

# 개방형 멀티서비스 교환 시스템에서 GSMP 의 시험열 생성 기법

이현정\*, 최영일\*, 이병선\*, 전경표\*

\*한국전자통신연구원 교환전송기술연구소 통신시스템연구부

e-mail : [hjlee294@etri.re.kr](mailto:hjlee294@etri.re.kr)

## Test Case Generation of GSMP Protocol for Open Multiservice Switching System

Hyun-Jeong Lee\*, Young-Il Choi\*, Byung-Sun Lee\*, Kyung-Pyo Jun\*

\*Telecommunications System Department, Switching & Transmission Lab., ETRI

### 요 약

최근 인터넷 수요의 증가로 통신망에서 음성, 영상 및 데이터(data)를 복합적으로 지원할 수 있는 멀티서비스(multiservice)의 교환 기술이 필수적이다. 또한 망 사업자들이 여러 벤더(vendor)들로부터 최적의 장비를 선택하여 망을 구축할 수 있도록, 통신 장비들의 상호 운용성을 지원하는 개방형 구조의 망 및 스위치 시스템(switch system)을 정의하는 작업이 필요하다.

페트리 넷(Petri Nets)은 시스템(system)을 분석하기 위한 도구로서, 시스템은 페트리 넷 이론에 의해 시스템의 수학적 표현인 페트리 넷으로 설계될 수 있다. CPN(Colored Petri Nets)은 페트리 넷의 확장형으로서, 토큰(token)에 칼라를 부여하여 다양한 특성을 지닌 시스템을 표현하기에 적합하다. Design/CPN은 CPN의 사용을 지원하는 소프트웨어 패키지(software package)이다.

본 논문에서는 개방형 멀티서비스 교환 시스템의 핵심으로 스위치와 제어기(controller) 사이의 표준 프로토콜인 GSMP(General Switch Management Protocol) 프로토콜을 Design/CPN으로 변환하고, 이로부터 시험열을 생성한다.

### 1. 서론

최근 인터넷 수요의 증가로 통신망에서 음성, 영상 및 데이터를 복합적으로 지원할 수 있는 멀티서비스의 교환 기술이 필수적이다[1]. 개방형 멀티서비스 교환 기술은 통신망 사업자들의 경쟁력 강화와 교환 장비 제조 업체들의 이익 창출을 위한 것으로 개방형이라는 개념과 멀티서비스 통합이라는 개념을 기반으로 하고 있는 것이 가장 큰 특징이다. 개방형 개념은 통신망 사업자로 하여금 가격 및 제품 경쟁력이 있는 여러 벤더들의 제품으로 망 구축을 가능하게 하며, 필요에 따라서 망 장비 일부 기능의 추가, 교체, 기능 향상을 가능하게 할 것이다.

GSMP 프로토콜[2]은 Ipsilon 사(현재 Nokia)에서 개발한 범용 프로토콜로서, 스위치 레벨(level)의 접속을 설정, 해제 및 관리하는데 사용된다. 개방형 멀티서비스 교환 시스템에서는 개방형 개념에 따라 스위치와 제어기 사이의 인터페이스로 GSMP를 사용함으로써 스위치와 제어기의 독립적인 개발이 가능하도록 한다.

페트리 넷은 시스템을 분석하고 설계하는 모델링 기법으

로서, 시스템의 병행성을 표현하는데 적합하지만, 토큰이 단일 칼라를 가지므로 시스템의 특성을 표현하는데 한계를 지닌다. 페트리 넷의 확장된 모델인 CPN[2]은 컴퓨터 언어 문장과 관련된 특수화된 시각적인 네트워크로서, 시스템 모델링과 시뮬레이션에 전형적으로 사용된다. 이것은 토큰에 칼라를 부여하여 다양한 특성을 지닌 시스템을 표현하기에 적합하다.

본 논문에서는 GSMP 프로토콜에서 동기 관리 기능을 하는 인접(adjacency) 프로토콜을 CPN 형태로 변환한 후, CPN의 사용을 지원하는 소프트웨어 패키지인 Design/CPN에 적용하여 이로부터 시험열을 생성한다. 본 논문의 구성은 다음과 같다. 2 절에서는 개방형 멀티서비스 교환 기술과 GSMP 프로토콜에 대해 소개하고, 3 절에서는 GSMP 인접 프로토콜을 CPN 형태로 변환하는 방법을 설명한다. GSMP 인접 프로토콜을 Design/CPN에 적용하여 생성된 내부정보로부터 시험열을 생성하는 과정은 4 절에서 기술되고, 마지막으로 5 절에서는 결론 및 향후 연구방향을 제시한다.

