

고도지능망의 기본호처리 SIB를 위한 프로토콜 모델링에 관한 연구

正會員 趙 顯 俊* 正會員 李 成 根* 正會員 金 惠 鎮*

A Study on Modeling of Protocol for Basic Call Process SIB in Advanced Intelligent Network

Hyun Joon Cho*, Sung Keun Lee*, Duck Jin Kim* *Regular Members*

要 約

고도지능망의 개념모델에서는 총괄기능평면에서 다양한 지능망서비스 제공을 위하여 서비스에 독립적인 기능블럭인 SIB(service independent building-block)을 정의하고 있다. 본 논문은 이중에서 가장 근간이 되는 기본호처리 SIB를 위한 지능망교환기(SSP: service switching point)와 서비스제어시스템(SCP: service control point) 간의 프로토콜 모델링에 대하여 기술하였다. 모델링을 위하여 Petri Net를 이용하며, 도달성 트리(reachability tree)를 분석하여 모델에 대한 검증을 한다. 본 논문의 결과는 고도지능망 기본호처리 SIB 설계와 구현을 위하여 이용될 수 있을 것이다.

ABSTRACT

SIBs(Service Independent Building-block) are defined in CCITT advanced intelligent network concept model for supporting various services in the future. This paper describes the protocol modeling and verification for basic call process SIB. For modeling, we use Petri Net and verify this modeling by analyzing reachability tree of Petri Net. Results of this paper should be used for design and implementation of basic call process SIB.

I. 서 론

미래의 다양한 서비스에 대한 요구를 수용하기 위하여 통신망은 나날이 광대역화, 개인화, 지능화되어지고 있다.(1) 고도지능망(AIN: advanced intelligent

network) 개념은 통신망의 지능화 추세에 맞추어 생겨난 개념으로, 서비스 이용자 관점에서 보면 기존의 전화망에서 서비스되었던 신용통화, 착신과금과 같은 부가서비스들과 크게 다르지 않지만, 망의 구조측면에서는 근본적으로 다르다. 즉, 기존의 통신망에서는 서비스가 전적으로 교환기 소프트웨어에만 의존하기 때문에 새로운 서비스가 추가되거나 기존의 서비스가 수정될 때에 통신망 내의 모든 교환기 소프트웨어를 수정 및 추가해야 하는 어려움이 있다. 이러

*高麗大學校 電子工學科
Dept. of Electronic Eng., Korea University
論文番號: 94-31

한 문제점들을 극복하고 미래의 다양한 서비스요구에 유연하고도 신속하게 대처하기 위하여 분산처리와 객체지향기법을 도입한 고도지능망 개념이 도입되었다.(2,3,4,5) 고도지능망은 서비스제공 측면에서도 한 단계 더 나아가 가입자가 서비스로직이나 서비스데이터를 직접 변경하여 자신에 맞는 서비스를 제공받을 수 있는, ISDN과 같은 다양한 정보 서비스를 처리할 수 있는 새로운 통신망 구조이다.

CCITT에서도 고도지능망에 대한 연구로 단기적 방안인 CS-1(capability set-1)에 대한 연구와 장기적 방안인 지능망 개념모델에 대한 연구를 수행하고 있다.(3,4) 그러나 아직 이들은 개념모델 수준에서 이루어지고 있고, 따라서 개념모델을 구체화하기 위한 연구가 시급한 실정이다.(2,5) 본 연구는 이러한 연구의 일환으로 CCITT의 단기적 연구인 CS-1에 대한 권고안을 기준으로하여, 지능망 개념모델의 총괄 기능평면에서 정의되는 SIB(service independent building block) 중에서 지능망에서 가장 기본이 되는 기본호처리 SIB(BCP SIB)를 구체화하기 위하여 지능망 개념모델의 물리평면에서 규정되는 SSF와 SCF 기능실체 사이의 프로토콜에 대하여 알아보고, 이에 대한 모델링 및 검증에 관하여 기술하였다.

본 논문의 구성은 2장에서 고도지능망구조에 대하여 개괄적으로 알아보고, 3장에서는 본 연구의 대상이 되는 기본호처리 SIB의 기능과 이 기능을 제공하기 위하여 필요한 프로토콜 시나리오에 대하여 기술하며, 4장에서는 3장에서 규정된 시나리오를 바탕으로 하여 페트리넷을 이용한 프로토콜 모델링에 대하여 알아보고, 도달성트리 분석하여 프로토콜에 대한 도달성, 생동성, 제한성등의 특성에 대하여 검증을 한다. 끝으로 5장에서 결론을 맺는다.

