論文 98-23-9-34

B-ISDN 사용자-망 호/파티제어 신호 프로토콜의 동작 모델링 및 검증

정회원 김 원 순*, 김 재 훈**, 이 석 기**, 이 은 주***, 오 창 석****

Behavior Modeling and Verification of Signaling Protocol of B-ISDN User-Network Call/Party Control

Weon-soon Kim*, Jae-hoon Kim**, Seog-ki Lee**, Eun-ju Lee***,
Chang-suk Oh**** *Regular Members*

요 약

본 논문에서는 프로토콜의 논리적 오류를 검출하기 위한 모델을 제시하고, 미국 벨연구소에서 개발한 프로토콜 검증 도구인 SPIN을 이용하여 시뮬레이션을 수행하였다. 시뮬레이션 대상 프로토콜은 초고속망 구축에 적용되는 HAN/B-ISDN 신호방식 프로토콜로서 지점 대 다지점 환경에서 멀티파티 제어 기능을 수행하는 HQ.2971 프로토콜이다. 시뮬레이션을 수행하여 대상 프로토콜이 가능한 입력에 대한 정의되지 않은 메시지 수신이 발생하지 않는 것을 통하여 정확성을 확인하였으며, 어떤 상태에서 다음 상태로의 상태 천이가 발생하고 최종적으로 초기 상태에 도달하는 과정을 메시지 순서도로 표시하고 도달성 그래프를 작성하여 생존성 및 도달 가능함을 보였다. 본 논문에서 제시된 시뮬레이션을 통한 프로토콜 검증 기법은 복잡한 통신 프로토콜을 직접 구현하기 전에 프로토콜의 올바른 동작 특성 및 서비스 제공능력을 확인하는데 효과적으로 사용될 수 있다.

ABSTRACT

In this paper, we present a protocol behavior model to detect logical errors and execute simulation using protocol validation tool developed by AT&T Bell Laboratory. We select target protocol as HQ.2971 variation of ITU-T version for multiparty call and party control in the point to multipoint service environment applied to the Korean Information Infrastructure. In this simulation, we confirm correctness property through tracing message sequence chart

* 로커스 정보통신연구소
** ETRI
*** 제주산업정보대학
**** 충북대학교 컴퓨터공학과
論文番號 : 98139-0326
接受日字 : 1998年 3月 26日

not happening inopportune messages for possible random event generation and finally show reachability graph tracing all reachable states of a model system. The proposed method can be effectively applied for verification of correctness and proper service providing feature without implementation in the field of complicated communication protocols.

I. 서 론

현재 국가적으로 추진되고 있는 HAN/B-ISDN 연구 사업에서는 멀티미디어를 제공하는 광대역 서비스를 제공하기 위한 서비스 및 장치 개발이 이루어지고 있다. 여기서 각 장치간의 정보 및 절차를 제어하기 위해서 프로토콜을 규격화하는데, 프로토콜의 설계단계에서 그 자체가 갖는 모순성이 있는가를 검증하는 것은 매우 중요하다. 따라서, 프로토콜의 규격작성 단계에서 프로토콜이 정상적으로 동작하는 지와 원하는 서비스를 상위 계층에게 제공하는지를 확인하는 것이 필요하다. 즉, 프로토콜의 설계 단계에서 그 자체가 갖는 모순성이 있는지를 검증하는 것은 매우 중요하다.

프로토콜 검증은 프로토콜의 논리적으로 정확한가를 조사하는 것이다. 따라서, 검증은 시스템 구현제품의 개발 전단계인 설계 단계에서 주로 이루어진다. 프로토콜 검증은 프로토콜의 정확성, 완전성, 일관성 등을 검사하는 것이며, 안전 특성(safety property), 생존 특성(liveness property), 제한성(boundedness), 완전성(completeness), 교착상태 없음(deadlock freeness), 차폐상태 없음(차폐상태 freeness), 종결 또는 진행(termination or progress), 전체적인 정확성(total correctness) 등과 같은 프로토콜 특성[1-6]을 검증한다.

본 논문에서는 HAN/B-ISDN 과거에 적용되는 호 및 파티 제어 프로토콜을 대상으로 각 특성에 대하여 기술하며, SPIN을 이용한 HQ.2931을 포함하는 HQ.2971에 대한 동작 및 제어 절차에 대한 모델링과 검증을 수행하고 그 결과를 분석하였다. 이를 위해 본 논문은 제2장에서 HQ.2971 프로토콜 특성을, 제3장에서는 프로토콜 검증도구인 SPIN, 제4장에서는 프로토콜 검증을 위한 시뮬레이션 및 결과 분석에 대하여 기술하며 제5장에서 결론을 맺었다.

II. HQ.2971 프로토콜 검증 특성

2.1 프로토콜 개요

HCS-1 규격 중에서 점 대 다중점 접속에 관한 규격이 HQ.2971[23]로서 다자간 화상회의 또는 상호협동 서비스 등 B-ISDN에서의 서비스를 기존의 HQ.2931[21,23]의 링크 상태에 파티 상태를 추가하여 파티 제어 기능을 처리한다. 점 대 다중점 접속은 루트와 한 리프 사이의 초기 연결 설정이 이루어진 후에 루트의 리프 추가 요구에 의하여 리프 파티들의 추가가 가능하게 된다. 그러나 호가 활성 상태인 경우에는 언제든지 루트에 의해 파티의 추가가 가능하며, 파티의 삭제는 루트 혹은 리프의 요구에 의해 가능하다. 이러한 다중 파티 제어를 위하여 점 대 점 프로토콜인 HQ.2931 프로토콜의 메시지에 파티 관련 메시지를 추가하고 메시지내에 종단점 참조 정보요소를 추가하여 파티 제어를 한다.

2.2 프로토콜 동작 절차

프로토콜을 모델링하기 위해 프로토콜의 상태 천이표를 활용하여 이벤트가 발생하였을 때 그에 따른 올바른 동작 및 상태 천이가 일어나는지를 HQ.2971 규격의 SDL을 중심으로 표기한다.

