

SAN을 이용한 제한된 버퍼 크기를 갖는 출력큐잉 ATM 스위치 성능평가

장 경 수[†] · 신 호 진[†] · 신 동 렬^{††}

요 약

네트워크 연결을 위한 고속 스위치는 계속해서 발달하여 왔으며, 스위치가 필요한 성능을 내는가를 여러 조건에서 분석하는 것은 중요한 일이다. 하지만, 복잡한 구조를 가진 시스템을 모델링하여 그 성능을 측정하는 것은 쉬운 일이 아니다. 큐잉이론을 이용한 모델링은 큰 상태 공간을 고려해야 됨은 물론이고 성능평가에 있어서도 복잡한 계산과정을 수행해야 하지만, SAN(Stochastic Activity Networks)에 의한 모델링과 성능평가는 그에 비해 간단하다는 장점이 있다.

본 논문의 목적은 출력포트에 큐를 갖는 고속 ATM 스위치를 확장된 SPN(Stochastic Petri Net)인 SAN을 이용해 모델링하고, 셀 도착 과정은 실제 트래픽과 유사한 특징을 가지고 있는 MMPP(Markov Modulated Poisson Process)로 모델링하여 그 성능을 평가하는데 있다. MMPP 모델을 이용한 버스티 트래픽을 고려한 성능측정과 아울러 SAN의 장점을 이용한 확장이 용이한 스위치 모델을 보이고자 한다.

제한된 버퍼 크기를 갖는 출력 큐잉 ATM 스위치에 도착하는 셀은 포아송 도착 과정에서는 정확히 표현할 수 없는 버스티 특징을 표현할 수 있어 좀더 실제 트래픽에 가까운 MMPP로 모델링한다. SAN 모델은 UltraSAN 소프트웨어 패키지를 이용해 대기행렬의 크기, 지연시간 그리고, 셀 손실률에 대한 성능을 측정한다.

Performance Evaluation of Output Queueing ATM Switch with Finite Buffer Using Stochastic Activity Networks

Kyung-Soo Jang[†] · Ho-Jin Shin[†] · Dong-Ryeol Shin^{††}

ABSTRACT

High speed switches have been developing to interconnect a large number of nodes. It is important to analyze the switch performance under various conditions to satisfy the requirements. Queueing analysis, in general, has the intrinsic problem of large state space dimension and complex computation. In fact, The petri net is a graphical and mathematical model. It is suitable for various applications, in particular, manufacturing systems. It can deal with parallelism, concurrence, deadlock avoidance, and asynchronism. Currently it has been applied to the performance of computer networks and protocol verifications. This paper presents a framework for modeling and analyzing ATM switch using stochastic activity networks (SANs).

In this paper, we provide the ATM switch model using SANs to extend easily and an approximate analysis method to apply ATM switch models, which significantly reduce the complexity of the model solution.

Cell arrival process in output-buffered queueing ATM switch with finite buffer is modeled as Markov Modulated Poisson Process (MMPP), which is able to accurately represent real traffic and capture the characteristics of bursty traffic. We analyze the performance of the switch in terms of cell-loss ratio (CLR), mean queue length and mean delay time. We show that the SAN model is very useful in ATM switch model in that the gates have the capability of implementing of scheduling algorithm.

† 준 회 원 : 성균관대학교 대학원 전기전자 및 컴퓨터공학과
†† 정 회 원 : 성균관대학교 전기전자 및 컴퓨터공학부 교수

1. 서 론

고속 통신을 위한 ATM 기술의 발달은 여러 종류의 스위치를 등장시켰으며, 또한 고속 스위칭 기술의 발달을 가져왔다. 다양한 QoS(Quality of Service)를 요구하는 각 서비스에 부합하기 위해 ATM에서의 성능측정은 중요한 문제이다. 그러나, ATM에서 성능측정의 문제는 그 시스템의 복잡성 때문에 매우 어렵다[1]. 또한 큰 상태 공간을 고려해야 하는 문제는 복잡한 시스템의 성능평가에서 일반적으로 부딪치는 문제이다. 본 논문에서는 이러한 시스템의 복잡성과 모델링할 때 큰 상태 공간을 직접 고려하지 않아도 되는 모델링 기술로써 PN(Petri Net)[2-4]을 선택했다. PN을 이용한 모델링은 knockout 스위치[5]를 [6]에서 SWN(Stochastic Well-formed Net)을 이용해 이루어졌다. [7]에서는 PN을 이용해 폴링 시스템을 모델링하였다. [8]에서는 knockout 스위치의 셀 도착 과정을 MMBP(Markov Modulated Bernoulli Process)로 모델링하였다. PN을 이용한 통신망 분야, 특히 ATM의 모델링은 아직 시작 단계에 있다.

본 논문에서는 ATM 망의 성능측정을 위한 모델링 기술로서 PN의 일종이며 SPN(Stochastic PN)이나 GSPN(Generalized SPN)보다 확장된 SAN(Stochastic Activity Networks)[9]을 이용한다. SAN 모델의 성능평가를 위한 소프트웨어 패키지로써 UltraSAN[9]을 이용한다. SAN은 일반 PN의 순간 천이(immediate transition)와 SPN의 확률 천이(stochastic transition), 즉 점화율(firing rate)이 지수분포함수나 결정된(deterministic) 시간에 따른 천이를 동시에 이용할 수 있을 뿐 아니라 점화할 때 토큰을 변화시킬 수 있는 입·출력 게이트(input gate, output gate)를 지원함으로써 시스템을 좀더 유동적으로 모델링할 수 있는 장점이 있다. 입·출력 게이트 내에는 알고리즘을 첨가할 수 있어 통신망을 모델링할 때 스케줄링 등의 다양한 알고리즘을 구현하기에 용이하다. 모델링 방법에 따라서 모델의 확장이 용이하게 할 수 있다. [8]에서는 SAN을 이용해 knockout 스위치를 모델링하였지만 모델의 확장이 용이하지 않도록 되어있다. 즉 입·출력포트의 개수를 변화시키거나 필요에 따라 입력되는 트래픽을 다른 모델로 적용할 때 시스템의 전체 모델을 고려해야 하도록 되어있다. 따라서 본 논문에서는 이러한 것을 극복할 수 있는 SAN 모델을 제시한다.

본 논문의 대상모델은 수식적 분석은 이미 되어있지만, SAN을 이용한 모델링은 아직 되어 있지 않은 출력 포트에 제한된 크기의 버퍼를 갖는 ATM 스위치로 한다. 셀 도착 과정은 버스티 특성을 고려할 수 있어 실제 트래픽과 유사한 MMPP(Markov Modulated Poisson Process)로 모델링한다. 포아송 과정(poisson process)은 일반적으로 컴퓨터나 통신망에서 도착 과정 모델링을 위해 많이 사용되어왔지만, 버스티 트래픽의 특징을 표현하지 못하기 때문에 실제 트래픽을 정확히 표현하지 못한다. MMPP는 이러한 단점을 보완할 수 있는 트래픽 모델이다[10]. 2-상태를 갖는 MMPP를 SAN으로 모델링하여 ATM 스위치의 입력단에 도착하는 셀을 생성하는데 이용한다. MMPP의 파라미터를 바꾸면서 그에 따른 셀 손실률, 버퍼에 남아있는 평균 셀의 크기, 셀 지연시간 등을 측정한다.