II. 고도지능망 구조(2,3)

고도지능망에서는 서비스제어시스템(SCP : service control point), 서비스데이터시스템(SDP : service data point), IP(intelligent peripheral) 그리고 지능망교환기(SSP : service switching point) 등의 물리적 시스템들을 망내에 포함시켜 이들간의 상호작용을 이용하여 분산처리함으로써 서비스를 제공하게 된다. 고도지능망의 물리적 구조는 다음 그림 1과 같다.

CCITT의 장기적 방안으로 연구되어지고 있는 개념모델에 따르면 서비스평면, 총괄기능평면, 분산기능평면, 물리평면등의 네가지 평면으로 지능망구조

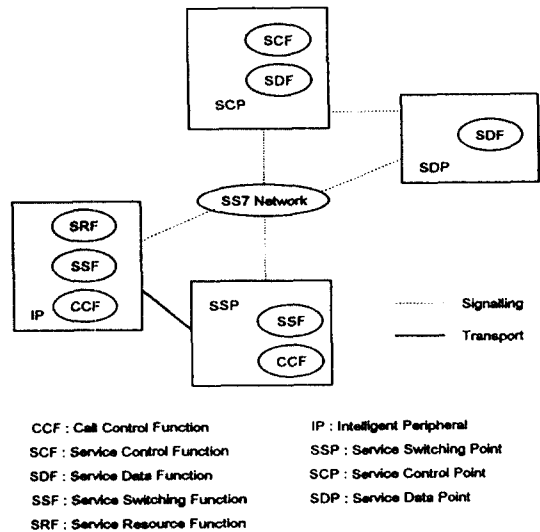


그림 1. 고도지능망의 물리적 구조
 Figure 1. Physical structure of AIN

를 나타내게 된다. 서비스평면은 단지 서비스 이용자의 관점에서 서비스를 나타내며 서비스의 구현과는 독립적이므로 구현에 대한 어떠한 정보도 나타나지 않는다. 총괄기능평면은 지능망을 하나의 실체로 모형화하고, 서비스를 구성하는 모듈인 SIB들은 총괄 기능평면에서 나타난다. SIB는 하나 이상이 모여서 서비스를 지원하며, 지원되는 서비스들과는 무관하게 정의하므로써 다른서비스에서의 재사용이 가능하게 된다. 분산기능평면은 지능망을 분산개념으로 모형화한다. SIB들은 망내에 분산되어 있는 기능모듈인 FE(functional entity)들과 이들간의 상호관계에 의하여 구성된다. FE들의 관계를 보여주는 평면이 바로 분산기능평면이며, FE 간의 상호관계는 물리평면에서의 프로토콜에 의하여 실제로 구현된다. 그림 1에서 나타난 것이 물리평면이며 SCP, SDP, IP, SSP 등과 같은 PE(physical entity)와 이들간에 프로토콜을 규정한다.

본연구에서 다루고 있는 기본호처리 SIB는 총괄기능평면에서 정의되는 SIB 중에 하나이며, 이 SIB는 분산기능평면에서 정의되는 FE 중에서 SSP에 위치한 SSF/CCF와 SCP에 위치한 SCF들의 상호작용에 의하여 구성된다. 이를 그림으로 나타내면 다음 그림 2와 같다.

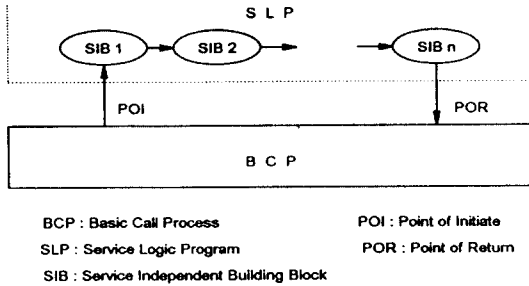


그림 2. 기본호처리 SIB
 Figure 2. Basic call process SIB

그러므로 본 연구에서는 기본호처리 SIB의 기능을 규정하고 이를 분산처리하기 위해서 필요한 SSF(service switching function)/CCF(call control function)와 SCF(service control function) 간의 상호작용을 구체화하는 프로토콜을 모델링하고 이를 검증한다.