본 논문에 기술되어 있는 부분은 점 대 다중점 프로토콜 중에서 사용자측 파티 제어에 대한 부분을 EFSM(Extended Finite State Machine)으로 구성하였으며, 이를 메시지 순서도의 형태로 기술하였다.

B-ISDN에서는 점 대 다중점 파티 제어 프로토콜 절차는 신호방식 서비스를 기존의 HQ.2931의 상태에 추가하여 파티 상태를 두어 그림 2-1과 같이 파티 제어 기능을 처리한다.

점 대 다중점 접속은 그림 2-2과 같이 루트와 하나의 리프 사이의 초기 연결 설정시 이루어진 후에 루트의 리프 추가 요구에 의하여 리프 파티의 추가가 가능하다. 호가 활성 상태인 경우에는 언제든지 루트에 의하여 그림 2-3에 나타낸 바와 같이 파티의 추가 및 삭제가 가능하며, 파티의 삭제는 루트 혹은 리프의 요구에 의해 가능하다.

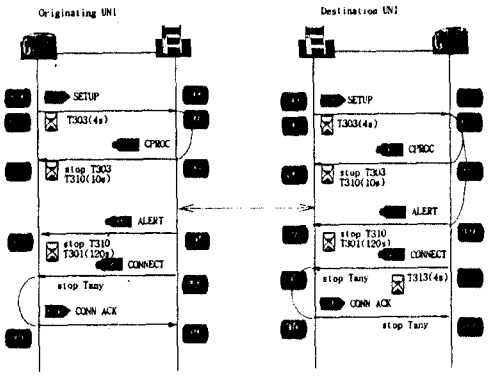

그림 2-1. HQ.2971 호 설정 절차

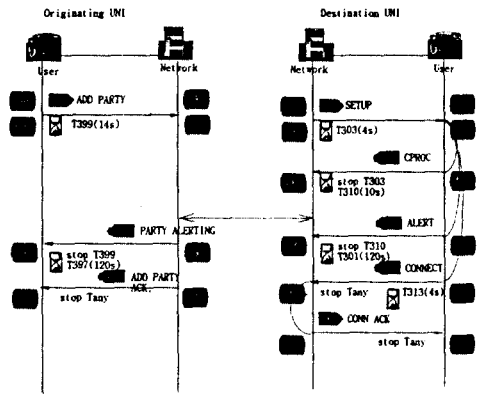

그림 2-2. HQ.2971 파티 추가 절차

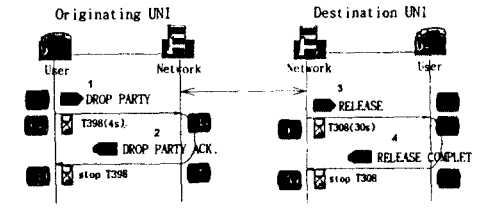

그림 2-3. HQ.2971 파티 삭제 절차

2.3 검증 대상

프로토콜의 정확성, 안전성, 일관성, 필연성 등의 특성을 확인 하기 위하여 다음과 같은 사항에 대하여 검증한다.

- 교착상태(deadlock) : 한 상태에서 다음의 어떤 상태로의 천이가 존재하지 않기 때문에 다음 행위를 할 수 없는 경우, 즉 그 상태에서 나가는 천이가 존재하지 않는다.
- 차폐상태(livelock) : 프로토콜 상태들의 부분집합 내에서 그 상태들만을 무한히 반복적으로 천이하는 경우로써 그 부분집합 이외의 다른 상태로의 천이가 존재하지 않는다.
- 도달성(reachability) : 프로토콜이 작동되기 시작할 때 즉, 프로토콜의 시작에서 정의되어지는 특별한 상태로써 초기 상태가 존재하는데, 이 초기 상태로부터 프로토콜은 정의된 천이의 순서에 의해 일부 또는 모든 다른 상태에 도달하게 된다. 프로토콜이 정의된 천이 순서에 의해 정의된 상태로 도달한다면 그 천이와 상태에 대해 올바른 프로토콜이다.
- 필연성(Liveness) : 프로토콜의 어떤 정당한 특성(상태 또는 행위)은 만족되어지고 도달되어져야 하는 상태와 반드시 발생해야 하는 행위를 나타낸다.

2.4 프로토콜 검증 기법

프로토코로 검증은 주어진 프로토콜이 어떠한 논리적인 특성을 만족하는지를 알아내는 과정이며, 이러한 논리적인 특성과 이 특성들을 검증하는 방법은 규격을 위해 사용된 모델에 따라 다르다. 프로토콜 설계에서 프로토콜이 주어진 특성들을 만족하는지에 대한 검증 방법은 여러 가지가 소개되어 있으며, 크게 도달성 분석 방법, 시간적 논리를 이용한 방법, 프로그래밍 언어를 이용한 방법 및 이들의 혼합 방법으로 분류되며[1-11], SPIN에서 사용하는 방법은 도달성 분석방법을 이용한다.

III. 프로토콜 검증 도구 : SPIN

SPIN[12-20]은 비동기 처리 시스템의 설계 및 검증을 지원하는 검증 도구로 통신시스템, 운영체제, 통신 프로토콜의 논리적 설계 오류를 검출하는데 많이 사용한다. SPIN은 미국 벨 연구소에서 개발되었으며, 검증을 위한 프로토콜은 모델링 언어인 SPIN은 미국 벨 연구소에서 개발되었으며, 검증을 위한 프로토콜은 모델링 언어인 PROMELA를 사용하여 프로토콜의

규격을 기술한다. SPIN의 구조는 그림 3-1과 같다.

