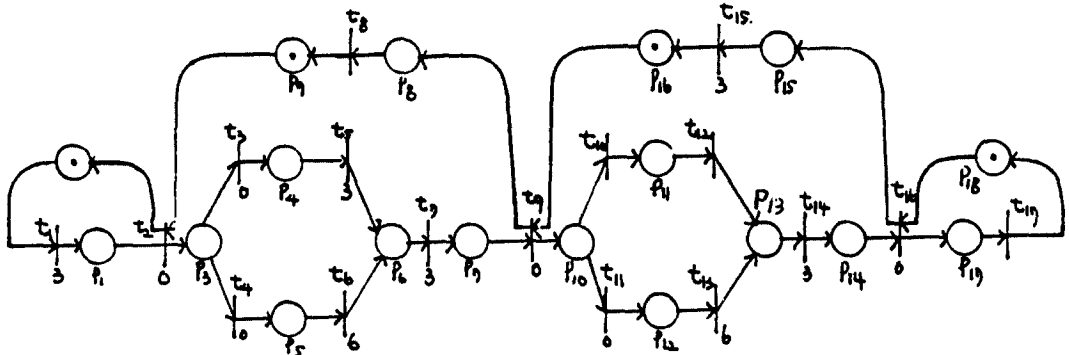


그림 3. 버퍼가 없는 지능 링크의 타임 페트리 넷 모델  
Time Petri net model of the intelligent link without buffers.

각 천이는 다음과 같은 의미를 가지고 있다.

- $t_1$ : 호스트 계산기의 전송
- $t_2, t_9, t_{16}$ : 전송 완료
- $t_3, t_4, t_{10}, t_{11}$ : 처리 개시
- $t_5, t_6, t_{12}, t_{13}$ : 처리 완료
- $t_7, t_{14}$ : 송신.  $t_8, t_{15}$ : 수신
- $t_{17}$ : 호스트 계산기의 수신

- $D_6, D_{13}$ : 송신 준비
- $D_7, D_{14}$ : 송신 중
- $D_8, D_{15}$ : 수신 준비
- $D_9, D_{16}$ : 수신 중
- $D_{17}$ : 호스트 계산기의 수신 준비
- $D_{18}$ : 호스트 계산기의 수신 중

$t_2$ 에서  $t_9$ ,  $t_9$ 에서  $t_{16}$ 까지가 각각 지능 링크의 한 세그먼트에 대응한 것이다.

각 세그먼트는 수신, 처리, 송신으로 한개의 사이클(cycle)를 형성하며, 그것을 병렬로 처리할 수 없다. 수신, 송신은 각각 앞의 세그먼트 송신, 다음의 세그먼트 수신이 타임(timing)을 취하여 종료하고, 그것을  $t_2, t_9, t_{16}$ 의 전송 종료에 의해 실현되고 있다. 처리는 매우 다른 두 종류가 있는데, 그 중 어느 한쪽 방향이 행해지고 있다. 또한, 각 천이의 점화 대기 시간은 그림 중 각 천이의 옆 숫자로 표시하였고, 송 수신과, 처리에 걸리는 시간이 거의 똑같다.

즉, 각 장소는 다음과 같은 상태로 대응한다.

- $P_1$ : 호스트 계산기의 송신 중
- $P_2$ : 호스트 계산기의 송신 준비
- $P_3, P_{10}$ : 처리 대기
- $P_4, P_5, P_{11}, P_{12}$ : 처리 중

세그먼트 수가 셋 이상인 경우는  $t_2$ 에서  $t_9$ 까지 전부 같은 천이 및 장소에 대응하는 수치만 직선으로 나열했다.

### III-2. 버퍼가 있는 경우

각 세그먼트의 처리 시간의 분산에 의한 성능 저하를 피하기 위한 방법으로써 세그먼트 사이에 버퍼를 설치해 처리 시간의 분산을 흡수시키는 것을 고려했다. 여기서, 한 개의 버퍼는 일반적으로 복수개의 대기실까지 설치하여, 그 수를 버퍼 사이즈(size)라고 한다. 버퍼 사이즈가 일인 버퍼를 수신전에 한 개 설치한 시스템을 고려, 그림 3과 같은 두 개의 세그먼트까지 성립된 지능 링크의 타임 페트리 넷 모델을 그림 4에 표시했다. 그림 3과 비교하면, 새로이 추가된 천이 및 장소는 다음과 같은 의미를 가지고 있다.