본 논문의 구성은 시스템의 모델링에 적용된 SAN에 대한 원형(primitives)을 설명하고, SAN을 이용한 모델링의 예로 M/M/1[11] 큐잉 모델을 선택하여 SAN 원형을 적용하고 각종 파라미터를 설정하는 예를 보여준다. 그리고 시뮬레이션 결과를 그래프로 출력하여 보여준다. 3장에서는 출력 큐잉 ATM 스위치를 기능적인 측면에서 간단히 설명하고, 셀 도착 프로세스를 MMPP로 모델링하고 동작을 설명한다. 이어서 출력 큐잉 ATM 스위치를 모델링하고, SAN 모델에 적용된 각종 파라미터 값을 표로 보여준다. 4장에서는 성능분석 분야로 성능측정할 때 적용된 셀 손실률과 평균 지연시간에 대한 수식을 설명하고, 시뮬레이터에서는 어떻게 적용되었는지를 표로 보여준다. 또한 시뮬레이션 결과를 그래프로 보여주고 각 그래프에 대해 설명한 후 마지막 5장은 본 논문의 결론이다.

2. SAN(Stochastic Activity Networks) 표현

2.1 원형(primitives)

SAN은 확장된 SPN으로 시간을 갖는 천이(timed transition)와 순간 천이(instantaneous transition)를 동시에 지원한다. SAN에서는 천이를 액티비티(activity)란 용어로 사용한다. 액티비티는 사건 확률(case probability)을 지원한다. 액티비티의 주어진 시간이 완료(completion)됐을 때 액티비티에 주어진 확률에 따라서 출력 장소(output place)로 토큰을 옮기고 그 수만큼 입력 장소(input place)에서는 제거된다. 즉, 다음 상태

(state)로 상태 천이(state transition)가 일어난다. 액티비티의 사건 확률을 이용함으로써 모델링하는데 좀더 유동적으로 할 수 있게 한다.

액티비티는 입력 게이트(input gate)와 출력 게이트(output gate)가 연결될 수 있으며 입·출력 게이트는 토큰의 이동과 관련이 있다. 입·출력 게이트 내에는 모델링하는 사람이 제공되는 코드를 이용해 필요에 따라 설정할 수 있다. 예를 들면, 어떤 조건이면 액티비티를 활성화시키고, 토큰을 이동시킬 것인지를 설정할 수 있다. 입·출력 게이트 내에 설정할 때 쓰이는 코드는 GLOBAL_S(), GLOBAL_D(), MARK(place_name), if-else구문 등이 있으며 이것은 C언어와 유사하게 사용된다.

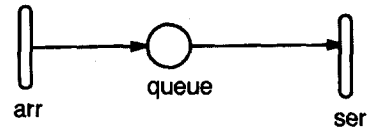
장소(place)는 SPN에서와 같은 역할을 한다. 어떤 동작상태(status)나 동작(operation)을 나타내는 것으로 액티비티가 활성화될 수 있는 조건을 표시한다. 장소 내에는 토큰을 가질 수 있다. 토큰은 시간을 갖는 액티비티의 경우에 그 액티비티에 지정된 시간이 지나면 다른 장소로 이동하게 된다. 시간을 갖지 않는 트랜지션(untimed transition)인 경우는 입력 장소의 토큰의 개수가 그 가중치(weight)보다 크거나 같은 때 다른 장소로 토큰이 이동한다. 가중치를 W, 입력 장소의 토큰의 개수를 I(P)라고 했을 때 토큰의 이동이 일어날 조건은 $W \leq I(P)$ 가 된다. 이 때 액티비티와 연결된 입·출력 게이트가 있을 경우는 게이트 내에 정의된 서술자(predicate)가 참일 때 토큰의 이동이 일어날 수 있는데, 어떤 장소로부터 다른 장소로 토큰이 이동하도록 각 게이트 내에 설정할 수 있다.

2.2 모델링 예

SAN을 이용해 시뮬레이션한 예를 보이기 위해 M/M/1 큐잉 모델을 선택했다. 앞에서 소개한 SAN을 실제 시뮬레이션에서 어떻게 적용시키는가를 보이기 위해서이다.

M/M/1 모델에 도착하는 프로세스는 포아송(Poisson)방식을 따르며, 서비스 시간은 지수 분포함수를 따른다. (그림 1)은 M/M/1 큐잉 모델을 시뮬레이션하기 위해 SAN으로 모델링한 것이다. 액티비티 arr(이하: A(액티비티 이름)으로 표시)은 도착 프로세스 모델이고, 장소 queue(이하: P(장소 이름)으로 표시)는 도착한 고객의 대기 행렬과 현재 서비스를 받고 있는 고객을 함께 표현한 것이다. 대기행렬과 서버를 분리해서

모델링한 것은 본 절의 (그림 6)에서 보여준다. A(ser)의 주어진 비율(rate)을 μ 라고 하면, $1/\mu$ 시간 후에 하나의 고객이 서비스를 받고 시스템을 떠난다.



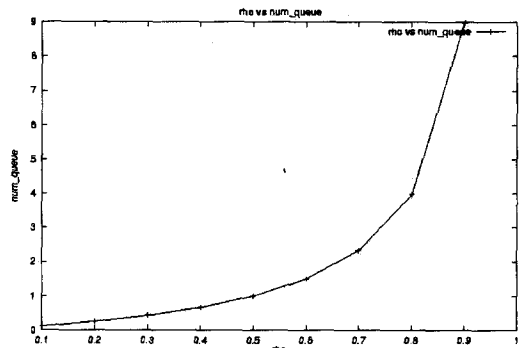
(그림 1) M/M/1의 SAN모델

서버가 서비스하는 시간의 정도를 표시하는 이용률(utilization) ρ 와 서비스를 받기 전에 시스템에서 기다리는 고객의 평균 수 \bar{N} , 한 고객이 시스템에서 보내는 평균 시간 T를 구할 수 있다. 도착하는 비율을 λ , 서비스하는 비율을 μ 라 하면, $\rho = \lambda/\mu$ 로 표시할 수 있다. 또한, $\bar{N} = \rho/(1-\rho)$ 로 계산할 수 있다. little's law[11]에 의해 $T = \bar{N}/\lambda$ 로 계산할 수 있다. 위의 식 중 실제 시뮬레이션에서 이용된 것은 T를 구할 때이다.

<표 1>은 M/M/1의 SAN모델을 시뮬레이션하기 위해 설정한 전역변수를 나타낸다. G-arr은 도착하는 비율로서 0.1부터 0.9까지 0.1씩 변화시켰고, G-ser는 서비스하는 비율을 나타내는 것으로 1.0으로 고정시켰다.

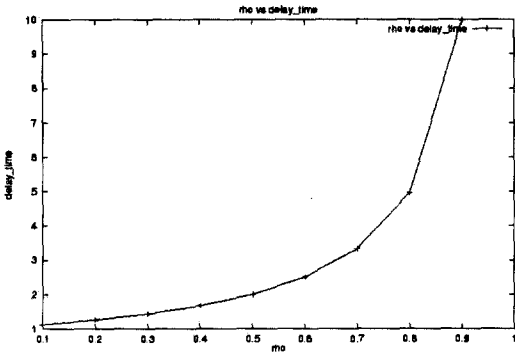
<표 1> 전역변수 설정 : M/M/1 SAN 모델

변수	분포	값
G-arr	exponential	[0.1, 0.9] by 0.1
G-ser	exponential	1.0

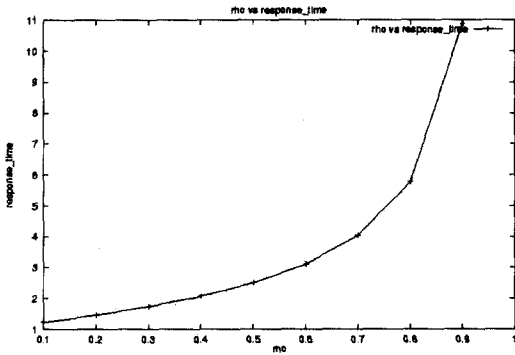


(그림 2) 이용률에 따른 시스템에서 대기하는 고객 수

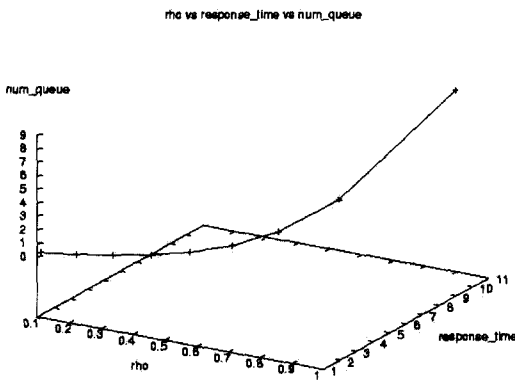
(그림 2), (그림 3), (그림 4), (그림 5)는 ρ 에 대한 대기 행렬, 대기 시간, 응답 시간 그리고 이것들 전체