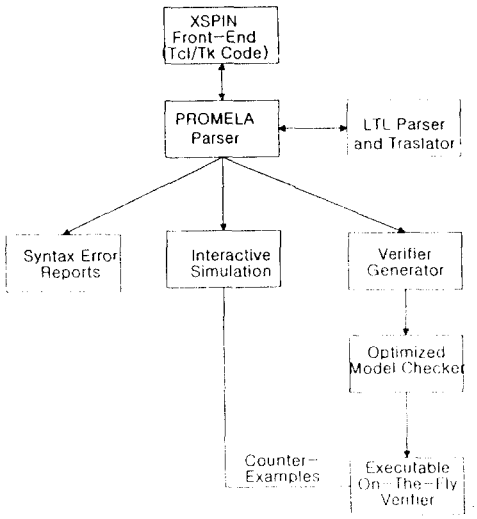

그림 3-1. SPIN 구조

3.1 PROMELA 언어

PROMELA는 동기와 비동기 통신을 모두 지원하고 Dijkstra 명령체계 언어 표시를 기본으로 하는 비결정적(Non-deterministic) 언어로써, 통신 시스템, 운영체제, 통신 프로토콜등과 같은 분산 시스템들의 논리적인 설계 오류를 검출하기 위한 검증 도구인 SPIN의 입력 언어로 개발하였다. 검증하고자 하는 시스템은 형식 언어인 PROMELA를 이용하여 기술하는데 이를 표현하기 위하여 PROMELA는 프로세스, 상태변수 및 메시지 채널을 제공한다.

- 상태변수 : PROMELA에서의 상태 변수는 전체 시스템에 대한 전역정보(global information) 또는 한 특정 프로세스의 지역정보(local information)를 저장하기 위하여 사용되며, 기본 데이터형으로 bit, bool, byte, short, int, chan을 제공한다.
- 프로세스 : 모든 프로세스는 proctype으로 정의되며, 프로세스가 수행하는 절차를 변수, 채널 및 선언문을 이용하여 기술한다. proctype으로 정의한 프로세와는 구분되는 init 프로세스가 있는데, 이는 proctype으로 정의된 프로세스들을 수행시킨다.
- 메시지 채널 : 메시지 채널은 한 프로세스에서 다른 프로세스로 데이터 전송을 기술하기 위하여 사용한다. 채널은 기본 데이터형으로 정의되어있다.

3.2 SPIN의 주요기능

PROMELA 언어를 사용하여 모델링한 내용을 SPIN의 시뮬레이터를 이용하여 모델의 이상 여부를 확인하며, 이를 위하여 각 프로세스의 수행상태, 각 변수 값의 변화, 타임 시퀀스 차트 및 메시지 시퀀스 차트를 제공한다. PROMELA로 모델링한 것에 이상이 없으면 검증을 위한 과정을 수행한다. 모델링한 내용을 이용하여 SPIN은 C 프로그램 파일을 생성한다. 생성된 프로그램 파일을 이용하여 모델링 하고자 하는 시스템의 크기가 작거나 중간 크기인 경우 전역 상태공간 방식을 이용하여 검증을 수행한다. 보다 큰 시스템에 대해서는 supertrace 방식을 이용하여 보다 적은 메모리를 가지고 수행하며, 검증 결과는 매우 우수하다.

IV. 프로토콜 시뮬레이션 및 결과 분석

4.1 PROMELA를 이용한 프로토콜 기술

가. 프로세스 구조

발신 사용자와 착신 사용자가 1:1로 연결하는 단순 호/연결 제어 프로토콜인 HQ.2931과 HQ.2971 프로토콜의 동작 및 데이터 흐름을 검증하기 위하여 HQ.2931과 HQ.2971 프로토콜의 모델링을 수행하였다. 본 논문에는 HQ.2971 프로토콜이 HQ.2931의 호 설정 기능을 반드시 이용해야 하기 때문에 HQ.2971 프로토콜을 대상으로 모델링과 검증 수행과정을 분석하였으며, HQ.2971 프로토콜의 호 설정 기능 및 파티 추가/삭제 기능을 모델링 하기 위하여 그림 4-1과 같이 프로세스를 구성하였다.

여기서 사용자측의 응용 프로세스는 Uapp로 모델링하고, 사용자측의 HQ.2971 프로토콜 엔티티로서 Upcc를 모델링하였다. 따라서 Upcc는 Uapp에게 호 기본 호/연결 제어 서비스를 제공한다. 기본 호의 설정을 위하여 Uapp0()/Uapp1() 프로세스와 기본 호 설정이 이루어진다. 이 과정은 HQ.2931에서의 호 설정 절차와 동일하다. 이때 망측의 응용 서비스로서 Napp를 모델링 하였으며, 망측의 HQ.2931 프로토콜 엔티티인 Npcc를 이용하여 Npcc0() 프로세스와 Npcc1() 프로세스가 수행되며, 여기서는 라우팅, 번호 번역, 전

달 채널 할당 등의 기능을 수행한다. 그리고 HQ.2971 프로토콜의 핵심기능인 파티 추가를 위하여 Upcc2() 프로세스와 Uapp2() 프로세스를 두어 추가되는 파티로서의 기능을 수행하도록 하였다. 한편 신호 링크를 통한 신호 메시지의 오류 없는 전달 기능을 제공하는 하위 계층은 안전하게 서비스를 제공하는 것을 가정하여 null 프로토콜로 처리한다. 그리고 처음 프로세스의 기동을 위하여 init 프로세스를 두는데, 이는 프로세스의 기동, 채널 초기화, 오류 메시지의 반송 등의 기능을 수행한다. 따라서 이를 이용하여 프로토콜의 정상적인 동작 절차뿐만 아니라 비정상적인 이벤트 발생시 처리절차도 검증할 수 있다.

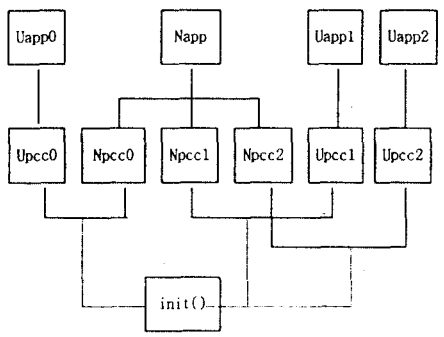

그림 4-1. HQ.2971 프로토콜의 프로세스 구조

나. 프로세스간 채널 구조

프로세스간 상호통신 채널을 구현하기 위하여 그림 4-2와 같이 채널과 각 프리미티브를 설정하였다.