### 2. 개방형 멀티서비스 교환 기술과 GSMP 프로토콜

2.1 개방형 멀티서비스 교환 기술

현재의 통신 인프라(infra)는 하나의 서비스 제어 방식과 운용 방식의 틀로 멀티서비스를 수용하는 방식이므로, 서비스 별로 서로 상이한 서비스 제어 방식과 관리 방식의 특성을 그대로 반영할 수 없으며, 서비스별로 독립적인 자원 관리 및 망 운용 관리가 이루어지지 못하고 있다.

이를 해결하기 위해서 제기된 개방형 멀티서비스 교환 기술은 하나의 교환 및 전송 인프라를 통해 현존하는 모든 서비스(음성, FR, ATM, IP 등)의 통합 수용과 새롭게 창출될 신규 서비스의 용이한 수용을 가능하게 한다. 또한, 이 기술은 새로운 신규 서비스를 손쉽게 도입하도록 도와줄 수 있으며 통신망 사업자로 하여금 하나의 공통된 통신망 인프라를 통해 현존하는 모든 서비스들과 함께 새로운 신규 서비스를 함께 수용하는 것을 가능하게 하며, 이와 더불어 이들 서비스들을 통합 관리할 수 있는 인프라 구축을 가능하게 할 것이다. 이러한 새로운 가능성들은 통신망 사업자들이 하여금 지금보다 훨씬 쉽고 원활한 방법으로 통신 장비의 일부분 또는 전체를 교체하거나 기능 확장이 가능하게 하여, 통신 장비 제조업체들의 이익 창출에 큰 도움을 줄 것이다.

2.2 GSMP 프로토콜

GSMP는 레이블(label) 스위치를 제어하기 위한 범용 프로토콜로서, 제어가 스위치를 통해 연결을 설정하고 해제할 수 있도록 하고, 스위치가 제어기에 비동기 이벤트(asynchronous event)를 통보하도록 한다[9]. GSMP는 비대칭(asymmetric)으로서, 제어가 마스터(master)가 되고, 스위치가 슬레이브(slave)가 된다. 하나의 제어기는 여러 스위치를 제어할 수 있고, 또한 하나의 스위치는 하나 이상의 제어기에 의해 제어될 수 있다.

개방형 멀티서비스 교환 시스템에서 GSMP 프로토콜은 제어부와 스위치부 사이에서 동작한다. GSMP를 이용하면 교환 시스템의 제어부와 스위치부는 상호 독립적인 개체가 가능하게 되어, 개방형 멀티서비스 교환 시스템을 구축할 수 있다. 그림 1과 같이 개방형 멀티서비스 교환 시스템에서 GSMP 마스터 부분은 제어부에 위치하고 GSMP 슬레이브 부분은 스위치에 위치한다.

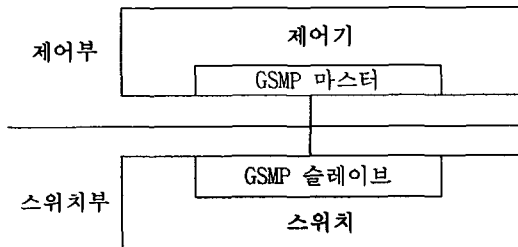


그림 1. 개방형 멀티서비스 교환 시스템에서의 GSMP 프로토콜.

본 논문에서는 GSMP 프로토콜에서 동기 관리 기능을 하는 인접 프로토콜을 CPN 형태로 변환하여 시험열을 생성한다. 인접 프로토콜을 사용하는 이유는, 초기에 GSMP 마스터와 슬레이브 간에 인접 프로토콜이 수행되어 일단 동기가 맞추어져 설정(establish) 상태가 되면 다른 상태로 변화하지 않기 때문이다.

3. GSMP 인접 프로토콜의 CPN 변환

본 절에서는 GSMP 인접 프로토콜과 CPN의 구성요소를 살

펴보고, GSMP 인접 프로토콜을 CPN으로 변환한다.