(그림 3) 이용률에 따른 지연시간



(그림 4) 이용률에 따른 응답시간

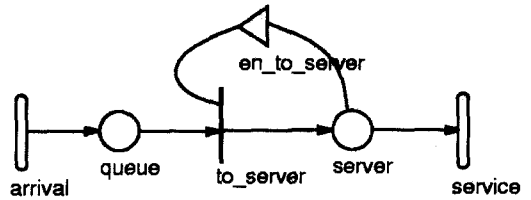


(그림 5) 이용률에 따른 지연시간과 응답시간

를 3차원 그래프로 표현한 것이다. 그래프에서 ρ 가 0.8 이상일 때는 대기 행렬의 수와 대기 시간, 응답 시간이 모두 급격히 상승함을 쉽게 알 수 있다. ρ 가 0.8일 때와 0.9일 때 시스템에 있는 고객의 수가 2배 이상 증가함을 볼 수 있다. 서버의 이용률과 대기행렬을 합

계 고려해야 할 경우 0.8미만의 ρ 를 유지하는 것이 바람직함을 알 수 있다.

(그림 6)은 앞에서 모델화된 M/M/1을 다른 방법으로 표현한 것이다. 앞에서의 모델과 다른 점은 대기행렬과 서버의 분리이다. 서버에서 서비스 받고 있는 고객과 큐에서 대기하는 고객을 시뮬레이션을 통해 각각 구할 수 있게 하기 위해서이다.



(그림 6) M/M/1의 다른 SAN 모델

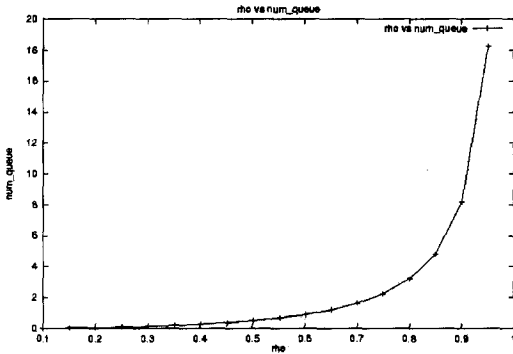
<표 2> 입력 게이트 en_to_server의 설정

서술자	함수
MARK(server) < 1	;/ * enable */

동작은 A(arrival)에 설정된 시간마다 도착하는 고객은 P(queue)에서 대기한다. 이 때 P(server)에 한 사람도 없으면 곧바로 P(server)로 이동하여 서비스를 받는다. A(service)는 서비스 받는 시간을 지수분포함수로 설정하였다. 입력 게이트 en_to_server(이하: I(입력 게이트 이름))으로 표시)는 순간 액티비티(immediate activity)인 to_server를 활성화시키는데, P(server)에 토큰이 없을 때만 활성화되도록 했다. 물론, P(server)에 토큰이 한 개 이상 있어야 한다.

I(en_to_server)의 설정은 if (MARK(server) < 1); /* enable */이다. 도착 시간 $1/\lambda$ 을 1로 고정시키고 서비스 시간 $1/\mu$ 을 변화시켰을 때, 즉 이용률 ρ 를 변화시켰을 때의 시뮬레이션 결과를 (그림 7)에서 보여주고 있다. 이용률이 증가함에 따라 큐에 쌓이는 고객의 수는 증가한다. 전체 시스템에서 대기하는 고객의 수도 큐에서 기다리는 수와 같은 유형의 그래프를 출력할 수 있다.

같은 시스템이라 하여도 SAN을 이용해 모델링하는 데는 꼭 한 가지만 있는 것은 아니다. 다르게 모델링할 수도 있다. 그러나 측정하고자 하는 관심부분이 같은 경우에 그 결과는 같게 나와야 할 것이다. 모델링의 예로 든 M/M/1의 두 가지 모델을 통해 이것을 확인할 수 있다.

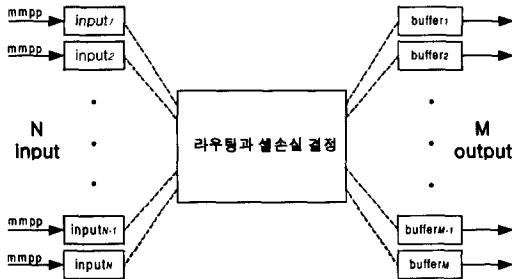


(그림 7) 이용률에 따른 큐에서 대기하는 고객 수

3. SAN 모델

3.1 모델링 대상

본 논문의 SAN을 이용해 모델링하고자 하는 대상은 출력포트에 버퍼를 갖는 ATM 스위치이다. (그림 8)은 ATM 스위치의 기능적인 면에서 표현한 것이다.



(그림 8) 모델링 대상 : ATM 스위치

(그림 8)에서 보는 바와 같이 입력포트에 도착하는 셀은 버스티 트래픽 특징을 표현할 수 있어 실제 트래픽과 유사한 2-상태를 갖는 MMPP 모델을 적용한다. 각 포트에 도착하는 셀은 서로 독립적이다. 본 논문에서는 각 입력포트에 도착하는 셀 중에서 어느 특정 출력포트로 향하는 셀만을 고려한다. 각 출력포트에서 서로 독립적인 버퍼를 갖고 있기 때문에 하나의 출력포트를 고려한 모델링은 다른 출력포트에도 똑같이 적용될 수 있으므로 특정 출력포트에 대해서만 모델링한다. 각 입력포트에 도착하는 셀 중 특정 출력포트로 향하는 셀이 1 셀 시간(cell time)내에 출력단의 버퍼에 여유가 있으면 버퍼에 쌓이게 된다. 출력단 버퍼가

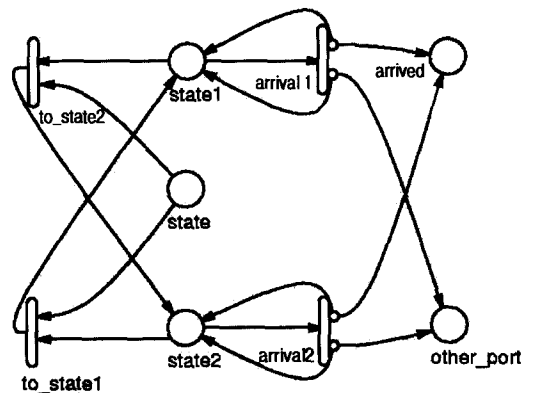
꽉 찼을 경우에는 셀은 버려지게 될 것이다. 셀을 버릴 것인지 아닌지를 결정하는 것은 ATM 스위치의 기능적 모델인 그림에서 라우팅과 셀손실 결정 부분에서이다. 입력단에 도착하는 셀은 1 셀 시간 내에 모두 처리된다고 가정한다. 라우팅과 셀손실 결정에 대한 모델링과 그것에 대한 설명은 본 논문의 3.3절에서 논의될 것이다. 버려지지 않고 출력단 버퍼에 쌓여있는 셀은 일정한 시간 간격으로 셀을 서비스하게 된다.