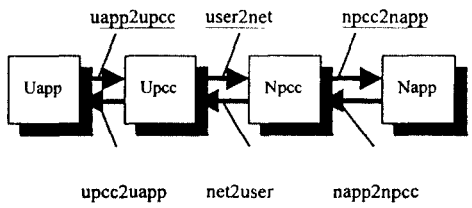

그림 4-2. HQ.2971 프로세스간 채널 구조

여기서 사용자측 내부 채널로 Uapp와 Upcc간에는 서비스의 명령 및 응답 기능을 수행하기 위하여 uapp2upcc 채널이 사용되고, 서비스의 표시 및 확인 기능을 수행하기 위해서 upcc2uapp 채널이 사용된다. 마찬가지로 망측 내부 채널로서 Napp와 Npcc간에는 npcc2napp와 napp2npcc 채널이 사용된다. 그리고 사용자와 망간 신호 링크를 통한 peer-to-peer 채널로는 Upcc에서 Npcc 메시지를 전달하기 위해 user2net 채널이 사용되고 Npcc에서 Upcc로 메시지를 전달하기 위해서 net2user 채널이 사용된다.

다. 동작 모델링

프로세스간의 동작 절차는 HQ.2971 SDL을 기준 모형으로 하여 SDL에 기술된 각 상태의 변화를 따라 동작 기능이 수행되도록 하였다.

여기서 "!" 부호는 메시지 송신을 "?" 부호는 메시지 수신을 나타낸다. 호 설정을 위한 사용자측 프로토콜 엔티티인 Upcc0() 프로세스의 동작 절차는 Null(U00) 상태에서 사용자측 응용 서비스인 Uapp()로 부터 호 설정 요구(setup_request)를 받음으로 해서 호 설정에 필요한 정보를 분석한 후 SETUP 메시지를 망측의 프로토콜 엔티티인 Npcc0()에게 전달하고 다음 상태인 Call Initiated(U01) 상태로 분기한다. 이러한 절차를 걸쳐 호 진행, 호출, 연결 메시지를 받고 호 설정

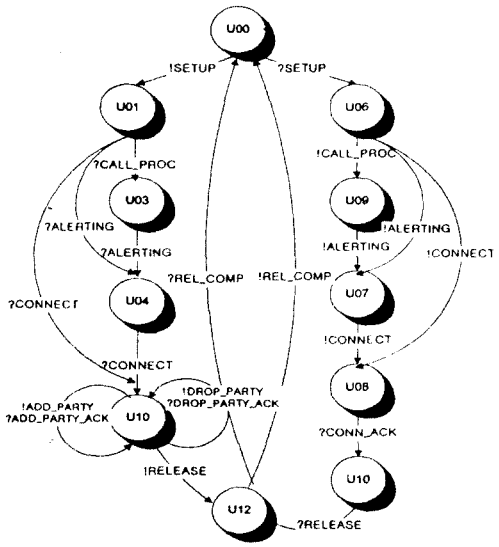

그림 4-3. Upcc() 프로세스의 동작 모델링

절차가 완료되면 Upcc0( ) 프로세스는 Active(U10) 상태에 있게 되며, 이때 연결된 호에 대한 해제 요청이 입력되면 이를 망측으로 보내어 망측으로부터 호 해제 완료 메시지(RELEASE COMPLETE) 메시지를 받고 다시 Null(U00) 상태로 되돌아 가게 된다.

망측의 프로토콜 엔티티인 Npcc0( )는 호가 설정되지 않은 Null(N00) 상태에서 Upcc0( )로 부터 SETUP 메시지를 받고 호 설정 요구를 수신한 후 Call Initiated (N01) 상태가 된다. 이러한 절차를 거쳐 호 설정이 완료되어 망이 수신자에게 호를 부여한 Active(N10) 상태에 도달하게 된다. 마찬가지로 호 해제 요청을 받은 사용자 측이 호 해제 요청을 보내면 다시 Null(N0) 상태로 되돌아가게 된다. 이러한 동작 절차는 SDL로 기술된 프로토콜의 동작 절차를 기준[14,16]으로 각 상태에서 정확한 메시지 전달 및 수신이 이루어지고 다음 상태로 천이 될 수 있도록 하였다.

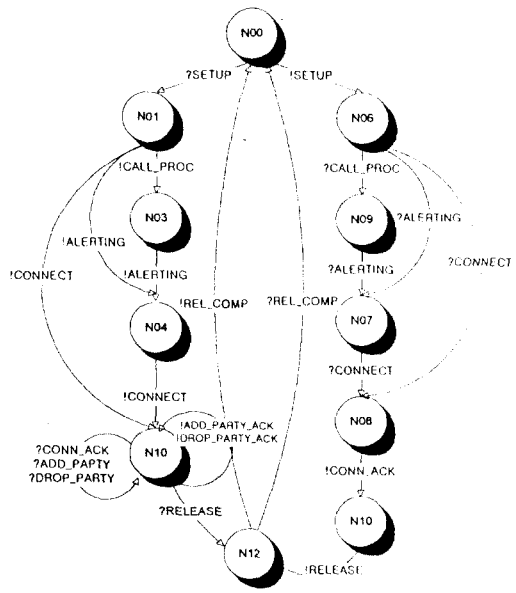

그림 4-4. Npcc() 프로세스의 동작 모델링

4.2 SPIN을 이용한 시뮬레이션 및 검증

가. 시뮬레이션

통신 프로세스 사이의 상호 동작을 검증하기 위하여 HQ.2971 프로토콜을 모델링 한 후 SPIN의 시뮬레이션 기능을 이용하여 프로세스들 사이의 신호 전달 및 메시지 전달기능을 확인하고, 검증기능을 이용하여 모델에 대한 검증을 수행하였다.