- $t_{18}, t_{19}$ : 송신 종료,  $P_{19}, P_{21}$ : 버퍼 완충(buffer full)
- $P_{20}, P_{22}$ : 버퍼 공 (buffer empty)

초기 상태에서 버퍼는 제로이고,  $P_{19}, t_2, P_{20}, t_1$

에 루프를 형성한 것으로 한 단위의 프로그램+데이터의 내용을 격납하는 것을 보증하고 있다.

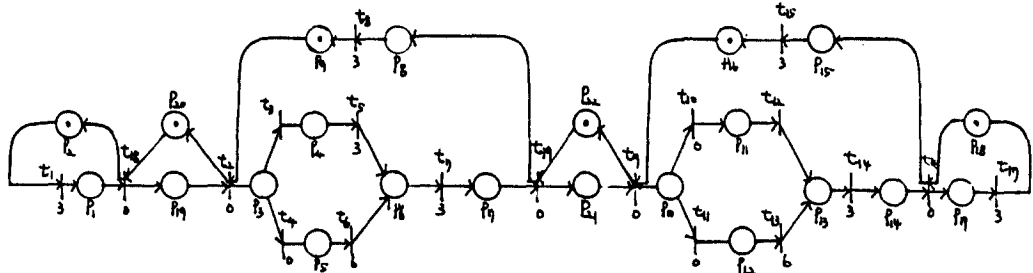


그림 4. 버퍼를 가진 지능 링크의 타임 페트리 넷 모델  
Time Petri net model of the intelligent link with buffers.

#### IV. 지능 링크의 분석 결과

##### IV-1. 도달가능 시점표시에 의한 분석

지능 링크의 사이클 타임과 지연 시간을 다음과 같이 정의한다.

(1) 각 세그먼트에 한 단위의 프로그램+데이터를 수신, 처리, 송신하기 위해 필요한 시간의 평균을 사이클 타임이라 한다. 이것은 자유 선택 장소에서 분기된 버스 이외에서는 II.3에 정의한 사인클 타임과 같다.

(2) 한 단위 프로그램+데이터가 지능 링크의 전체 세그먼트를 통과하는데 필요한 시간의 평균을 지연 시간이라 한다.

단, 버퍼가 있는 경우는 제일 세그먼트의 앞에 버퍼를 고려하지 않은 것이다.

더우기 세그먼트 수가 다른 경우의 지연 시간을 비교하기 위해 지연 시간을 할당해 한 세그먼트에 적합한 지연 시간을 고려하는 것도 있다.

두 세그먼트에 버퍼가 없는 경우 도달가능 시점표시의 그래프 표현을 그림 5에 주었다. 각 시점표시는 간단한 숫자로 대응했다. 이 그래프에서 분리된 것을 보면, 그림 3의 시스템은 초기 상태(노드 0)에서 출발하여 최종적으로는 노드 5, 노드 6을 통과하는 루프를 회전하는 정상 상태로 되고, 끝으로 정상 상태에서 사이클 타임은 그 루프를 회전하기 위한 평균 시간을 구하면,