입력포트의 수를 변화시키거나 입력단에 도착하는 트래픽 모델을 변경하기 위해 모델 전체를 바꿔야 하는 일이 있을 수 있다. 다음 두 절에서 제시하는 SAN 모델을 이용하면 모델의 일부분만을 교체함으로써 모델 전체를 바꾸지 않아도 된다. 입력단의 트래픽 모델과 스위치부분을 따로 모델링함으로써 가능하게 한다. 실제 네트워크의 트래픽 형태에 따라 셀 도착 과정 모델을 변경하여 시뮬레이션해야 할 경우 유용할 것이다.

3.2 셀 도착 프로세스 : MMPP

셀 도착 프로세스는 두 가지 상태를 갖는 MMPP 모델을 적용하여 SAN으로 모델링하였다. MMPP 모델은 천이율(transition rate)과 도착율로 표현될 수 있다. σ_{ij} 를 $i \neq j$ 일 경우, 상태 i 에서 상태 j 로 천이되는 비율이라 하고, 상태 i 에서의 셀 도착율(생성율)을 λ_i 라 하면 다음과 같이 2-상태 MMPP를 행렬로 표현할 수 있다.

$$Q = \begin{bmatrix} -\sigma_{11} & \sigma_{12} \\ \sigma_{21} & -\sigma_{22} \end{bmatrix} \quad A = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



(그림 9) 셀 도착 프로세스 : MMPP의 SAN 서버네트 모델

(그림 9)는 ATM 스위치의 입력단에 도착하는 셀을

2-상태를 갖는 MMPP를 SAN으로 모델링한 것이다. <표 3>은 SAN 모델에 적용된 각 액티비티에 대한 설정으로 모든 액티비티는 지수분포함수를 따르도록 하였다. <표 4>는 사건 확률에 대한 설정이다. 초기 마킹을 x_0 라 하면 $x_0 = \{P(\text{state}), P(\text{state1}), P(\text{state2}), P(\text{arrived}), P(\text{other_port}) = \{0, 1, 0, 0, 0\}$ 로 설정하였다. $P(\text{state1})$ 에만 1개의 마킹을 설정하였다. 이 마킹은 $P(\text{state1})$ 과 $P(\text{state2})$ 를 이동하면서 각 장소에 머무는 동안 셀을 생성하는데 사용된다.

<표 3> 액티비티 설정 : MMPP의 SAN 서브네트 모델

액티비티	분포	파라미터 값
arrival1	exponential	
	rate	GLOBAL_D(gen_rate1)
arrival2	exponential	
	rate	GLOBAL_D(gen_rate2)
to_state1	exponential	
	rate	GLOBAL_D(trans_rate1)
to_state2	exponential	
	rate	GLOBAL_D(trans_rate2)

<표 4> 액티비티 사건 확률 설정 : MMPP의 SAN 서브네트 모델

액티비티	사건	확률
arrival1	1	return(GLOBAL_D(P)/GLOBAL_D(N));
	2	return(1 - GLOBAL_D(P)/GLOBAL_D(N));
arrival2	1	return(GLOBAL_D(P)/GLOBAL_D(N));
	2	return(1 - GLOBAL_D(P)/GLOBAL_D(N));

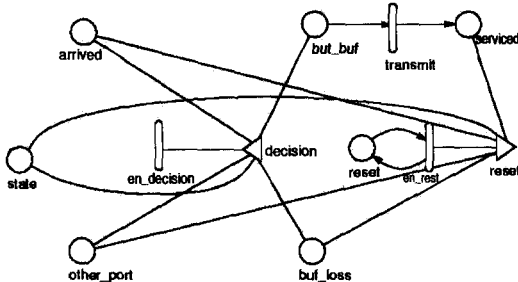
$P(\text{state1})$ 에 토큰이 존재할 때 $A(\text{arrival1})$ 과 $A(\text{to_state2})$ 가 활성화될 수 있고, 액티비티의 주어진 시간이 완료됐을 때 점화된다. 이로써 토큰은 다음 장소로 이동한다. 토큰의 이동은 결국 상태 천이를 의미하므로 상태 천이가 일어난다. $P(\text{state1})$ 은 $A(\text{to_state2})$ 에 설정된 시간이 완료되기 전까지 머무는 것을 표현한 것으로 그 시간은 위의 Q행렬에서 $1/\sigma_{12}$ 이다. $A(\text{arrival1})$ 에 설정된 시간은 $1/\lambda_1$ 이며, $P(\text{state1})$ 에 토큰이 있을 때 활성화되고, 설정된 시간이 완료되면 점화하여 토큰이 주어진 사건 확률에 따라 $P(\text{arrived})$ 나 $P(\text{other_port})$ 로 이동한다. 이동하는 토큰과 같은 수를 $P(\text{state1})$ 로 다시 복사한다. 즉 $A(\text{arrival1})$ 이 점화되어도 토큰은 $P(\text{state1})$ 에 계속 남아 있게 된다. MMPP 모델에서 상태 1에 머무는 동안 λ_1 의 비율로 계속 셀이 도착하는 것을 모델링한 것이다. 특정 출력단으로 향하는 셀은 $P(\text{arrived})$ 에 놓이게 된다. $P(\text{state1})$ 에 머

무는 동안 $A(\text{arrival1})$ 과 $A(\text{to_state2})$ 가 동시에 활성화되는데, $A(\text{to_state2})$ 의 주어진 시간 $1/\sigma_{12}$ 이 완료되면 토큰은 $P(\text{state2})$ 로 이동하게 된다. $P(\text{state})$ 에는 토큰이 한 개 이상 있어야 한다. $P(\text{state})$ 는 한 셀 시간이 지났을 때 토큰이 없으면 다시 1로 초기화되도록 했다. ATM 스위치 모델링 부분인 3.3절에서 다시 언급한다. $P(\text{state2})$ 에 토큰이 있을 때 $A(\text{to_state1})$ 과 $A(\text{arrival2})$ 는 활성화된다. $A(\text{arrival2})$ 에 설정된 $1/\lambda_2$ 시간이 지나면 토큰은 $A(\text{arrival2})$ 에 설정된 사건 확률에 따라 특정 출력단으로 향하는 셀을 $P(\text{arrived})$ 에 보내고, 그렇지 않은 셀은 $P(\text{other_port})$ 에 보내진다. 동시에 같은 수의 토큰을 다시 $P(\text{state2})$ 에 복사한다. $P(\text{state2})$ 와 $P(\text{state})$ 에 토큰이 있을 경우 $A(\text{to_state2})$ 는 활성화되고, $A(\text{to_state2})$ 에 설정된 시간 $1/\sigma_{21}$ 이 지나면 토큰은 $P(\text{state1})$ 로 이동하게 된다. 토큰이 $P(\text{state1})$ 에 있을 경우는 앞에 설명된 과정을 반복해서 거치게 된다. 이 과정들은 모두 1 셀 시간 간격으로 계속해서 일어난다. 1 셀 시간이 지난 후에는 $P(\text{state})$ 를 초기화하고 다음 슬롯(slot)으로 넘어간다.

<표 4>에서 P는 입력단에 도착하는 셀이 존재할 확률이고, 전역변수로 설정하여 시뮬레이션할 때 변경시킬 수 있도록 했다. N은 입력포트의 개수이고, 시뮬레이션할 때 변경할 수 있도록 설정한 실수형 전역변수이다. $A(\text{arrival1})$ 에 의해 생성되는 셀 중 특정 출력포트로 향하는 셀은 P/N 확률로 $P(\text{arrived})$ 에 도착하게 되고, 다른 출력포트로 향하는 셀은 $1-P/N$ 의 확률로 $P(\text{other_port})$ 에 놓이게 된다. $P(\text{arrived})$ 와 $P(\text{other_port})$ 에 놓인 셀들은 ATM 스위치 모델에서 처리하도록 했다.