III. 기본호처리 기능 및 시나리오

기본호처리 SIB는 호접속, 호질단, 호유지등의 기본호접속능력을 제공하는 특별한 SIB라 할 수 있다. 지능망에서의 서비스요구는 일반적인 호접착상에서 발생하므로 기본호처리중에 발생하게 된다. 일단 지능망서비스요구가 발생하게되면 CCF/SSF와 SCF의 인터페이스를 통하여 SCF의 서비스로직에게로 제어권을 넘겨준다. 그리고나면 서비스로직에 의하여 SIB 재인이 수행되므로써 요구된 지능망서비스가 분산처리되고, 이와동시에 기본호처리 SIB는 서비스를 요구한 호의 상태를 유지시켜주는 역할을 한다. 서비스로직에 의한 서비스처리가 모두 끝나게되면 인터페이스를 통하여 다시 제어권이 기본호처리 SIB로 넘어오게되고, 일반적인 호처리가 다시 재개된다. 이와같은 과정상에서 보면 기본호처리 SIB와 서비스로직사이에서는 두번의 인터페이스가 발생하게되는데 이를 POI(point of initiation), POR(point of return)이라한다. 즉, POI는 서비스 착수점이며, POR은 귀환점으로 인식될 수 있다. 이들간의 관계는 다음그림 3 과 같다.

이상에서 기술한 POI, POR의 기능은 결과적으로 일반호처리와 지능망서비스제어로직을 분리시키는

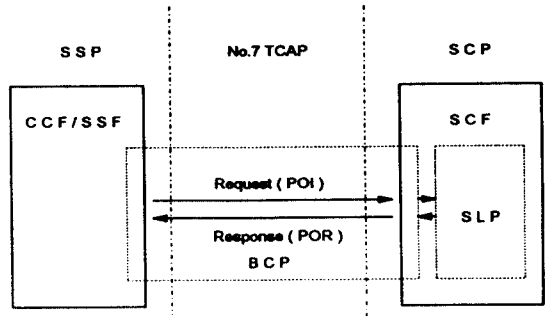


그림 3. 기본호처리 SIB와 서비스로직 사이의 인터페이스
 Figure 3. Interface between BCP SIB and service logic

역할을 한다. 이렇게 분리하므로써 다양한 서비스로직의 구성이 지능망교환기와는 독립적으로 이루어질 수 있고, 따라서 교환기는 단지 기본적인 호처리만을 수행하는 결과를 갖는다.

유연한 POI, POR 기능을 제공하기 위하여 고도지능망에서는 아밍(arming) 메카니즘을 이용하고 있다. 지능망서비스로직의 관점에서 접근가능한 사건들이 있는데 검출점(detection point)은 이 사건들을 감지하고 제어전달을 할 수 있는 점이다. 검출점은 일반적인 호처리를 담당하는 CCF의 기본호상태모델 상에서 정의되므로 일반적인 호처리 도중에서 지능망서비스에 대한 요구가 검출된다. 검출점에서 사건의 발생을 서비스로직에게 통보할 수 있기 위해서는 검출점이 아밍되어있어야 하며, 아밍되어있지 않으면 CCF는 SCF와는 무관하게 호처리를 계속한다.

특정 검출점에서 서비스요구가 검출되고 SSF가 SCF에게 트랜잭션을 요구할 때 SCF의 응답이 호처리에 영향을 줄 수 있을 때를 제어관계가 성립하였다고 하며, SCF의 응답이 호처리에 영향을 줄 수 없을 때를 비제어관계라고 한다. SSF/CCF와 SCF 사이의 제어관계는 아밍 메카니즘에 의하여 특징지워진다. 그러므로 프로토콜의 모델링도 이 아밍메카니즘에 의하여 결정된다. 아밍에는 정적아밍과 동적아밍이 있다. 정적아밍은 지능망이 초기화될때 이미 결정되어지는 아밍이며, 동적아밍은 지능망서비스도중 SCF의 요구에 의하여 아밍된다. 정적아밍을 트리거 아밍(trigger arming) 이라고하고 동적아밍을 이벤트 아밍(event arming)이라 한다. 그리고 아밍은 다시 SCF에게 제어를 요구하는가의 여부에 따라 요구하는 경우는 request, 그렇지않은 경우에는 notifi-

cation으로 나눈다. 아밍메카니즘을 다음 표에 나타내었다.