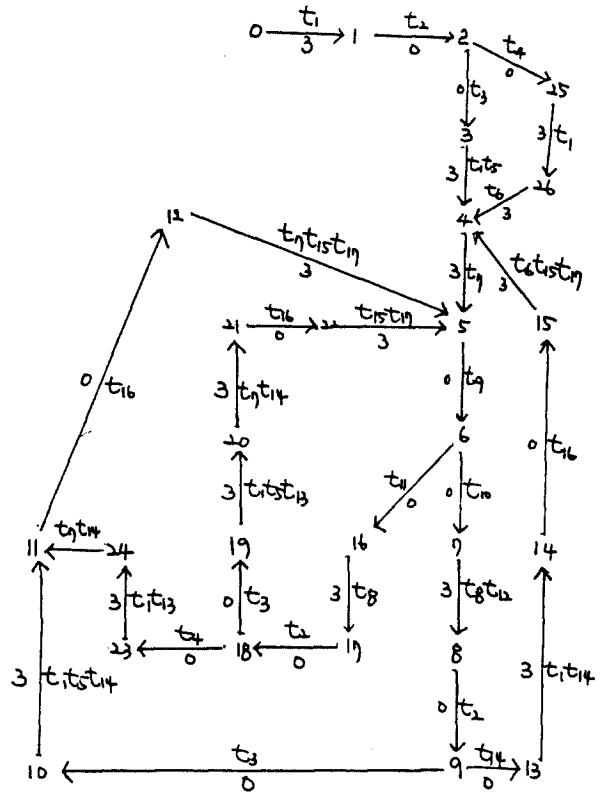


그림 5. 그림 3의 도달가능 시점 표시의 그래프  
Graph of reachable ID of Fig.3.

$9+3(p(t_{11})+p(t_4)p(t_{10}))$  로 된다.

단,  $p(t_{11}), +p(t_{10}), p(t_4)$ 는 각각  $t_{11}, t_{10}, t_4$ 의 발생 확률이다.

또, 제일 세그먼트로 수신을 완료할 때 부터 제일 세그먼트로 송신을 완료한 시간, 즉 지연 시간은  $12+3(2p(t_{11})+p(t_4) p(t_{10}))$ 이다.

#### IV-2. 시뮬레이션 결과

두개의 세그먼트에 버퍼가 없는 경우는 도달가능 시점표시까지 사이클 타임, 지연 시간등이 용이하게 나타났지만 타임 페트리 넷가 매우 복잡하기 때문에 해석은 간단하지 않다. 시뮬레이션은 타임 페트리 넷를 직접 실행한 것으로 즉시 점화가능한 천이를 차례 차례로 점화된 것을 기본으로 한다. 즉, 초기 마킹(marking)과 점화 대기 함수에서 II.3에 표시된 직접 도달가능한 시점표시가 몇개 있는 경우에는 미리주어진 확률로써 그 안에서 한개씩 증가한다. 타임 페트리 넷의 점화

시간 허용 범위의 상한과 하한등이 있는 제한을 가지고 있고, 계산기에 의한 시뮬레이션이 쉽게 되고 있다. 이 시뮬레이션은 IBM-PC XT 컴퓨터를 사용했고, 그래프는 LOTUS 1-2-3를 사용했다.

#### IV-2.1 버퍼가 없는 경우

$t_9, t_{10}$ 등의 발생 확률의 변화에 의한 사이클 타임의 변화를 그림 6에 표시했다. 각 세그먼트에 대한

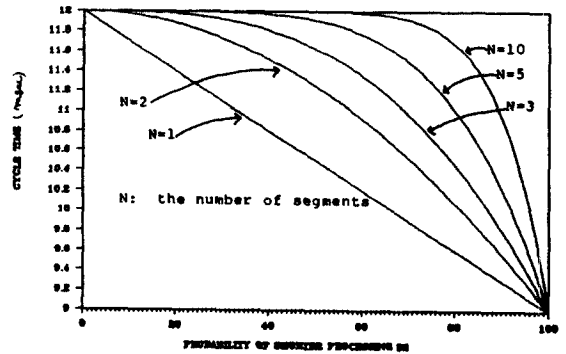


그림 6. 버퍼가 없는 지능 링크의 사이클 타임  
Cycle time of the intelligent link without buffer.