3.3 출력큐잉 ATM 스위치

(그림 10)은 출력단에 버퍼를 갖는 ATM 스위치를 모델링한 것으로 출력단의 버퍼 용량에 따라 손실되는 셀을 측정하기 위한 모델이다. 그리고, 1 셀 시간 동안에 서비스된 셀도 같이 측정할 수 있도록 했다. 앞 절의 셀 도착 프로세스 모델에 의해 도착하는 셀들은 $P(\text{arrived})$ 와 $P(\text{other_port})$ 에 놓인다. 사건 확률에 의해 결정된 $P(\text{arrived})$ 에 놓이는 셀들이 특정 출력단으로 향하는 셀이 된다. $P(\text{state})$ 는 앞 절의 셀 도착 프로세스에서 사용된 것과 같은 것으로 1 셀 시간 동안만 셀을 생성하기 위해 두었다. 다음 셀 시간에는 1로 초기화된다. 초기화시키는 것은 출력 게이트 reset(이



(그림 10) ATM 스위치의 SAN 서브네트 모델

하 : O(출력 게이트 이름)으로 표시)에 의해서이다. P(out_buf)는 출력단에 있는 하나의 버퍼를 위해 두었으며 그 크기를 변경하면서 시뮬레이션할 수 있도록 전역변수 Q로 설정하였다. P(buf_loss)는 P(out_buf)의 용량이 허락하지 않을 경우 버려지는 셀을 측정하기 위해 두었다. P(reset)은 입력단의 셀 도착과 측정을 위한 다른 장소의 초기화를 위해 두었다. 결국 1 셀 시간을 결정하는 역할을 한다. A(en_reset)에 설정된 결정된 시간 간격이 완료되면 O(reset)에 설정된 값에 의해 토큰을 이동하거나 초기화시키는 역할을 한다. P(serviced)는 1 셀 시간 동안 출력포트를 떠나는 셀을 측정하기 위해 두었다. A(en_decision)의 설정은 <표 5>에서 보는 바와 같이 결정된 시간 1/N으로 하였다. 셀 손실 결정이 고속으로 이루어져야 하는데 최악의 경우인 입력포트 N의 역수 시간 사이에 이루어지도록 했다. 이 시간이 완료되면 I(decision)의 설정에 따라 셀이 이동하게 된다. I(decision)에 설정된 내용은 <표 6>에 나타내었다. 실제로 셀 손실 결정을 하는 부분이다. 도착한 셀이 출력포트의 버퍼에서 수용할 수 있으면 버퍼를 모델링한 P(out_buf)에 셀을 보내고, 그렇지 않을 경우는 P(buf_loss)에 셀을 보낸다. P(out_buf)에 놓인 셀은 A(transmit)에 설정된 결정된 시간 간격으로 서비스하게 된다. 서비스된 셀은 P(serviced)에 놓이며, 1 셀 시간동안 서비스된 평균 셀을 계산할 수 있도록 했다. 버퍼에 셀이 존재하는 한 계속해서 A(transmit)에 의해 전송된다. A(en_reset)에 의해 1 셀 시간마다 P(arrived), P(other_port), P(buf_loss)와 P(serviced)는 0으로 초기화되고, P(state)는 1로 초기화되어 다음 슬롯으로 넘어간다. 초기화는 O(reset)에 의해 이루어지며 이것에 대한 설정 내용은 <표 7>에서 보여주고 있다. O(reset)에 의해 초기화된 후 다음 슬롯으로 넘어가는데, 다음 슬롯에서도 전 슬롯과 마찬가지로

가지로 같은 과정을 반복하게 된다. 이렇게 반복되는 과정을 통해 평균적으로 버퍼에 남아있는 셀의 수를 P(out_buf)에 있는 토큰의 개수로 측정할 수 있고, 버려지는 셀이 놓이는 P(buf_loss)에 의해 평균 셀 손실을 구할 수 있다. 이것을 구하는 방법과 설정은 4.1절에서 다룬다.

<표 5> 액티비티 설정 : ATM 스위치의 SAN 서브네트 모델

액티비티	분 포	파라미터 값
en_decision	deterministic	
	value	1.0/GLOBAL_D(N)
en_reset	deterministic	
	value	1.0
transmit	deterministic	
	value	1.0

<표 6> 입력 게이트 설정 : ATM 스위치의 SAN 서브네트 모델

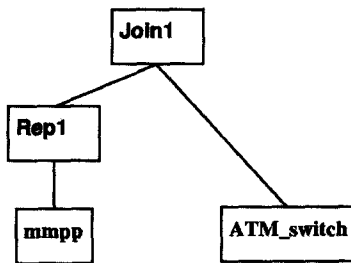
게이트	정 의
decision	Predicate MARK(state) <= 0
	Function if(MARK(arrived) > (GLOBAL_D(Q)-MARK(out_buf))) { MARK(buf_loss) = MARK(arrived)- (GLOBAL_D(Q)-MARK(out_buf)); MARK(out_buf) = GLOBAL_D(Q); } else { MARK(buf_loss) = 0; MARK(out_buf) += MARK(arrived); }

<표 7> 출력 게이트 설정 : ATM 스위치의 SAN 서브네트 모델

게이트	정 의
reset	MARK(arrived) = 0; MARK(other_port) = 0; MARK(serviced) = 0; MARK(buf_loss) = 0;
	if(MARK(state) <= 0) { MARK(state) = 1; }

(그림 11)은 셀 도착 프로세스와 ATM 스위치의 SAN 모델을 결합하기 위한 UltraSAN에서 제공하는 Composed Model을 이용한 것이다. 그림에서 mmpp는 셀 도착 과정의 SAN모델이고, 앞의 (그림 9)에서 그

세부 모델을 보이고 동작을 설명하였다. ATM_switch는 앞의 (그림 10)에서 세부적인 SAN 모델과 동작을 설명하였다. Rep1은 SAN의 서브네트모델인 mmpp를 복사하기 위한 것이다. mmpp 모델을 몇 개 복사할 것 인지는 Rep1에 개수를 설정함으로써 결정할 수 있다. 본 논문에서는 복사 개수를 8개로 하였다. MMPP 트래픽 형태로 8개의 각 입력포트에 독립적으로 셀이 도착하는 것이다. Join1은 여러 개로 복사된 mmpp와 ATM_switch 서브네트모델에서 공통되는 장소를 결정해 주고 서로 연결해 주는 역할을 한다. 공통되는 장소는 P(state), P(arrived), P(other_port)이다. 결국 각 입력포트에 서로 독립적으로 MMPP 형태로 셀이 도착하는 것이며, 입력포트의 수는 8포트가 된다. 입력포트의 수를 증가시키려면 Composed Model에서 Rep1의 설정을 증가시키면 되고, 그렇게 함으로써 전체 모델을 변경하지 않고도 간단히 모델을 바꾸어 시뮬레이션할 수 있다. 본 논문에서는 PN의 일종인 SAN으로 ATM 스위치를 모델링하는데 중점으로 두었기 때문에 입력포트는 8포트로 설정하여 시뮬레이션하였다. 입력단에 도착하는 셀들의 트래픽 형태를 바꾸기 위해서는 (그림 11)의 mmpp 서브네트모델을 다른 서브네트모델로 교체함으로써 모델 전체를 바꾸지 않고도 가능하게 해 준다.



(그림 11) Composed Model을 통한 서브네트 모델의 결합

4. 성능측정

4.1 성능분석 방법과 설정

SAN으로 모델링된 셀 도착 과정과 출력단에 유한한 버퍼를 갖는 ATM 스위치의 성능평가를 위한 설정 중, 평균 셀 손실률과 평균 지연시간을 구하기 위해 아래의 식을 적용하였다. 두 식은 UltraSAN에서 평균 셀 손실률과 지연시간을 구하기 위해 그대로 적용되었

으며, 그 설정은 <표 8>에 제시하였다.