시뮬레이션은 구문상의 오류나 프로세스들 사이의 동작 및 메시지 전달이 올바르게 수행이 가능한지를 각각의 프로세스 동작 상태와 메시지를 주고 받는 과정을 문자 형식으로 기술된 시뮬레이션 결과나 시간 순서도(time sequence chart)로 보여주고 이를 그림으로 표현한 메시지 순서도(Message Squence Chart : MSC)로 출력한다.

본 논문에서 모델링한 HQ.2971 프로토콜의 시뮬레이션 수행 결과의 일부인 시뮬레이션 트레이스 결과를 그림 4-5에 나타내었다. 이를 통하여 각 채널을 통하여 메시지가 전달되는 과정을 프로세스의 수행 순서에 따라 확인해 볼 수 있다. 특히 전달되는 메시지의 내용을 확인하고 원하는 출력을 생성하며, 상태변수의 변화를 확인할 수 있다.

```
0:    proc-(:root:) creates proc  0 (:init:)
1:    proc  0 (:init:) creates proc  1 (uapp0)
1:    proc  0 (:init:) line 105 "pan_in" (state 11)
      [(run uapp0( ))]
2:    proc  0 (:init:) creates proc  2 (upcc0)
2:    proc  0 (:init:) line 107 "pan_in" (state 2)
      [(run upcc0( ))]
3:    proc  0 (:init:) creates proc  3 (npcc0)
3:    proc  0 (:init:) line 108 "pan_in" (state 3)
      [(run npcc0( ))]
4:    proc  0 (:init:) creates proc  4 (napp)
4:    proc  0 (:init:) line 109 "pan_in" (state 4)
      [(run napp( ))]
5:    proc  0 (:init:) creates proc  5 (npcc1)
5:    proc  0 (:init:) line 110 "pan_in" (state 5)
      [(run npcc1( ))]
6:    proc  0 (:init:) creates proc  6 (npcc2)
6:    proc  0 (:init:) line 111 "pan_in" (state 6)
      [(run npcc2( ))]
7:    proc  0 (:init:) creates proc  7 (upcc1)
```

그림 4-5. 시뮬레이션 결과

이 결과를 좀 더 쉽게 확인 할 수 있도록 프로세스 및 채널들 간의 메시지 전달과정을 그림으로 표현한 것이 메시지 순서도이다. 이를 그림 4-6에 나타내었다. 메시지 순서도는 각각의 메시지 전달 과정을 진행 순서에 따라 일련 번호(step number)로 표현할 수도 있고 이를 좀더 명확히 표현하기 위해 PROMELA로 기술된 문자를 그대로 보여주는 소스 텍스트 라벨(source