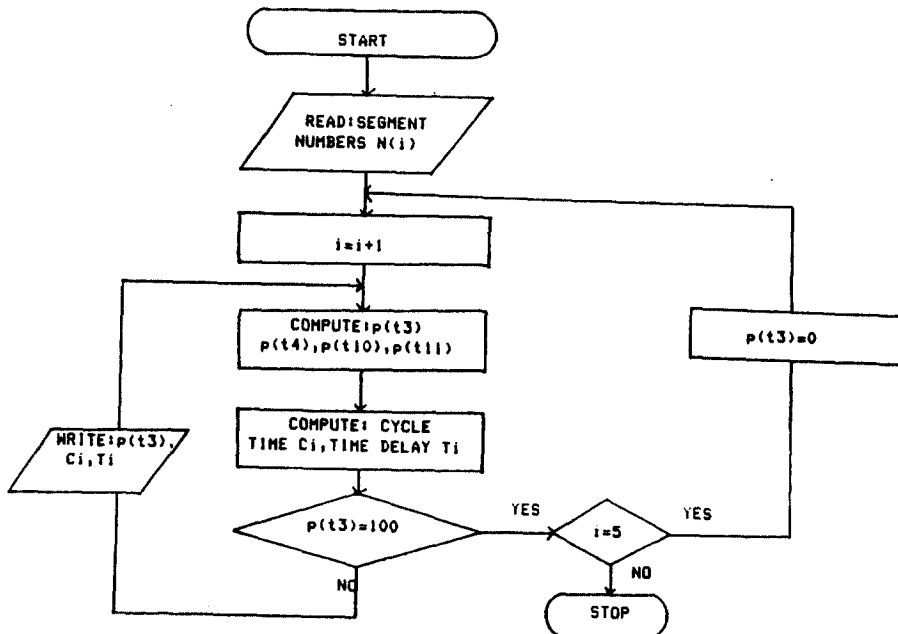


그림 7. 버퍼가 없는 경우의 흐름도  
Flowchart without buffers.

처리의 발생 확률은 똑같으므로, 예를 들면 제일 세그먼트에 대한 짧은 처리는  $t_3$ 의 발생 확률이 30% 정도이며, 제이 세그먼트의  $t_{10}$ 의 발생 확률도 30%가 되고, 제삼, 제사 세그먼트 이상에서도 똑같은 것이다. 처리가 긴것은 시간이 6, 짧은 것은 3, 송신, 수신 공히 3이 되고, 사이클 타임의 최장은 12, 최단은 9가 되고, 짧은 처리의 발생 확률이 크게되면 당연히 작게 된다. 그러나, 세그먼트 수  $N$ 가 증가하고 다음의 세그먼트 이상의 처리가 길게되어 블록킹(blocking) 과 앞단의 세그먼트 처리가 길게되어 휴무 상태가 보존되어 사이클 타임이 길게 된다.

한 세그먼트 당 지연 시간이 최장 9, 최단이 6이된다. 그러나 지연 시간에서는 앞의 세그먼트 처리가 길게 되기 때문에 휴무 상태로 되는 경향을 표시하고 있다. 사이클 타임과 비교해보면 세그

먼트 수  $N$ 에 따라 변화는 적게 된다. 이 변화를 그림 8에 표시했고, 버퍼가 없는 경우에 대한 흐름도(flowchart)는 그림 7에 표시했다.

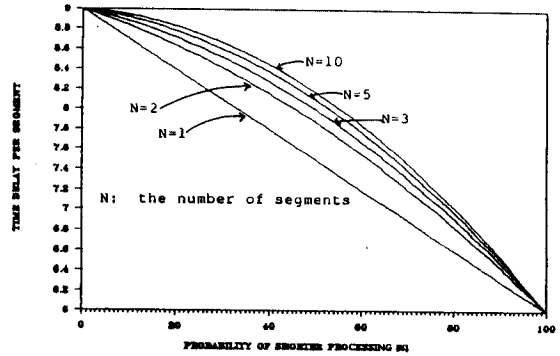


그림 8. 버퍼가 없는 지능 링크의 지연 시간  
Time delay per segment of the intelligent link without buffer.