표 1. 검출점 유형

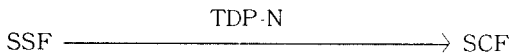
Table 1. Detection Point Type

유형	아밍형태	관계	지연
TDP-N	static	비제어	X
TDP-R	static	제어	O
EDP-N	dynamic	비제어	X
EDP-R	dynamic	제어	O

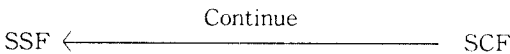
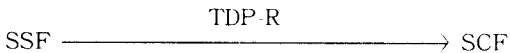
T : Trigger
E : Event
DP: Detection Point
N : Notification
R : Request

아밍메카니즘을 고려하여 발생가능한 POI, POR 시나리오를 구성해 보면 다음과 같다.

1) 기본호처리 도중 TDP-N이 검출되었을 경우: 이때에는 SCF에게 트리기가 검출되었다는 정보만을 전송하고 진행중이던 호처리는 계속된다.

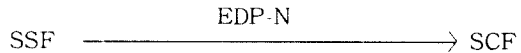
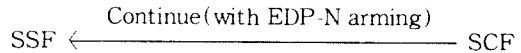
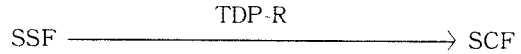


2) 기본호처리중 TDP-R이 검출되었고, SCF의 응답에 이벤트아밍 정보가 없이 단순히 호처리가 재개되었을 경우: 이때에는 트리기가 검출되었다는 정보를 SCF에게 전송한 후에 진행중인 호처리는 정지되며 SCF로부터 POR이 전송될 때에 호처리는 재개된다.

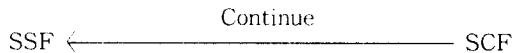
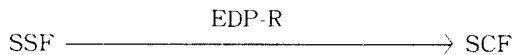
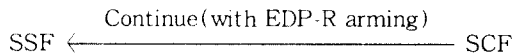
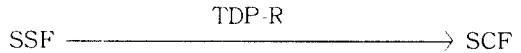


3) 기본호처리중 TDP-R이 검출되었고, 이벤트아밍을 요구하면서 호처리가 재개되었을 경우: 2)의 경우와 동일하게 처리되며 단지 다른점은 SCF로부터 POR이 전송되어 호처리가 재개될때 해당 호에 이벤트를 아밍시키는 것이다. 그러므로 재개된 호처리도중에 아밍된 이벤트가 발생하면 아밍의 특성에 따라 EDP-N일 경우에는 단순히 이벤트발생을 알리고 호처리는 계속되며, EDP-R일 경우에는 제어권을 SCF에게 넘겨주고 호처리는 정지된다.

a) 아밍이 EDP-N 일때

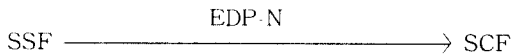
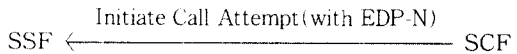


b) 아밍이 EDP-R 일때

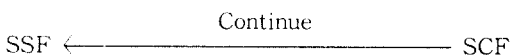
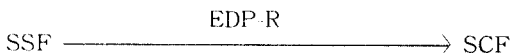
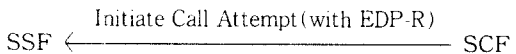


4) POI, POR 이외에 기본호처리 SIB가 제공하는 기능으로는 SCF의 요구에 의하여 새로운 호를 시작하게 하는 기능이 있는데 아밍메카니즘을 고려하여 시나리오를 구성하면 다음과 같다.

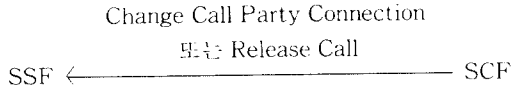
a) EDP-N 아밍을 갖는 호시도 경우



b) EDP-R 아밍을 갖는 호시도 경우



5) 이상의 경우외에 기존의 호의 연결관계를 수정하는 기능, 기존호를 제거하는 기능등을 제공해야 한다. 이러한 SCF의 요구에 의하여 시작되는 기본호처리 기능을 위한 인터페이스는 SSF와 SCF 사이에 제어관계를 갖지 않고 단순히 기능을 요구하므로써 이루어지므로 다음의 시나리오처럼 단순하다.