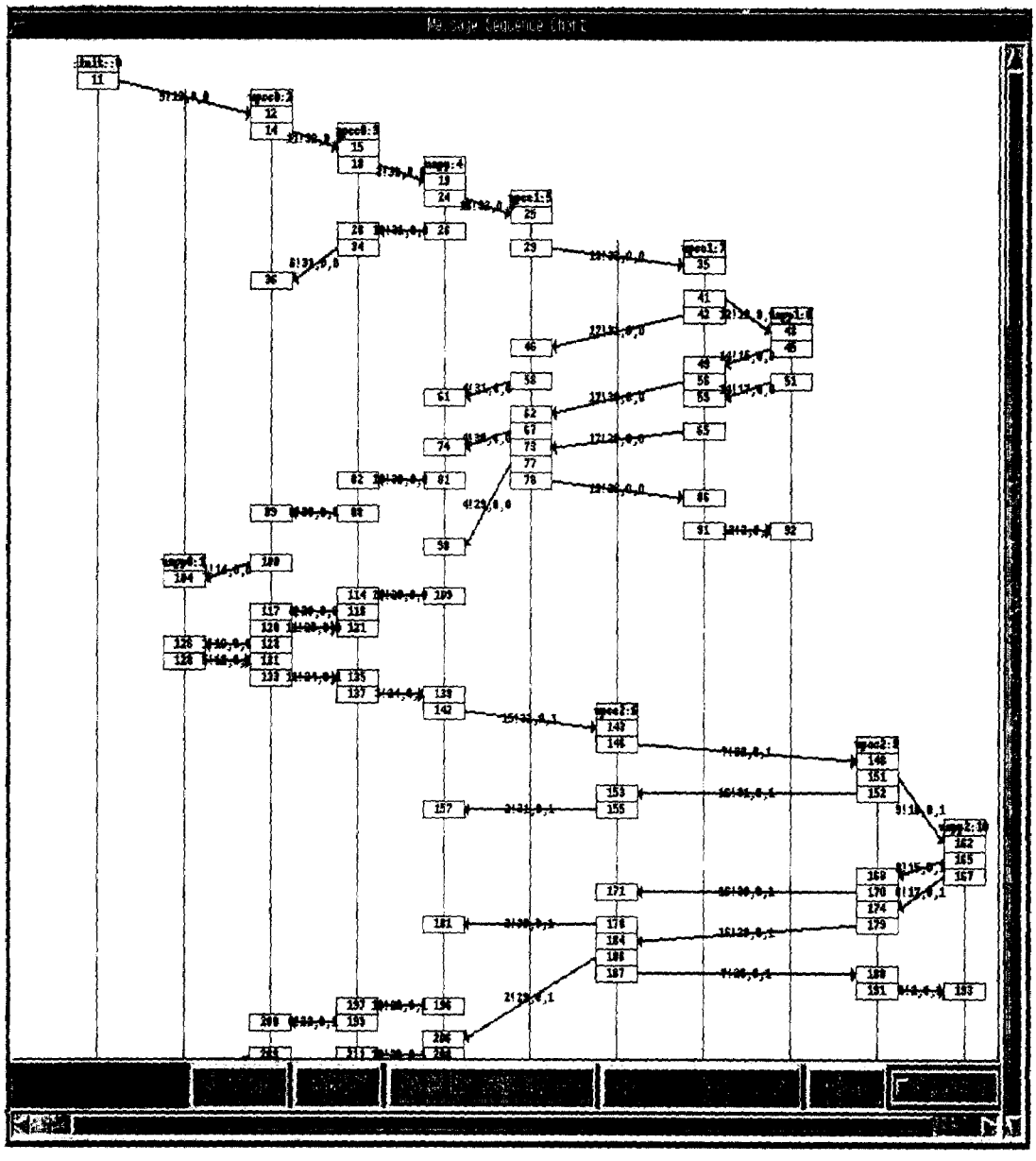

그림 4-6. 메시지 순서도

text label)의 두 가지 경우로 출력하여 볼 수 있다. 이렇게 표현되는 MSC를 통하여 실제 전달되는 정보의 내용과 채널들간의 메시지 전송에서 발생하는 오류를 확인할 수 있다. 즉 프로토콜 검증 파라미터 중 교착상태 발생여부를 눈으로 확인할 수 있으며, 원하지 않는 메시지의 수신이나 오류 없는 메시지의 전송이 가능한 정확성을 만족시킬 수 있다.

일반적으로 시뮬레이션 과정은 프로세스간 채널들의 동작과 메시지 전달에서의 오류 및 PROMELA 구현상의 구문 오류를 검사할 수 있다.

그림 4-6에 나타낸 시뮬레이션 결과는 시뮬레이션 수행 선택시 표시 형식은 MSC 패널에서 일련번호와 condenced Spacing을 선택하고 시뮬레이션 스타일은 랜덤 모드로 설정하고 시뮬레이션을 수행한 결과이다. 처음에 각 프로세스의 생성이 수행된 것이 나타났으며, 그 아래의 숫자는 각 상태에서의 메시지 및 신호가 전달될 때 단계번호를 의미한다. 각 단계번호에서는 메시지나 신호를 바으면 다음 상태로 천이하게 되는데 이때 전달되는 메시지들을 화살표를 통하여 표현하고 있다. !는 메시지 송신을 의미하고 ?는 메시지 수신을 나타내고 그 뒤에 표시되는 숫자는 메시지나 신호에 해당하는 번호와 호 식별을 위한 call_ref번호, 그리고 종단점 식별 정보를 위한 끝점(endpoint) 번호이다. MSC에서는 Uapp0()/Upcc0() 프로세스와 Uapp1()/Upcc1() 프로세스가 Napp()/Npcc() 프로세스를 통하여 프리미티브 및 메시지 전달을 통하여 호가 설정되고 파티가 추가된 결과인 Upcc2()와 Uapp2()프로세스의 동작상태를 확인할 수 있었다. MSC를 통하여 살펴볼 때 본 논문에서 수행한 HQ.2971 프로토콜의 논리적인 오류나 메시지 전달 오류는 발생하지 않았으며, 차폐상태와 교착상태에 빠지지 않았음을 확인할 수 있다. 또한 호 설정이나 해제 요청을 받아 setup_req나 rel_req 프리미티브가 수행되어 이러한 요청을 수행한 후 최종적으로 어떤 상태에 도달하였는가를 확인할 수 있다.

나. 검 증

검증과정은 검증 명령을 통해 이루어지며 시뮬레이션 결과로 만들어지는 pan_in과 pan.c, pan.h 파일을 컴파일 하는 것으로 프로토콜의 안전성과 liveness 검사를 위한 non-progress cycles, 도달성 검사를 위한 report unreachable code와 assertion 검사를 할 수 있다.

본 논문에서는 HQ.2971 프로토콜의 검증 과정을 프로토콜의 safety와 unreachable code 분석, assertion 위반 사항을 검사하고 검증 수행의 형태로서 검증시 메모리를 절약할 수 있는 partial order 방법을 선택하였다. 또한 오류 발생시 더 이상 진행하지 않고 멈추도록 하고 탐색 모드는 exhaustive 모드를 주어 수행하였다. 그 결과를 그림 4-7에 나타내었다.

도달성 분석 방법(reachability analysis method)은 통신 프로토콜 개체들의 모든 가능한 상호 작용들을 조사한다. 이 방법은 상태 기반 모델을 분석하는데 가장

```
pan : invalid endstate (at depth 223)
pan : wrote pan_in.trail
(Spin Version 2.9.6 -- 20 March 1997)
Warning : Search not completed
        + Partial Order Reduction

Full statespace search for :
        never-claim         -(not selected)
        assertion violations  +
        cycle checks   -(disabled by-DSAFETY)
        invalid endstates     +

State-vector 1480 byte, depth reached 223, errrors: 1

215 states, stored
          0 states, matched
        215 transitions (=stored + matched)
          9 atomic steps
hash conflicts : 0 (resolved)
(max size 2^19 states)

2.916 memory usage (Mbyte)

real        0.2
user        0.0
sys         0.0
```

그림 4-7. HQ.2971 프로토콜 검증 결과

적합한 방법 중의 하나이다. 한 프로토콜 계층에 지역적인 프로세스들의 동시적 행위 상태를 탐색하여 도달성 그래프를 얻을 수 있다. 이 그래프에서 각 노드는 모든 지역 프로세스들의 결합된 상태인 전역 상태를 나타내고, 각 호는 지역 천이를 나타낸다. 그래프의 초기 상태에서 출발하여 초기 상태로부터 도달되는 모든 상태들의 경로를 탐색함으로써 프로세스들의 상호 작용을 조사한다. 교착상태와 예측되지 않은 상태로의 도달 여부를 검증하기 위하여 도달되는 각 노드들을 검사한다. 차폐상태, 채널 오버플로우 등과 같은 프로토콜의 일반 특성을 위해서는 그래프의 전체가 검사되어야 한다.

호 설정과 해제 파티 추가 파티 삭제에 대한 도달성 그래프를 통하여 도달성 분석을 수행한다.

가. 호 설정 및 해제

Init() 프로세스는 프로세스의 기동 및 채널을 초기화 하는 기능으로 전체적인 프로세스의 생성과 천이 과정을 볼 수 있다.