3.1 GSMP 인접 프로토콜 정의

GSMP 인접 프로토콜은 컨디션(Condition), 액션(Action), 새로운 상태(New state)로 구성된다. GSMP 인접 프로토콜을 정의하면 다음과 같은 3개의 요소(Q, C, A)로 구성된다.

$$GSMP\_Adjacency\_Protocol = (Q, C, A)$$

여기서,

Q : 유한한 상태( $q_1, q_2, \dots, q_n$ )의 집합

C : 컨디션( $c_1, c_2, \dots, c_j$ )의 집합

A : 액션( $a_1, a_2, \dots, a_k$ )의 집합

3.2 CPN 정의

CPN은 페트리 넷에 칼라의 개념을 추가하여 다음과 같이 6개의 튜플(P, T, C, I, O,  $\mu$ )로 구성된다[4].

$$CPN = (P, T, C, I, O, \mu)$$

여기서,

P : 플레이스(place)의 집합

$$= p_1, p_2, \dots, p_n, n \geq 0$$

T : 트랜지션(transition)의 집합

$$= t_1, t_2, \dots, t_m, m \geq 0$$

C : 칼라함수

$$C(p) = \sum k \cdot c, k \geq 0, k \text{는 계수, } c \text{는 칼라}$$

I : 입력함수

O : 출력함수

$\mu (P \rightarrow 0, 1, \dots)$  : 토큰의 초기 상태 집합

$$PN \cap T = 0 \text{ and } PUT \neq 0$$

3.3 GSMP 인접 프로토콜의 CPN 변환

GSMP 인접 프로토콜의 CPN 변환 방법 및 대응 관계는 다음과 같다.

$$P = QU \text{ Imsi}$$

Q : 상태

Imsi : Q와 Q 사이의 플레이스

$$T = \{t_{q, \text{imsi}} \mid q \in Q \text{ and } \text{imsi} \in \text{Imsi}\}$$

$$C(p) = C(q), \text{ if } p \in Q$$

$$C(\text{imsi}), \text{ if } p \in \text{Imsi}$$

$$I(t_{q, \text{c/a}}, \{\text{imsi}_c/\text{a}\}) = \{q_c/\text{a}, \{\text{imsi}_c/\text{a}\}\}$$

$q_c/\text{a}$  : 현재 상태 플레이스의 컨디션/액션

$\{\text{imsi}_c/\text{a}\}$  : 임시 플레이스의 컨디션/액션

으로서 생략되는 경우 존재

$$O(t_{q, \text{c/a}}, \{\text{imsi}_c/\text{a}\}) = Q'(q'_c/\text{a})$$

$Q'$  : 다음 상태 플레이스

$q'_c/\text{a}$  : 다음 상태 플레이스의 컨디션/액션

$$\mu(p) = Q(0) \cup \sum (\text{Imsi})$$

$$Q(0) : \text{초기상태 플레이스}$$

GSMP 인접 프로토콜의 상태(Q)는 CPN의 플레이스의 집합인 P로 표현한다. P는 두 가지로 정의하여, Q는 상태 플레이스로, Q와 Q 사이에 위치하는 임의의 플레이스인 Imsi는 임시 플레이스로 한다. 여기서 각 상태 플레이스는 컨디션/액션에 따른 칼라를 가지고, 초기 상태 플레이스에만 토큰이 부여된다. 초기 상태 플레이스는 토큰을 이용하여 점화하며 다음 상태 플레이스로 토큰을 전달한다. Imsi는 두 상태 플레이스 사이에 정의한 플레이스로서 컨디션/액션에 따른 칼라와 토큰을 가지며, 두 상태 플레이스 사이의 컨디션/액션을 나타내기 위해 정의한 것이다. 즉, 현재 상태 플레이스에서 다음 상태 플레이스로 점화 가능한 트랜지션이 여러 개 있는 경우 병행성을 나타내기 위해, 현재 상태 플레이스의 출력 아크(arc)에는 동일한 임의의 토큰을 출력하

고, 현재 상태 플레이스의 본래 컨디션/액션은 임시 플레이스로 표현한다. 여기서 컨디션과 액션은 각각 입력과 출력으로 대응되기 때문에 컨디션/액션으로 나타낸 것이다.

T는 현재 상태 플레이스 Q와 임시 플레이스 Imsi의 카테션 프로덕트(cartesian product)로 표현되며, 현재 상태 플레이스와 임시 플레이스가 접화하여 다음 상태 플레이스로 전이됨을 나타낸다.