이상과 같은 정보흐름은 제어관계의 관점에서 보면 같은 의미를 갖는 3가지 형태의 정보흐름군으로 나눌 수 있다. 첫번째가 1), 5)와 같은 시나리오에 해당하는 제어관계가 성립하지 않는 경우이고, 두번째는 3)과 같은 시나리오에 해당하는 SSF에 의하여 제어관계가 성립하는 경우, 마지막으로 4)와 같은 SCF에 의하여 제어관계가 성립하는 경우이다. 이렇게 나뉘어진 3가지 경우에 대하여 4장에서 모델링하였다.

IV. 모델링 및 검증

4.1 페트리네트(Petri Net)

페트리네트는 시스템을 구현하기 이전에 정보나 프로토콜의 흐름을 분석할 수 있고 다양한 시스템에 적용이 가능한 모식적, 수학적 모델링도구로 1962년 Carl Adam Petri에 의하여 제안되었다.(6,7) 페트리네트는 동시에 수행가능한 실행체들끼리의 상호작용을 나타내는 분산시스템, 통신프로토콜 등의 분야에서 주로 이용되고 있다. 이 모델의 분석은 시스템에 대한 도달성, 안정성, 생존성, 데드락 등의 특성을 파악할 수 있으며, 분석방법으로는 도달성트리에 의한 방법과 행렬방정식에 의한 방법이 있다.(6,7) 본 연구에서는 도달성트리에 의한 방법으로 모델링 검증을 수행하였다.

페트리네트의 도달성트리를 이용하여 분석할 수 있는 특성은 다음과 같다.

1) 도달성(reachability): 페트리네트의 초기상태 M0에서 임의의 상태 M으로 변환시키는 정당한 집화순서(firing sequence)가 존재하면 상태 M은 초기상태 M0로부터 도달가능하다고 한다.

2) 생존성(liveness): 초기상태 M0에서 정당한 집화순서에 의해서 각 전이가 모두 집화 가능하다. 초기상태로부터 도달 가능한 모든 상태가 1개 이상의

점화가능한 전이를 갖는다면 이를 생존적이라고 한다. 이는 데드락과 관련이 있다.

3) 제한성(boundedness): 초기상태 M0로부터 도달 가능한 모든 상태에 대하여 각 place가 최대 K개의 토큰을 갖도록 하는 유이 아닌 정수 K가 존재하면 이 페트리네트를 제한되었다고 하며, 특히 K=1 일때 안전(safe)하다고 한다.

4.2 기본호처리를 위한 프로토콜에 대한 모델링

경우 1) 제어관계가 필요하지 않은 경우

이 경우는 제어관계가 성립되지 않으므로 단순히 한변의 정보흐름으로 완료된다. 그러므로 SSF, SCF에서 제어관계가 성립하지 않고 따라서 프로토콜 정보 유지를 위한 인스턴스의 상태는 Idle 뿐이다. 3장에서 기술한 시나리오 1)에 대한 페트리네트 모델은 다음 그림 4와 같다. 3장에서 기술한 시나리오 5)에 대한 모델은 SCF, SSF가 서로 바뀌고 전송되는 메시지만 다를뿐 제어관계 관점에서 동일하므로 모델링을 생략한다.

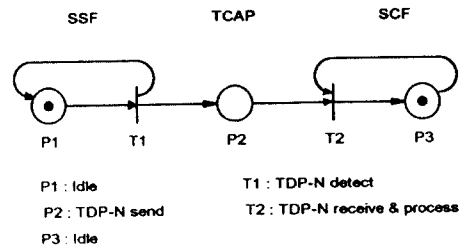
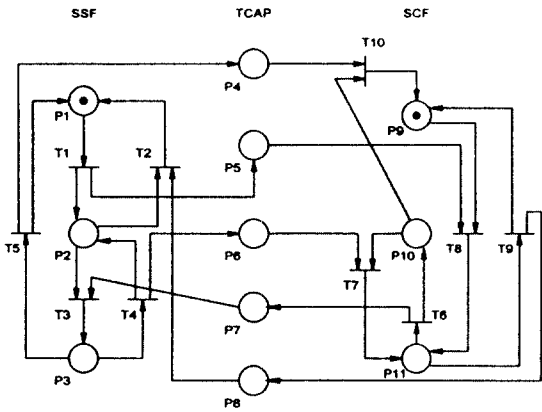