<표 8> UltraSAN에서의 성능측정을 위한 설정

변수	정의
CLR	$\frac{\text{서술자}}{\text{합수}}$ $1 \frac{(\text{double})\text{MARK}(\text{buf_loss})}{(\text{double})\text{MARK}(\text{arrived})}$
Mean_buf_loss	$\frac{\text{서술자}}{\text{합수}}$ $1 \text{MARK}(\text{buf_loss})$
Mean_out_buf	$\frac{\text{서술자}}{\text{합수}}$ $1 \text{MARK}(\text{out_buf})$
Mean_delay	$\frac{\text{서술자}}{\text{합수}}$ $1 \frac{(\text{double})\text{MARK}(\text{out_buf})}{(\text{double})\text{MARK}(\text{arrived})}$
Mean_serviced_cell	$\frac{\text{서술자}}{\text{합수}}$ $1 \text{MARK}(\text{serviced})$
Mean_arrived_cell	$\frac{\text{서술자}}{\text{합수}}$ $1 \text{MARK}(\text{arrived})$

특정 출력포트의 버퍼 크기를 Q, T 셀 시간에 도착한 셀 개수를 K_T , T 셀 시간에 사용 중인 버퍼의 크기를 R_T 라고 하면, 한 셀 시간 당 입력되는 셀 중에서 출력단의 버퍼에 도달하지 못하고 버려지는 셀의 비율인 CLR(Cell Loss Ratio)은 아래의 식에 의해 계산할 수 있다. 이것은 버퍼에서 수용할 수 있는 용량에 따른 셀 손실률을 말한다.

$$CLR = \frac{E[K_T - (Q - R_T)]}{E[K_T]}$$

또한, T 셀 시간에 특정 출력포트의 버퍼에 남아 있는 셀의 개수를 N_T 라고 하면 전체 스위치에서의 평균 지연시간 \bar{W} 는 little's law[11]를 이용하여 아래의 식에 의해 계산할 수 있다.

$$\bar{W} = \frac{E[N_T]}{E[K_T]}$$

<표 8>은 성능측정을 위해 UltraSAN에서 설정된 내용을 보여준다. CLR은 모든 슬롯에 대해 평균 CLR을 구하기 위한 설정이다. Mean_buf_loss는 평균 셀 손실, Mean_out_buf는 출력단의 버퍼에서 대기하는 평균 셀 개수, Mean_delay는 평균 셀 지연시간, Mean_served_cell은 한 셀 시간 당 서비스되는 평균 셀 개수 그리고, Mean_arrived_cell은 평균적으로 도착하는 셀의 개수를 측정하기 위해 설정된 변수이다.

4.2 결과 분석

앞으로 논의될 수치 결과는 UltraSAN에서의 시뮬레이션 결과이다. 모든 시뮬레이션에서 신뢰구간은 95%로 설정하였다. UltraSAN에는 여러 가지 성능측정용 도구가 있는데, 그 중에서 steady-state simulator를 이용했다.

버스티 특성을 갖고 있어 실제 트래픽과 유사한 소스를 생성하는 MMPP 모델의 파라미터 값을 실제 환경과 일치하도록 선택하는 것은 어려운 일이다. 포트의 속도나 스위치의 입력포트에 도착하는 실제 트래픽의 형태가 각기 다르기 때문이다. 시뮬레이션을 위해 2-상태 MMPP 모델의 파라미터 4가지를 변화시키면서 버스티 특성을 갖는 트래픽을 생성할 수 있지만, 그것 중 상태 천이율을 변화시켜 버스티 트래픽을 생성했다. 그리고, 버퍼의 크기를 변화시키면서 생성된 트래픽에 따른 성능을 측정했다.

먼저 출력단의 버퍼의 크기를 변화시키고, 나머지 파라미터 값은 고정시켰다. 입력포트의 개수 N은 8, 입력포트에 셀이 존재할 확률 P는 0.9, 2-상태 MMPP 모델의 상태 1에서 상태 2로 천이되는 비율 σ_{12} 는 1, 상태 2에서 상태 1로 천이되는 비율 σ_{21} 을 10으로 설정했다. σ_{12} 와 σ_{21} 의 역수인 상태 1과 2에서 머무는 시간을 T_1 과 T_2 라 하면, T_1 과 T_2 시간에 생성되는 셀은 단위 시간당 10의 비율인 트래픽 환경으로 설정하였다. 평균적으로 도착하는 셀이 일정하기 때문에 버퍼의 크기가 증가함에 따라 셀 손실은 감소하며, 이러한 트래픽 환경에서는 $1e-05$ 이하의 셀 손실 결과를 얻기 위해서는 출력단의 버퍼 크기를 적어도 40셀 크기 이상의 메모리가 필요함을 알 수 있다.

두 번째는 MMPP 모델의 파라미터 중 상태 1에서

상태 2로 천이되는 비율인 σ_{12} 를 변화시키고 나머지 파라미터 값은 고정시킨 후 시뮬레이션하였다. σ_{12} 을 0.005~0.045까지 0.005씩 증가시켰고, σ_{21} 은 0.1, 생성율은 0.1과 1로 설정하였다. N은 8, P는 0.9, 버퍼의 크기 Q는 20으로 값을 설정하였다. 결과에서 버퍼에 남아있는 평균 셀 개수만을 고려했을 때는 T_1 의 값이 약 22일 때 남아있는 셀의 개수가 2미만임으로 2셀 크기의 버퍼 크기만 있으면 될 것으로 판단되지만, 같은 환경에서 평균 셀 손실 개수를 보면 약 $1e-04$ 개이다. 2셀 크기만으로는 버퍼의 크기가 충분하지 않다는 것을 알 수 있다. CLR이 $1e-05$ 이하의 값을 얻기 위해서는 T_1 파라미터 값이 100이어야 함을 알 수 있다. T_1 이 22일 때 CLR은 약 $1e-03$ 으로 나타났다. T_1 이 100일 때 평균 지연시간이 약 2이지만 T_1 파라미터 값이 22일 때 평균 지연시간은 약 9셀 시간임을 알 수 있다.

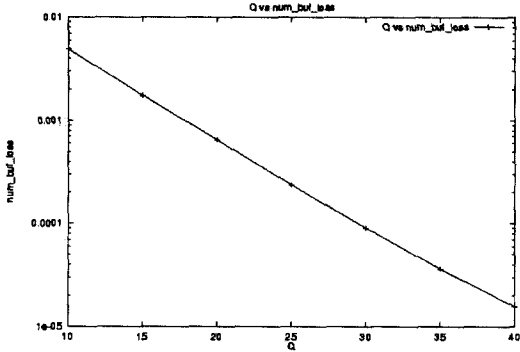
4.2.1 버퍼크기에 따른 평균 셀 손실

성능측정을 위한 전역변수와 성능측정 결과 중 하나인 평균 셀 손실을 <표 9>에 나타내었다. 표에서 G-로 시작하는 변수는 전역변수를 나타내며, P-로 시작하는 변수는 성능측정을 위한 변수이다. 표에서 G-Q는 출력포트의 버퍼 크기, G-N은 입력포트의 수, G-P는 입력포트에 셀이 존재할 확률, G-gen_rate1, 2는 MMPP 모델에서 셀이 생성되는 비율 λ_1, λ_2 를 각각 나타내는 것이고, G-trans_rate1, 2는 천이되는 비율인 σ_{12}, σ_{21} 을 각각 나타내는 변수이다. P-buf_loss는 평균 셀 손실을 위해 설정된 변수를 나타낸다. 다른 전역변수의 값은 고정시키고 버퍼 크기인 Q의 값만을 변화시켰을 때 평균 셀 손실을 구하기 위한 설정이다. Q의 변화에 따른 평균 셀 손실 값을 같이 나타냈다.