토른은 초기 상태 플레이스와 임시 플레이스들에 위치하게 된다. 토른은 토른의 개수를 나타내는 계수와, 칼라로 구성된다. 예를 들면, A라는 칼라를 가진 토른 한 개는 1'a로 나타낸다. '는 back quote이다. 현재 상태 플레이스와 임시 상태 플레이스는 자신의 컨디션과 액션에 따라 칼라와 토른을 가진다. I는 상태 플레이스가 셀프루프를 처리하는 경우 그 상태 플레이스의 컨디션과 액션인 q\_c/a가 트랜지션의 입력이 되고, 상태 플레이스가 다음 상태 플레이스로 전이하는 경우 q\_c/a와 임시 플레이스의 컨디션과 액션인 imsi\_e가 트랜지션의 입력이 된다. O는 트랜지션의 수행 결과로서 다음 상태 플레이스 Q'에 토른을 출력한다. 현재 상태 idle에서 컨디션과 액션 "a"를 처리하고 다음 상태 q2로 전이하는 GSMP 인접 프로토콜을 CPN으로 변환하면 그림 1과 같다.

그림 1에서 GSMP 인접 프로토콜의 상태는 CPN의 상태 플레이스로 표현되고, 두 상태 플레이스 사이에는 컨디션과 액션에 따라 a1과 같은 임시 플레이스가 위치한다. 임시 플레이스 a1은 칼라 A1과 토른 a를 가진다.

t1은 a, x를 입력으로, 다음 상태 플레이스인 synsent와 다음 상태 플레이스의 토른인 b를 출력으로 한다.

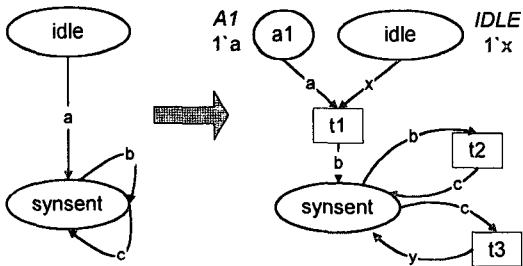


그림 1. GSMP 인접 프로토콜의 CPN 변환 방법

GSMP 인접 프로토콜의 CPN 변환 알고리즘은 다음과 같다.

- 단계 1) GSMP 인접 프로토콜의 상태를 CPN의 플레이스로 대응시키고 상태에 따라 칼라를 부여한다. 토른은 초기 상태 플레이스에만 부여한다.
- 단계 2) GSMP 인접 프로토콜의 컨디션/액션에 따른 전이를 CPN의 트랜지션으로 대응시킨다. 이때 셀프루프 전이는 트랜지션으로만 대응시키고, 다음 상태로의 전이는 트랜지션 및 컨디션/액션에 따라 임시 플레이스를 구성하여 트랜지션의 입력에 포함시킨다.
- 단계 3) 임시 플레이스에 컨디션/액션에 따라 칼라와 토른을 부여한다.
- 단계 4) 각각의 상태와 컨디션/액션에 대하여 step 1), 2), 3)의 과정을 반복한다.

4. GSMP 인접 프로토콜의 시험열 생성

본 논문에서는 GSMP 인접 프로토콜을 CPN 형태로 변환하여 Design/CPN에 입력하고, 시뮬레이션을 수행하여 생성된 OG와 디스크립터로부터 GSMP 인접 프로토콜의 시험열

을 생성한다. 인접 프로토콜을 사용하는 이유는, 초기에 GSMP 마스터와 슬레이브 간에 인접 프로토콜이 수행되어 일단 동기가 맞추어져 설정(establish) 상태가 되면 다른 상태로 변화하지 않기 때문이다.

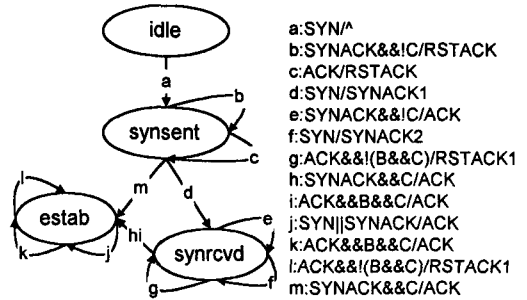


그림 2. GSMP 인접 프로토콜

그림 2와 같이 GSMP 인접 프로토콜은 4개의 상태와 12개의 컨디션, 액션을 가진다. 컨디션과 액션은 각각 입력과 출력에 대응하기 때문에 슬래쉬(/)로 구분하여 기호로 나타낸다. 이러한 기호는 간결하게 표현하기 위한 것이고, 시험열 생성 시에는 원래의 입력과 출력으로 나타낸다. 또한 동일 입력/출력인 경우에는 마지막에 숫자를 덧붙여 d, f와 같이 각각 다른 기호로 나타내었는데, 이것은 CPN으로 변경 시 토른의 고유성을 위한 것이다.