그림 4. 경우 1에 대한 페트리네트 모델
Figure 4. Petri Net Model for case 1

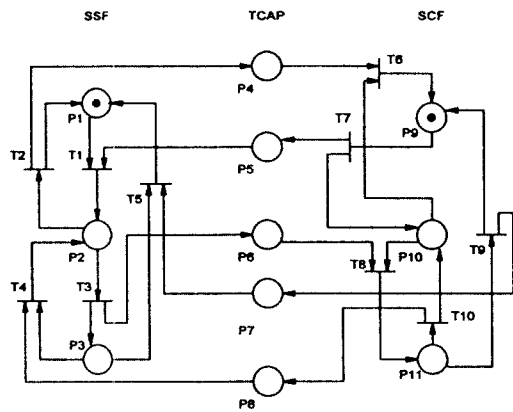
경우 2) SSF 요구에 의하여 제어관계가 시작되는 경우

3장에서 기술한 시나리오 3)에 대한 모델은 다음 그림 5와 같다. 제어관계가 시작되면 SSF는 active 상태로 전이하고 SCF로부터 야밍이 있는 호재개 메시지(continue 등)를 받으면 monitor 상태로 전이한다. monitor 상태에서는 야밍된 이벤트가 발생하는가를 모니터링하는 상태로 이벤트가 검출되면 이를 SCF에게 알리고 자신은 idle 상태로 전이한다. 한편 SCF는 SSF로부터 서비스요구를 받으면 서비스로직을 공동시켜서 지능망서비스를 수행하는 active 상태로 가며 이벤트를 야밍한 때에는 이를 기다리는 wait 상



- P1 : Idle
- P2 : Active
- P3 : Monitor
- P4 : EDP-N send
- P5 : TDP-R send
- P6 : EDP-R send
- P7 : POR (with EDP) send
- P8 : POR (no EDP) send
- P9 : Idle
- P10 : Active
- P11 : Wait
- T1 : TDP-R detect
- T2 : POR(no EDP) receive & process
- T3 : POR(with EDP) receive & process
- T4 : EDP-N detect
- T5 : EDP-N detect
- T6 : POR(with EDP) send
- T7 : EDP-R receive & process
- T8 : TDP-R receive & process
- T9 : POR(no EDP) send
- T10 : EDP-N receive & process

그림 5. 경우 2에 대한 페트리네트 모델
Figure 5. Petri Net Model for case 2



- P1 : Idle
- P2 : Monitor
- P3 : Active
- P4 : EDP-N send
- P5 : Initial Call (with EDP)
- P6 : EDP-R send
- P7 : POR (no EDP) send
- P8 : POR (with EDP) send
- P9 : Idle
- P10 : Wait
- P11 : Active
- T1 : EDP arming
- T2 : EDP-N detect
- T3 : EDP-R detect
- T4 : POR(with EDP) receive & process
- T5 : POR(no EDP) send
- T6 : EDP-N receive & process
- T7 : Initial Call (with EDP)
- T8 : EDP-R receive & process
- T9 : POR(no EDP) send
- T10 : POR(with EDP) send

그림 6. 경우 3에 대한 페트리네트 모델
Figure 6. Petri Net Model for case 3

태로 천이한다.

경우 3) SCF 요구에 의하여 제어관계가 시작되는 경우

3장에서 기술한 시나리오 4)에 대한 모델은 다음 그림 6과 같다. 이때에는 SCF가 호제어를 먼저 요구하면서 이벤트를 야밍하는 경우이다. 이벤트야밍을 요구하고나면 wait 상태로 천이하고 SSF로부터 EDP-R 발생이 통보되면 이를 처리하기 위한 active 상태로 간다. 그러나 EDP-N이 통보되면 제어관계가 필요없으므로 idle 상태로 천이한다. 한편 SSF는 이벤트야밍 요구를 SCF로부터 받으면 이벤트를 모니터하기 위하여 monitor 상태로 천이하고, 이벤트가 검출되면 EDP-N 일때에는 제어관계가 필요없으므로 idle 상태로 천이하고 EDP-R 일때는 제어관계가 시작되므로 active 상태로 천이한다.