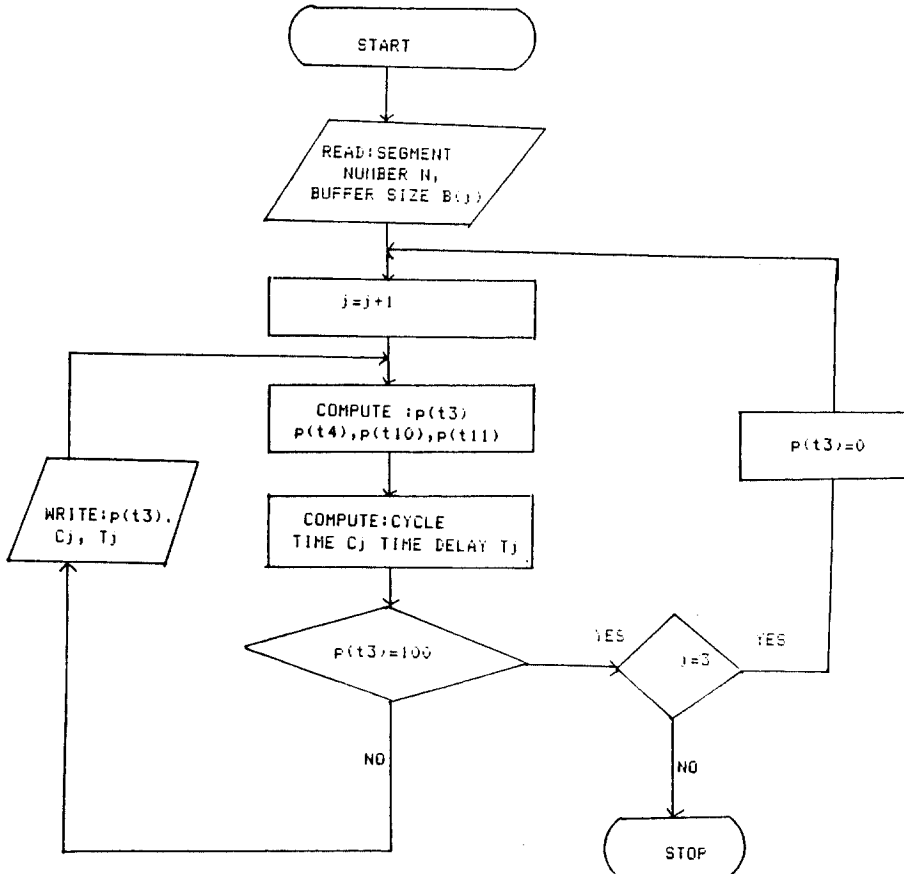


그림 9. 버퍼가 있는 경우의 흐름도.  
flowchart with buffers.



#### IV-2.2 버퍼가 있는 경우

세그먼트 사이에 버퍼를 삽입하였을 경우를 조사해보자. 세그먼트 수가 5인 경우 각 세그먼트 앞의 버퍼 크기를 변화시켜 얻은 시뮬레이션의 결과를 그림 10, 그림 11에 표시했다. 그림 10은 버퍼를 삽입하므로써 세그먼트에 대한 처리 시간의 분산을 흡수하는 정도를 나타내고, 사이클 타임이 감소되는 양상을 나타내고 있다. 양단을 연결한 직선이 프로세서가 휴무 상태가 아닌 상태로 대응하고 있으며, 버퍼를 삽입하므로써 그 직선에 근접하는 모양을 나타내고 있다. 그러나 버퍼를 삽입한 것에 대한 지연 시간은 크게 증가한다 (그림 11). 이 경우의 지연 시간이란 한 단위의 프로그램 + 데이터가 제일 세그먼트로 수신되는 것부터 제 5 세그먼트 송신이 완료되기까지의 시간이다. 이 지연 시간은 버퍼 크기가 2인 경우는 버퍼가

없는 경우의 두 배 가까이 된다. 버퍼가 있는 경우의 흐름도 (flowchart)가 그림 9에 있다.

### V. 결 론

본 논문에서는 타임 페트리 넷을 이용해 비동기식 병행 시스템의 성능 평가를 하기 위해 지능 링크를 예를 들어서 실시했다.

지능 링크에 대해서는 세그먼트의 처리 시간을 다르게 하므로써, 시스템의 성능이 저하한다. 그 저하의 양상 및 세그먼트간에 버퍼를 삽입한 경우 사이클 타임과 시스템 성능등의 개선을 타임 페트리 넷을 이용해 시뮬레이션으로 분석했다.

그것에 의하면, 세그먼트간에 버퍼를 삽입하므로써, 사이클 타임이 개선되고 시스템의 성능이 향상되고, 비동기형 파이프라인에서의 버퍼효과는 매우 크다.

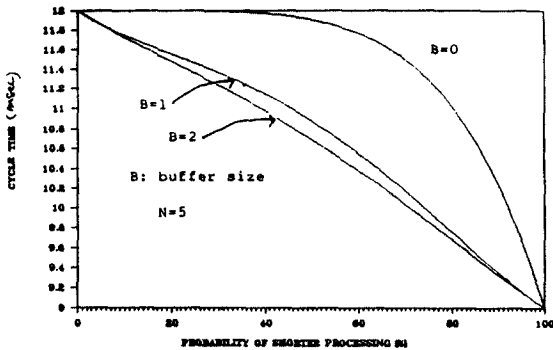


그림 10. 5개 세그먼트로 구성된 지능 링크의 사이클 타임  
Cycle time of the intelligent link composed of five segments.