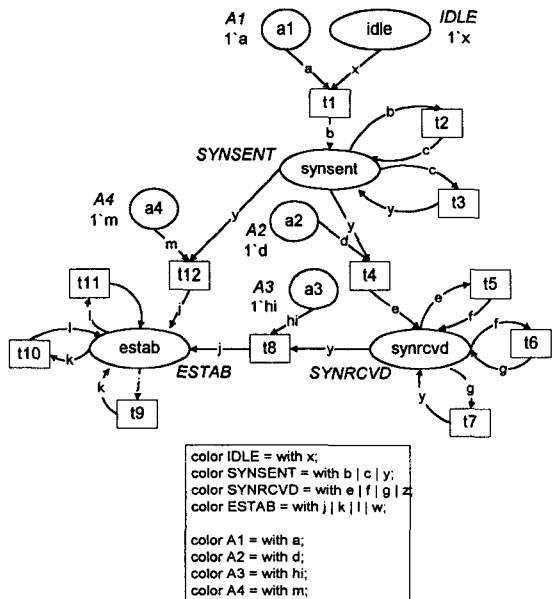


그림 3. GSMP 인접 프로토콜의 CPN 표현

그림 3에서 각 상태 플레이스는 고유한 칼라를 가지고, 초기 상태 플레이스인 idle만이 임의의 토른 x를 가진다. 초기 플레이스 idle이 synsent로 전이할 때 사용된 컨디션과 액션은 입력/출력에 대응되어 기호 a로 나타낸 후 임시 플레이스 a1의 칼라와 토른으로 구성되었다. 시험열 생성 시에는 각 입력/출력을 원래대로 나타낸다.

그림 3을 Design/CPN에 입력하여 얻은 결과인 OG와 디

스크립터는 그림 4와 그림 5에 나타나 있다.

그림 4는 그림 3의 각 플레이스를 노드로, 입력/출력을 아크로 표현하여, 초기의 현재 상태 플레이스인 idle 이 근 노드(root node)가 되어 idle로부터 점화가능한 모든 플레이스를 선화한 경로를 트리 형태로 나타낸 것이다. 그림 4와 같은 트리를 도달가능 트리[4]라고도 한다.

그림 4에서 각 노드에서 윗부분에 있는 것은 노드의 번호이고 아랫부분에 있는 것은 부모노드와 자노드의 개수이다. 첫번째 노드는 윗부분에 노드 번호 1, 아랫 부분인 0:1에 부모노드의 개수 0, 자노드의 개수 1을 나타낸다. OG의 각 노드는 그림 5와 같은 디스크립터를 가진다.

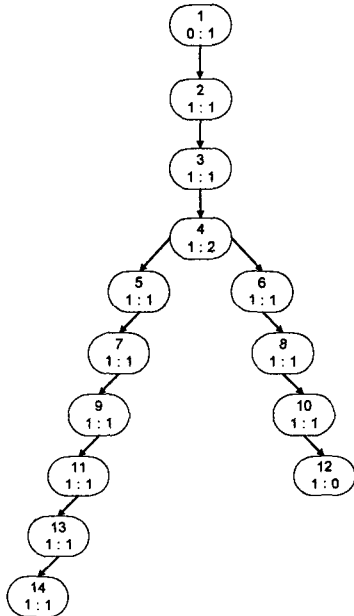


그림 4. 그림 3의 OG

디스크립터는 각 노드의 토큰 정보를 나타낸다. 따라서 디스크립터를 이용하여 그림 3의 플레이스와 토큰을 OG의 노드와 아크에 대응시킬 수 있다.

1	New idle 1:1'x
	New a1 1:1'a
	New synsent 1: empty
	New a2 1:1'd
	New synrcvd: empty
	New a3 1:1'hi
	New estab 1: empty
	New a4 1:1'm

그림 5. 그림 4의 노드 1의 디스크립터

표 1은 그림 4의 각 아크를 연결한 시험열로서, CPN으로 표현 시 기호로 간결화된 입력/출력을 원래대로 나타낸 것이다.