4.3 도달성 트리에 의한 검증

모델링된 페트리네트의 동적 특성을 분석하기 위하여 도달성트리를 구해야 한다. 일반적으로 도달성

을 분석하기 위한 알고리즘(8)은 다음과 같이 의사코드로 나타낼 수 있다.

```

start()
{
    W = { initial state }
    A = { }
    analyze() ;
}
analyze() /* full search */
{
    if (W is empty)
        return ;
    q = element from W ;
    add q to A ;
    if (q == error_state)
        report error ;
    else {
        for each successor state s of q
            if (s is not in A or W) {
                add s to W ;
                analyze() ;
            }
    }
    delete q from W ;
}
    
```

이상의 알고리즘을 적용하여 구해지는 A가 발생가능한 모든 상태들이고, 각 상태가 A에 포함되는 순서를 트리로 구성하면 도달성트리를 생성할 수 있다. 모델링된 3 가지 케트리네트에 대한 도달성트리를 구하면, 다음에 나타낸 그림 7, 그림 8, 그림 9와 같다.

M0 : (1, 0, 1)
M1 : (1, 1, 1)
M1' : (1, w, 1)

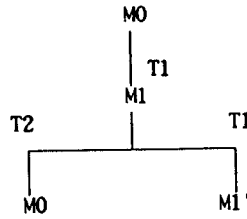


그림 7. 경우 1 모델링에 대한 도달성 트리
Figure 7. Reachability tree for case 1 modeling.

M0 : (1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0)
M1 : (0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0)
M2 : (0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0)
M3 : (0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1)
M4 : (0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0)
M5 : (0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)
M6 : (0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1)
M7 : (1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1)
M8 : (0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1)

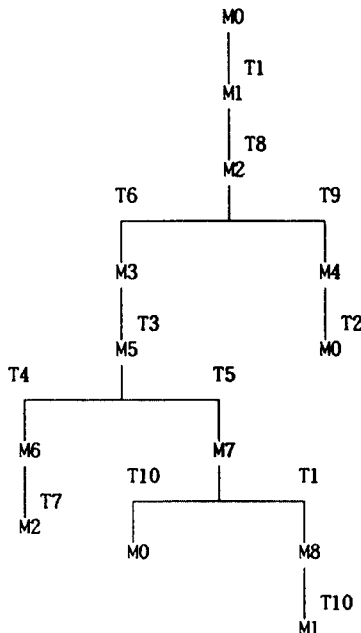


그림 8. 경우 2 모델링에 대한 도달성 트리
Figure 8. Reachability tree for case 2 modeling.

M0 : (1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0)
M1 : (1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0)
M2 : (0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0)
M3 : (1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0)
M4 : (0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0)
M5 : (0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)
M6 : (0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0)
M7 : (0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0)
M8 : (0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0)

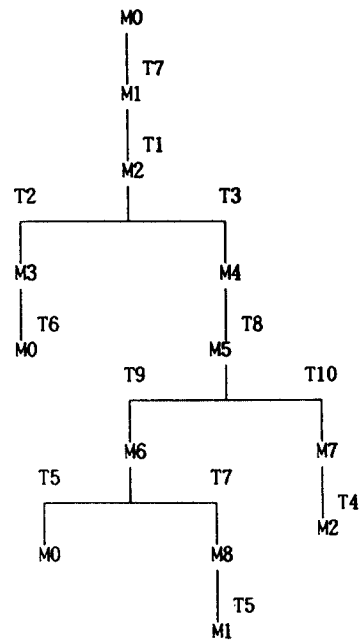


그림 9. 경우 3 모델링에 대한 도달성 트리
Figure 9. Reachability tree for case 3 modeling.

그림 7을 보면 초기상태 M0로부터 천이를 시작하여 M1를 거쳐 다시 초기상태 M0로 천이하므로 데드락은 없다. 그러나 본 모델은 S2에 토큰이 2개 이상 (M1')일 수 있으므로 안정성(safeness)를 보장할 수 없다. 실제 프로토콜이 구현되면 천이 T1과 T2에 대한 천이율은 평형을 이루게 되고, 전송채널에는 적절한 버퍼가 존재하므로 안정성은 보장못하지만 제한성(boundness)는 보장할 수 있다.

그림 8의 모델에서 모든상태는 순차적인 천이에 의하여 초기상태로 되돌아오므로 데드락은 없고, 각 place에 토큰의 수가 1개를 넘지 않으므로 안정성을

보장할 수 있다. 또한 다음과 같은 천이로 3 장에서 고려한 시나리오를 모델이 포함하고 있음을 알 수 있다.