<표 9> 전역변수 설정과 버퍼 크기 변화에 대한 평균 셀 손실

변수	Exp1	Exp2	Exp3	Exp4	Exp5	Exp6	Exp7
G-Q	1.00e+01	1.50e+01	2.00e+01	2.50e+01	3.00e+01	3.50e+01	4.00e+01
P-buf_loss	4.91e-03	1.74e-03	4.49e-04	2.37e-04	9.09e-05	3.64e-05	1.58e-05

변수	G-N	G-P	G-gen_rate1	G-gen_rate2	G-trans_rate1	G-trans_rate2
값	8.00e+00	9.00e-01	1.00e+01	1.00e+01	1.00e+00	1.00e+01



(그림 12) 버퍼 크기에 따른 평균 셀 손실

(그림 12)는 <표 9>에서 설정된 값에 의해 출력된 결과인 P-buf_loss를 그래프로 출력한 것이다. 그래프에서 볼 수 있듯이 버퍼의 크기를 점점 크게 했을 때 손실되는 셀의 개수는 감소하는 것을 볼 수 있다. MMPP 모델에서 상태 1에 머무는 시간 $T_1 = 1/\sigma_{12}$ 라고 하고, 상태 2에 머무는 시간 $T_2 = 1/\sigma_{21}$ 이라고 하면 $T_1 = 1$, $T_2 = 0.1$ 시간 동안 각각 10의 비율로 셀이 생성되는 것이다. 결국 생성되는 셀의 수를 일정하게 유지하고 버퍼의 크기만을 변화시켰을 때의 결과이다.

4.2.2 버퍼에 대기하는 평균 셀

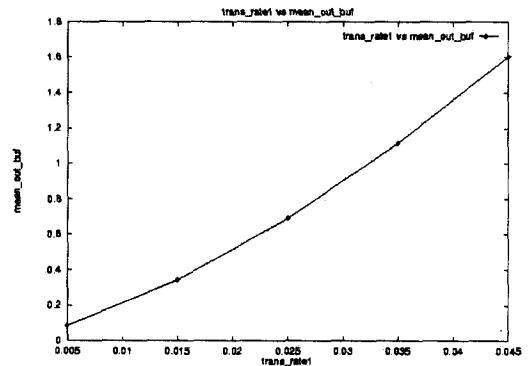
<표 10>은 또 다른 시뮬레이션을 위한 전역변수 설정과 측정결과 중 하나를 나타낸 것이다. T_1 시간을 변화시키고, T_2 와 다른 전역변수는 고정하였다. T_1 을 200부터 22까지 변화시켰고, T_2 는 10으로 고정하였다. 그리고, 셀 생성 비율을 0.1과 1로 설정하였다. 상태 1에서는 200~22 셀 시간 동안 평균적으로 0.1 비율로 셀이 생성되고, 상태 2에서는 10 셀 시간 동안 1의 비율로 셀이 생성된다.

<표 10> 전역변수 설정과 평균 셀 손실

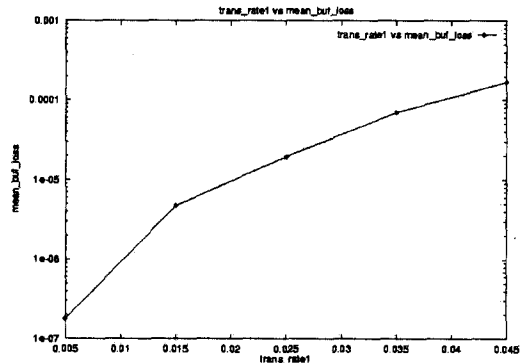
Source	Exp1	Exp2	Exp3	Exp4	Exp5
G-trans_rate1	5.00e-03	1.50e-02	2.50e-02	3.50e-02	4.50e-02
P-buf_loss	1.79e-07	4.72e-06	1.95e-05	6.99e-05	1.68e-04

변수	G-N	G-P	G-Q	G-gen_rate1	G-gen_rate2	G-trans_rate2
값	8.00e+00	9.00e-01	2.00e+01	1.00e-01	1.00e+00	1.00e-01

(그림 13), (그림 14), (그림 15), (그림 16)은 <표 10>에서 설정된 전역변수에 의해 측정된 결과를 그래프로 표현한 것이다. (그림 13)은 출력단 버퍼에 남아 있는 평균 셀의 개수를 그래프로 표시했다. (그림 13)의 결과로만 고려했을 경우, T_1 이 22이고 T_2 가 10인 시간 구간 사이에 0.1과 10의 비율로 도착하는 트래픽 형태를 갖는 스위치 환경일 경우 출력단의 버퍼 크기는 2개의 셀을 수용할 수 있는 용량이면 충분한 것으로 나타났다. 그러나, 다음 결과 그래프인 손실되는 평균 셀 개수를 참조해보면 실제 버퍼 크기는 20임에도 불구하고 손실되는 셀의 개수는 최대 $10e-3$ 까지 나타난다.



(그림 13) 버퍼에 남아있는 평균 셀



(그림 14) 평균 셀 손실

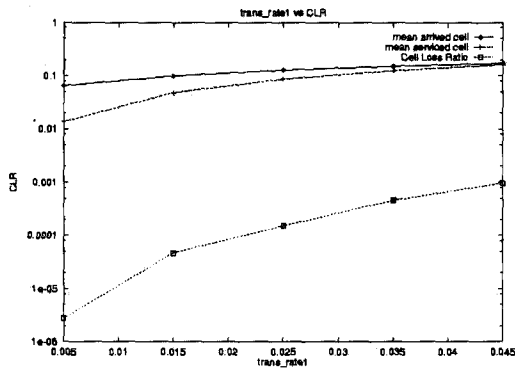
4.2.3 손실되는 평균 셀 개수

(그림 14)는 T_1 시간의 변화에 따라 손실되는 평균 셀 개수를 표시한 것이다. T_2 는 10으로 일정하게 유지

했을 경우이다. T_1 이 점점 증가함에 따라 생성되는 셀이 증가함에 따라 손실되는 평균 셀의 개수도 증가함을 알 수 있다. T_1 이 100 셀 시간, 즉 천이율 σ_{12} 가 0.01일인 트래픽 형태일 때 버려지는 셀의 평균 개수는 $10e-06$ 개이다. 100만 셀 시간 중 1개의 셀이 손실된다는 것이다. 이것은 100만 셀 시간마다 1개의 셀이 제한된 버퍼 크기로 인해 버려진다는 것을 의미한다.

4.2.4 평균 셀 손실률

(그림 15)는 T_1 을 변화시켰을 때 CLR을 표시했다. 출력단에서 서비스되는 시간은 결정된 시간 간격으로 계속 일어난다. 특정 출력단으로 향하는 셀의 평균 개수는 거의 일정하게 유지한다. 그렇지만 CLR이 T_1 의 변화에 민감하게 변하는 것을 볼 수 있다. T_1 의 변화에 따라 한꺼번에 도착하는 셀이 많음을 의미한다. 그러므로 특정 출력단의 버퍼 크기를 20 셀로 했음에도 CLR이 크게 나타나는 것을 볼 수 있다.