표 1. GSMP 인접 프로토콜의 시험열

시험열	
1	[idle] SYN/~- [synsent] SYNACK&&!/C/RSTACK-[synsent] ACK/RSTACK - [synsent] SYN/SYNACK1- [synrcvd] SYNACK&&!/C/ACK - [synrcvd] SYN/SYNACK2 - [synrcvd] ACK&&!(B&&C)/RSTACK1 - [synrcvd] SYNACK&&C/ACK, ACK&&B&&C/ACK - [estab]

	SYN  SYNACK/ACK - [estab] ACK&&B&&C/ACK - [estab] ACK&&!(B&&C)/RSTACK1 - [estab]
2	[idle] SYN/~- [synsent] SYNACK&&!/C/RSTACK-[synsent] ACK/RSTACK - [synsent] SYNACK&&C/ACK - [estab] SYN  SYNACK/ACK - [estab] ACK&&B&&C/ACK - [estab] ACK&&!(B&&C)/RSTACK1 - [estab]

### 5. 결론 및 향후 연구방향

최근의 무한 경쟁의 통신 시장의 환경 변화로 인해, 현존하는 서비스의 경제적 제공과 신규 서비스의 유연한 수요가 가능한 통신망의 조기 구축이 모든 통신망 사업자들의 관심사가 되고 있다[5-8]. 개방형 멀티서비스 교환 기술은 통신망 사업자들의 경쟁력 강화와 교환 장비 제조 업체들의 이익 창출을 위한 것으로 개방형이라는 개념과 멀티서비스 통합이라는 개념을 기반으로 하고 있는 것이 가장 큰 특징이다. 개방형 개념은 통신망 사업자로 하여금 가격 및 제품 경쟁력이 있는 여러 벤더들의 제품으로 망 구축을 가능하게 하며, 필요에 따라서 망 장비 일부 기능의 추가, 교체, 기능 향상을 가능하게 할 것이다.

GSMP 프로토콜은 Ipsilon사에서 개발한 범용 프로토콜로서, 스위치 레벨의 접속을 설정, 해제 및 관리하는데 사용된다. 개방형 멀티서비스 교환 시스템에서는 개방형 개념에 따라 스위치와 제어기 사이의 인터페이스로 GSMP를 사용함으로써 스위치와 제어기의 독립적인 개발이 가능하도록 한다.

페트리 넷은 시스템을 분석하고 설계하는 모델링 기법으로서, 시스템의 병행성을 표현하는데 적합하지만, 토큰이 단일 칼라를 가지므로 시스템의 특성을 표현하는데 한계를 지닌다. 페트리 넷의 확장된 모델인 CPN은 컴퓨터 언어 문장과 관련된 특수화된 시각적인 네트워크로서, 시스템 모델링과 시뮬레이션에 전형적으로 사용된다. 이것은 토큰에 칼라를 부여하여 다양한 특성을 지닌 시스템을 표현하기에 적합하다.

본 논문에서는 GSMP의 인접 프로토콜을 CPN 형태로 변환한 후, Design/CPN에 적용하여 이로부터 시험열을 생성하였다. 이러한 자동화된 틀을 사용함으로써 기존의 수작업에 의한 시험열 생성 방법에 비해 간편하고 효율적으로 시험열을 생성하였으며, 수작업으로 인한 오류의 가능성을 줄였다. 향후 연구방향으로는 GSMP 프로토콜을 SDL(Specification and Description Language)로 설계한 후 SDT(SDL Design Tool)를 사용하여 시험열 생성 및 검증하는 것이다.

### 참고문헌

- [1] 이병선, 이정규, 전경표. "개방형 멀티서비스 교환 기술", 텔레콤 제 16권 제 1호, pp. 34-51, 2000
- [2] Avri Doria, Fiffi Hellstrand, Kenneth Sundell, Tom Worster. "General Switch Management Protocol V3," draft-ietf-gsmp-05.txt, IETF, April 2000
- [3] K. Jensen. "Coloured Petri nets," Vol. 1, 2nd Ed., Springer-Verlag, 1997
- [4] 이현정, 우성희, 이상호. "Design/CPN을 이용한 테스트 케이스 생성", 한국통신학회 학술발표논문집, pp. 1285-1288, 1998.
- [5] Active and Programmable Networks, IEEE Network, May/June, 1998.
- [6] Programmable Networks, IEEE Communications Magazine, Oct. 1998.
- [7] A.A. Lazar, K.S.Lim, F.Marcocini. "Realizing a foundation for programmability of ATM networks with the binding architecture," IEEE JSAC, vol.14, No.7, Sep. 1996.
- [8] A.A. Lazar. "Programming telecommunication networks," IEEE Network, Sep./Oct., pp. 2-12, 1997.