호 설정 과정은 Uapp0()/Upcc1()와 Uapp1()/Upcc1()이 Napp0()/Npcc0(), Npcc1() 프로세스를 통하여 메

시지를 주고 받는 과정을 통하여 이루어진다. 호 설정 요구는 어느쪽에서나 발생할 수 있으므로 Uapp0() 나 Uapp1() 어느쪽도 착신측 또는 발신측이 될 수 있으며, 호 해제 절차는 발신 및 착신측 어느곳에서나 먼저 시작하는 것이 가능하다.

그림 4-8에 Uapp0()가 발신측인 경우의 도달성 트리를 나타내었으며, 그림 4-9에 Uapp0()가 착신측인 경우의 도달성 트리를 나타내었다. 두 경우의 도달성 트리와 상태 경로 분석에서 볼 수 있듯이 각각의 상태에서 어떤 상태로 도달이 가능한, 즉 어떤 상태에서든 만족할 만한 상태로 도달이 가능함을 보였으며, 같은 경로를 가지고 계속 반복되는 경로가 없음을 보이고 있다. 또한 호 해제 절차 후 항상 초기상태로 되돌아옴을 확인할 수 있다.

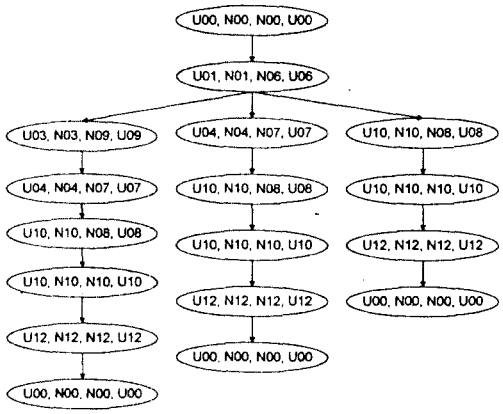

그림 4-8. Uapp0()가 발신측인 경우 도달성트리

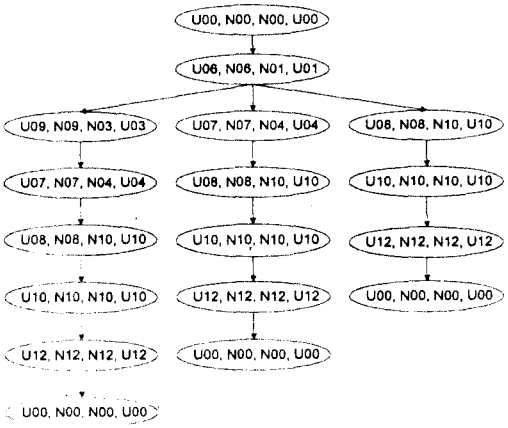

그림 4-9. Uapp0()가 착신측인 경우 도달성트리

나. 파티 추가 및 삭제

망측의 프로세스 Napp()는 npcc2napp 채널을 통하여 입력되는 종단점 정보를 통하여 호 설정 정보인지 파티 추가정보 인지 판단하여 다음 상태로 진행하여야 한다. 본 논문에서는 Uapp2()/Upcc2() 프로세스를 추가되는 파티로 정하여 파티 추가 기능을 수행하였다.

- 모든 메시지 처리의 경우(Upcc0, Npcc0, Npcc2, Upcc2)

(U10, N10, N00, U00) → (U10, N10, N06, U00) → (U10, N10, N06, U06) → (U10, N10, N06, U09) → (U10, N10, N09, U09) → (U10, N10, N09, U07) → (U10, N10, N07, U07) → (U10, N10, N07, U08) → (U10, N10, N08, U08) → (U10, N10, N10, U08) → (U10, N10, N10, U10)

- ADD_PARTY 메시지 수신 후 ALERT 메시지 처리의 경우

(U10, N10, N00, U00) → (U10, N10, N06, U00) → (U10, N10, N06, U06) → (U10, N10, N06, U07) → (U10, N10, N07, U07) → (U10, N10, N07, U08) → (U10, N10, N08, U08) → (U10, N10, N10, U08) → (U10, N10, N10, U10)

- ADD_PARTY 메시지 수신 후 CONNECT 메시지 처리의 경우

(U10, N10, N00, U00) → (U10, N10, N06, U00) → (U10, N10, N06, U06) → (U10, N10, N06, U08) → (U10, N10, N08, U08) → (U10, N10, N10, U08) → (U10, N10, N10, U10)

- 파티 해제 절차

(U10, N10, N10, U10) → (U10, N10, N12, U10) → (U10, N10, N12, U12) → (U10, N10, N00, U00)

다. 검증 결과

HQ.2971 프로토콜은 호 설정, 호 해제, 파티 추가, 파티 해제과정으로 구성되어 있다. 지금까지 각 프로

세스별로 생존성과 도달성, 정확성을 검증하였는데 생존성은 각각의 프로세스에 대한 생존성이 존재하는 것을 보이는 것도 중요하지만 전체에 대한 생존성이 존재하는 것을 보일 수 있어야 한다. 이를 위해 Uapp0() 프로세스와 Uapp1() 프로세스가 호 설정이 이루어진 상태에서 Uapp2() 프로세스가 파티 추가되는 시나리오를 가정하여 HQ.2971 프로토콜 전체에 대한 도달성 나무 그래프를 작성하였으며, 이를 분석하여 보면 서로 다른 경로를 갖는 주기가 존재하므로 각 프로세스에 대한 생존성과 더불어 전 과정에 대한 생존성이 있음을 알 수 있다. 또한, 교착상태와 같은 경로를 가지고 계속 반복되는 경로가 없으며, 호 해제 절차 후 모든 프로세스는 항상 초기상태로 되돌아옴을 확인할 수 있다. 이러한 결과에서 나타나듯이 HQ.2971 프로토콜의 동작 및 제어 절차가 오류 없이 기술되어 있음을 확인할 수 있다.