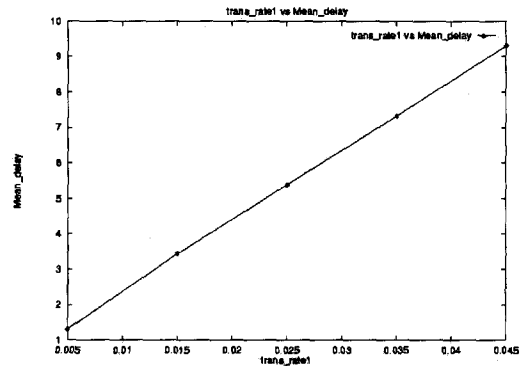


(그림 15) CLR

4.2.5 평균 지연시간

(그림 16)은 T_1 의 변화에 따른 평균 지연시간을 그 그래프로 표시한 것이다. 그림에서 볼 수 있듯이 T_1 이 감소함에 따라 지연시간은 선형적으로 증가한다. 이것은 T_1 의 변화에 대해 특정 출력포트로 향하는 셀의 수가 증가하고, 그에 따라 손실되는 셀의 양이나 CLR도 함께 증가함을 앞에서 그래프로 보여주었다. 특정 포트에 향하는 셀이 증가하면 그만큼 버려지는 셀도 많아진다. 뿐만 아니라 한꺼번에 도착하는 셀들도 출력단 버퍼의 남아있는 용량에 따라 버려지게 된다. 출

력단의 버퍼에 쌓이는 셀은 순간적으로 많아질 수는 있지만 평균적으로는 거의 일정하게 유지하게 된다. 그렇기 때문에 지연시간은 선형적으로 증가함을 볼 수 있다. 그래프에서 T_1 이 22일 때 평균 지연시간은 9셀 시간이다. T_1 이 22, T_2 가 10, 셀 생성율이 각각 0.1과 10인 트래픽 환경에서 출력단에 도착하는 셀은 평균 9셀 시간이 지난 후에 서비스를 받고 전송된다는 것을 알 수 있다.



(그림 16) 평균 지연시간

5. 결 론

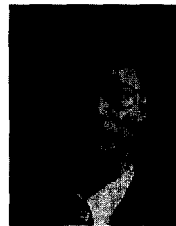
고속 통신망의 발달과 더불어 스위치의 발달을 가져왔다. 새로운 스위치의 구현을 전후해 고속 통신망에 맞는 성능을 내는가를 측정하는 것은 중요한 일이다. 복잡한 구조를 가진 시스템을 모델링하여 그 성능을 측정하는 것은 쉬운 일이 아니다. 포아송 과정으로는 트래픽의 버스티 특성을 표현할 수 없기 때문에 본 논문에서는 셀 도착 과정을 MMPP 트래픽 모델을 적용하여 SPN의 일종인 SAN으로 모델링하였다. 또한, 제한된 버퍼 크기를 갖는 출력 큐인 ATM 스위치를 SAN으로 모델링하고 그 성능을 측정하였다. SAN으로 모델링된 스위치는 UltraSAN 소프트웨어 패키지를 이용해 시뮬레이션하였다. SAN 모델을 위한 UltraSAN은 마코프 체인을 자동으로 생성해주고 해석해 준다. SAN에 의한 모델링은 모델의 전체를 바꾸지 않고 그 일부만 교체하는 것이 쉽고, 상태 공간을 직접 고려하지 않아도 되는 장점이 있다. 또한 SAN을 이용한 모델링은 알고리즘을 첨가할 수 있어 다양한 알고리즘을 적용하기에 용이하다.

본 논문에서 제시한 출력큐잉 ATM 스위치의 SAN 모델은 입력단의 셀 도착 과정을 달리하거나 입력포트의 수를 변화시킬 필요가 있을 때 그 확장이 용이함을 설명하였다.

출력큐잉 ATM 스위치의 버퍼 크기와 도착하는 셀의 트래픽 형태를 달리하기 위해 MMPP에서 T_1 의 값을 바꾸면서 그것에 따른 CLR, 버퍼에 남아있는 평균 셀 개수 그리고, 평균 셀 지연시간 등을 측정하여 그래프로 보여주고 분석하였다.

참 고 문 헌

- [1] Marco Ajone Marsan and Rossano gaeta, "Modeling ATM systems with GSPNs and SWNs," *Performance Evaluation Review*, Vol.26, No.2, pp.28-37, 1998.
- [2] Tadao Murata, "Petri Nets : Properties, Analysis and Applications," *Proc. of the IEEE*, Vol.77, No.4, pp.541-580, Apr. 1989.
- [3] Michael K. Molloy, "Performance Analysis Using Stochastic Petri Nets," *IEEE Trans. on Computers*, Vol.C-31, No.9, pp.913-917, Sep. 1982.
- [4] Giovanni Chiola et al, "Generalized Stochastic Petri Nets : A Definition at the Net Level and Its Implications," *IEEE Trans. on Software Engineering*, Vol.19, No.2, pp.89-107, Feb. 1993.
- [5] Yu-Shuan Yeh et al, "The Knockout Switch : A Simple, Modular Architecture for High-Performance Packet Switching," *IEEE Journal on Selected Areas in Communications*, Vol.SAC-5, No.8, pp. 1274-1283, Oct. 1987.
- [6] Rossano Gaeta and Marco Ajmone Marsan, "SWN Analysis and Simulation of Large Knockout ATM Switches," *ICATPN*, pp.326-344, 1998.
- [7] Oliver C. IBE, Kishor R. Tribedi, "Stochastic Petri Net Models of Polling Systems," *IEEE Journal of Selected Areas in Communications*, Vol.8, No.9, pp. 1649-1657, 1990.
- [8] L. A. Kant, "Analysis of Cell-Loss Processes and Restoration Schemes in ATM Networks," *PhD thesis, the university of arizona*, 1996.
- [9] W. H Sanders, "UltraSAN User's Manual v3.0," UIUC, 1995.
- [10] Robin A. Sahner, Kishor S. Trivedi, Antonio Puliafito, "Performance and Reliability analysis of Computer Systems-An Exmapple-Based Approach Using the SHARPE Software Package," Kluwer Academic Publishers, 1996.
- [11] Leonard Kleinrock, "Queueing Systems, Vol.1 : Theory," John Wiley & Sons. 1975.
- [12] H.Yoon et al, "The Knockout Switch Under Non-uniform Traffic," *IEEE Trans. on Communications*, Vol.43, No.6, pp.2149-2156, Jun. 1995.
- [13] W. H. Sanders *et al*, "The UltraSAN Modeling Environment," *Performance Evaluation*, Vol.24, pp. 89-115, 1995.
- [14] Mischa Schwartz, "Broadband Integrated Networks," Prentice Hall, 1996.
- [15] Christos G. Cassandras, "Discrete Event Systems- Modeling and Performance Analysis," Irwin and Aksen Assocites, 1993.
- [16] Raif O. Onvural, "Asynchronous Transfer Mode Networks : Performance Issues, 2nd Ed.," Artech House, 1995.



장 경 수

e-mail : kschang@ece.skku.ac.kr

1994년 성균관대학교 전기공학과

졸업(학사)

1998년 성균관대학교 대학원

(공학석사)

2000년 성균관대학교 대학원 전기전자 및 컴퓨터공학과
박사과정 수료

1994년~1995년 LG산전(주)

관심분야 : ATM, Discrete Event Systems, Industrial
Networks 등



신 호 진

e-mail : hjshin@ece.skku.ac.kr

1994년 성균관대학교 전기공학과
졸업(학사)

1999년 성균관대학교 대학원
(공학석사)

2000년 성균관대학교 대학원 전기
전자 및 컴퓨터공학과
박사과정

1994년~1995년 삼성중공업(주)

관심분야 : ATM, Industrial Networks, Discrete Event
Systems 등



신 동 렬

e-mail : drshin@ece.skku.ac.kr

1980년 성균관대 졸업(학사)

1982년 한국과학기술원
(공학석사)

1992년 Georgia Institute of
Tech. (Ph.D)

1994년~현재 성균관대학교 전기전자 및 컴퓨터공학부
부교수

1982년~1986년 Assistant researcher, Daewoo Heavy
Ind., Limited

1992년~1994년 Senior researcher, Samsung Data
Systems

관심분야 : Industrial networks including MMS, ICCP,
and Fieldbus, High-speed telecommuni-
cation networks with emphasis on ATM
modelling and switching, Performance eval-
uation with markovian and queueing net-
works 등