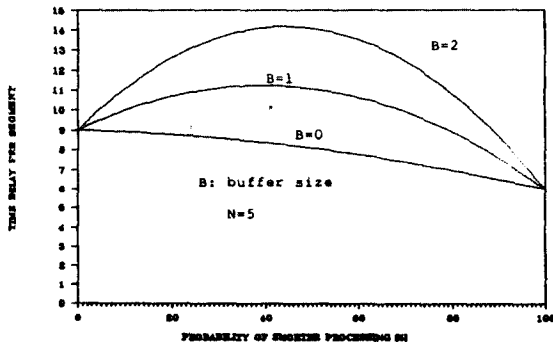


그림 11. 5개의 세그먼트로 구성된 지능 링크의 지연 시간  
Time delay per segment of the intelligent link composed of segments.

### 참 고 문 헌

1. T.Agerwala, "Putting Petri nets to work", IEEE Computer, pp.85-94. Dec.1979.
2. Tadao Nakamura and Poshiharu Shigei, "An Effect of Employing Intelligent Links into Data Networks", p50.4.150.4.5
3. H.Frank and W.Chou, "Topological optimization of Computer networks", Pro.IEEE, Vol.60, pp.1385-1397, NOV.1972.
4. N.Abramson, "The throughput of packet broadcasting channels", IEEE Trans. Commun., Vol. COM-25, pp. 117-128, Jan.1977.
5. L.Kleinrock and S.S.Lam, "Packet switching in a multiaccess broadcast channel : Performance evaluation", IEEE Trans. Commun., Vol.COM-23, pp.410-423, April 1975.
6. J.L. Peterson, Petri net theory and Modelling of systems", Prentice Hall, 1981.

7. C.Ramchandani, "Analysis of asynchronous concurrent systems by Timed petri nets" Ph.D.Thesis, MIT 1974 , Project MAC Report MAC-TR-120
8. W.M.Zuberek, "Timed Petri nets and Preliminary performance evaluation",m 7TH Annual Symposium on Computer Architecture, pp.88-96. 1980.
9. P.M.Merlin, "A Methodology for the Design and Imp-

- limentation of Communication Protocols", IEEE Trans. Commun., COM-24.6, pp.614-621 (June 1976)
10. C.V. Ramamoorthy and S.Gray HO, "Performance Evaluation of Asynchronous Concurrent System Using Petri Nets", IEEE Trans. Software Eng., SE-6.5, pp. 440-449 (Sept. 1980)



李 富 寧 (Boo Young LEE) 正會員  
1949年 1月 3日生  
1973年 2月：漢陽大學校 卒業  
1981年 8月：慶熙大學校 大學院 電子工  
學科 卒業(工學碩士)  
1987年 2月：慶熙大學校 大學院 電子工  
學科 博士課程 修了  
1985年～現在：鐵道專門大 講師  
1988年～現在：慶熙大學校 講師

• 關心分野：Computer network, operating system,  
Data Base System.



宋 榮 宰 (Young Jae SONG) 正會員  
1947年 4月 20日生  
1969年 2月：仁荷大學校 電子工學科 卒  
業  
1972年 10月：日本 Toyo Seiko 研究員  
1976年 3月：日本 Keio Univ. 大學院 卒  
業  
1979年 8月：明知大學校 卒業(工學博士)  
1980年 1月：工業振興庁 工業標準 審議  
委員

1982年 8月：美國 Univ. of Maryland 客員教授  
1985年 1月：韓國情報科學會 平委員  
1986年 1月：大韓電子工學會 電子計算研究會 專門委員長  
1986年 1月～現在：IEEE Computer Society 한국지회 副會長  
1987年 6月～現在：全國大學電算所長協議會 總務理事, 副會長  
1976年～現在：慶熙大學校 電子計算工學科 教授  
• 關心分野：소프트웨어 엔지니어링, 데이터베이스 시스템  
Object-oriented Programming & Systems.