V. 결 론

본 논문에서는 SPIN을 이용하여 HQ.2971 프로토콜에 대한 모델링과 검증을 수행하였다. 사용자-망 인터페이스를 위해 7개의 프로세스를 모델링 하였으며, 각각의 프로세스들 사이의 상호 동작을 위해 12개의 채널을 이용하였다. 시뮬레이션 및 검증을 통하여 HQ.2971 프로토콜의 정확성, 안전성, 생존성 등을 검증할 수 있었다. SPIN의 이용결과 기존의 방법보다 간단한 방법으로 프로토콜을 모델링하고 검증이 수행될 수 있었으며, 현재 정상적인 동작 상태 만을 고려하여 검증을 수행하였으며, 향후 오류상황 처리 등을 고려한 검증이 이루어져야 하겠다.

HQ.2971 신호 프로토콜은 점 대 다중점 호 및 파티 제어를 규정한 것으로서 주로 동작 및 제어 절차가 오류없이 기술되어 있음을 확인하였다. 실제 통신 프로토콜의 특성에 맞게 동작 및 제어 절차를 중심으로 모델링하고 검증을 수행하였으므로 이러한 방법은 향후 다른 프로토콜 시스템을 검증할 때도 상당히 유용하게 활용될 수 있을 것이다.

참 고 문 헌

1. P. M. Merlin, "Specification and Validation of Protocols," IEEE Trans. Comm., Vol. COM-27, No. 11, pp.1671-1680, 1979.
2. A. S. Danthine, "Protocol Representation with Finite-State Models," IEEE Transaction on Communications, Vol. 28, No. 4, pp.638-643, 1980.
3. L. Lamport, "Specifying Concurrent Program Modules," ACM Transactions On Programming Languages and Systems, Vol. 5, No. 2, pp.1900-1922, 1983.
4. M. Uyar, A. Lapone, K. K. Sabnani, "Algorithmic Verification of ISDN Network Layer Protocol," AT&T Technical Journal, Vol. 69, No. 1, pp. 17-31, 1990.
5. G. J. Holzmann, "Algorithms for Automated Protocol Verification," AT&T Technical Journal, Vol. 69, No. 1, pp.32-44, 1990.
6. G. J. Holzmann, Design and Validation of Computer Protocols, Prentice-Hall, 1991.
7. F. J. Lin, "Two Application of PROMELA/SPIN," Proceedings of the SPIN Workshop, 1995.
8. H. E. Jensen, K. G. Larsen, "Modeling ans Analysis of a Collision Avoidance Protocol using SPIN and UPPAAL," Proceedings of the SPIN Workshop, 1996.
9. P. Merino, J. M. Troya, "Modelling and Verification of the ITU-T Multipoint Communication Service with SPIN," Proceedings of the SPIN Workshop, 1996.
10. S. Loffler, A. Serhrouchni, "Creating Implementations from PROMELA Models," Proceedings of the SPIN Workshop, 1996.
11. S. Leue, P. B. Ladkin, "Implementing and Verifying Scenario-Based Specifications using Promela/Xspin," Proceedings of the SPIN Workshop, 1996.
12. T. Nakatani, "Verification of Group Address Registration Protocol using PROMELA and SPIN," Proceedings of the SPIN Workshop, 1997.
13. B. Knaack, "Real-time Extension of SPIN-Ongoing Research," Proceedings of the SPIN Workshop, 1997.
14. ITU-T 권고 Q.2931, "B-ISDN DSS2 User Network Interface(UNI) Layer 3 Specification for

Basic Call/Connection Control," COM11-R 78-E, Geneva, Oct. 1994.
15. 이은주, 김원순, 이석기, 오창석, "SPIN을 이용한 HAN/B-ISDN의 HQ.2931 프로토콜 검증," 한국통신학회 추계 학술발표회, Vol. 16, No. 2, pp. 1023-1026, 1997.
16. "HAN/B-ISDN 접속표준 선행 규격," 한국통신 초고속관리단, 1995년 10월.

김 원 순(Weon Soon Kim) 정회원
1983년 2월 : 서울시립대학 전자공학과(학사)
1994년 8월 : 충남대학교 전자공학과(석사)
1998년 2월 : 충북대학교 컴퓨터공학과(박사 수료)
1983년 3월 ~ 1997년 10월 : 한국전자통신연구원(책임연구원)
1997년 10월 ~ 현재 : 로커스 정보통신연구 소장
※ 주관심분야 : ATM, 지능망, CTI, 프로토콜 공학

김 재 훈(Jae Hoon Kim) 정회원
1983년 2월 : 숭전대학교 전자계산기공학과(학사)
1993년 2월 : 한남대학교 전자계산기공학과(석사)
1983년 3월 ~ 현재 : 한국전자통신연구원(선임연구원)
※ 주관심분야 : 프로토콜 검증, 위성통신, 객체지향 언어, 객체지향설계방법론

이 은 주(Eun-Ju Lee) 정회원
1990년 2월 : 청주대학교 전자계산학과 졸업(학사)
1996년 2월 : 충북대학교 대학원 컴퓨터공학과 졸업(석사)
1998년 2월 : 충북대학교 대학원 컴퓨터공학과 박사 수료
1998년 2월 ~ 현재 : 제주산업정보대학 사무자동화과 전임강사
※ 주관심분야 : ATM/B-ISDN, 차세대 인터넷

이 석 기(Seog-Ki Lee) 정회원
1980년 2월 : 서강대학교 전자공학과(학사)
1982년 2월 : 서강대학원 전자공학과(공학석사)
1983년 9월 ~ 1988년 2월 : 현대전자산업(주) 시스템연구소
1988년 3월 ~ 현재 : 한국전자통신연구원(선임연구원)
※ 주관심분야 : B-ISDN Signalling, Network Protocol, Gigabit Ethernet

오 창 석(Chang Suk Oh) 정회원
현재 : 충북대학교 컴퓨터공학과 교수
한국통신학회 논문지 제21권 제6호 참조