시나리오 2) : M0 T1 M1 T8 M2 T9 M4 T2 M0

시나리오 3) -a) : M0 T1 M1 T8 M2 T6 M3 T3 M5
T5 M7 T10 M0

시나리오 3) -b) : M0 T1 M1 T8 M2 T6 M3 T3 M5
T4 M6 T7 M2 T9 M4 T2 M0

그림 9 모델도 그림 8과 같이 데드록이나 안정성에 문제가 없으며 다음과 같은 천이로 3 장에서 고려한 시나리오를 모델이 잘 나타내고 있음을 알 수 있다.

시나리오 4) -a) : M0 T7 M1 T1 M2 T2 M3 T6 M0

시나리오 4) -b) : M0 T7 M1 T1 M2 T3 M4 T8 M5
T9 M6 T5 M0

V. 결 론

가까운 미래에 실현될 고도지능망의 구축에 맞추어, 통신망 내에서 분산처리되는 기능블럭인 SIB에 대한 연구가 시급한 실정이다. 본 연구는 CCITT에서 권고하고 있는 개념모델을 기준으로하여 SIB 기능블럭중에서 가장 근간이 되는 기본호처리 SIB 구현을 위한 SSF와 SCF 간의 프로토콜을 모델링하고, 페트리네트를 이용하여 이를 검증하였다. 프로토콜은 SSF와 SCF 사이의 분산처리를 위한 정보흐름을 분석하여, 이들을 모두 포함할 수 있는 제어관계를 3 가지 형태로 나누어 모델링하였다.

이 프로토콜을 검증하기 위하여 형식화 모델인 페트리네트를 이용하여 모델링하였고, 각 모델에 대한 도달성트리를 작성하여 분석한 결과는 모델링된 프

로토콜이 고려된 시나리오들을 모두 만족하고, 데드록과 제한성등의 특성에 문제가 없음을 알 수 있었다.

앞으로의 연구방향은 개념적으로 이루어진 본 모델을 참조하여 모델링된 프로토콜을 구현하는 방법에 대한 연구가 뒤따라야 할 것이다.

참 고 문 헌

1. 홍진표, "지능망 기술동향," 전자공학회지 Vol. 18 No. 1, 1991. 1
2. 김덕진, 광대역 ISDN과 지능망을 위한 신호방식 연구, 고려대학교 부설 정보통신기술 공동 연구소 연구보고서, 1992. 3
3. 한국전자통신연구소 지능망연구부, 차세대지능망 개념서, 1991. 12
4. CCITT Draft Recommendations(Q.1200, Q.1201, Q.1202, Q.1203, Q.1204, Q.1205, Q.1208, Q.1211, Q.1213, Q.1214, Q.1215, Q.1218, Q.1219, Q.1290, Q1400), March, 1992.
5. 김덕진, AIN을 위한 교환기구조에 관한 연구, 고려대학교 부설 정보통신기술 공동연구소 연구보고서, 1993. 4
6. Peterson, Petri Net Theory and the Modelling of Systems, 1981
7. Petri Nets : Properties, Analysis and Applications, Proceedings of the IEEE, VOL. 77, NO. 4, April 1989.
8. Gerard J. Holzmann, Design and Validation of Computer Protocols, Prentice-Hall, 1991



趙顯俊(Hyun Joon Cho) 正會員

1963年 1月 28日生

1985년 2월 : 고려대학교 전자공학과 졸업(공학사)

1987년 8월 : 고려대학교 대학원 전자공학과 졸업(공학석사)

1991년 9월 ~ 현재 : 고려대학교 대학원 전자공학과 박사과정

1989년 1월 ~ 1991년 8월 : (주)쌍용컴퓨터 시스템연구소 연구원

※주관심분야: 프로토콜, 지능망, 광대역 ISDN 등 임



李成根(Sung Keun Lee) 正會員

1962年 8月 23日生

1985년 2월 : 고려대학교 전자공학과 졸업(공학사)

1987년 2월 : 고려대학교 대학원 전자공학과 졸업(공학석사)

1987년 3월 ~ 현재 : 고려대학교 대학원 전자공학과 박사과정

1987년 2월 ~ 1992년 9월 : 삼성전자 정보통신연구소 근무

※주관심분야: 광대역 ISDN, 지능망, 컴퓨터 네트워크 등 임

金惠鎭(Duck Jin Kim)

正會員

第 18 卷 第 7 號 參照

현재 : 고려대학교 전자공학과 교수