SPIN을 이용한 차량용 MOST Network Service 프로토콜 스택 정형검증*

이동아O 윤상현, 이무열, 진현욱, 유준범
건국대학교 컴퓨터 공학부
ldalove@konkuk.ac.kr, pctkdgus@konkuk.ac.kr, zlemy@konkuk.ac.kr, jinh@konkuk.ac.kr, jbyoo@konkuk.ac.kr

Formal Verification of Protocol Stack for MOST Network Service using SPIN

Dong-Ah LeeO Sanghyun Yoon, Mu-Youl Lee, Hyun-Wook Jin, Junbeom Yoo
Department of Computer Science and Engineering, Konkuk University

1. 서론

최근 차량 내부에 다양한 멀티미디어 장비들이 장착됨에 따라, 이들 간의 원활한 통신을 지원할 수 있는 광대역 네트워크 시스템인 MOST (Media Oriented Systems Transport) 가 제안되었다. MOST는 네트워크 프로토콜 스택인 Network Service를 제공하고 있지만, 다양한 OS 플랫폼에 이식성이 높은 구조는 제안되지 않았다. 이러한 단점을 보안하기 위하여 POSIX (Portable Operating System Interface) 를 이용해 플랫폼에 상관없이 사용할 수 있도록 네트워크 프로토콜 스택을 정의한 u-OMNiPro (user-level Open Most Network service Protocol Stacks) 1.0 [1] 이 개발되었다. u-OMNiPro는 제공되는 기능의 정확성과 안전성이 높은 수준으로 확보되어야만, 기존의 MOST Network Service 의 대체 기술로 사용될 수 있다. 본 연구에서는 높은 수준의 정확성과 안전성을 보장하기 위하여 PROMELA (Protocol Meta Language) 와 SPIN [2] 을 이용한 정형명세 및 정형검증 기법을 u-OMNiPro 1.0 에 적용하였다.

2. 본론

u-OMNiPro 를 정형검증하기 위하여 개발자와의 긴밀한 협의를 통해, 정형검증이 필요한 핵심 모듈과 복잡한 모듈을 선별하였다. 또한, 이들을 두 개의 독립적인 PROMELA 모델 (FBlock Framework, Message Core) 로 정형명세한 후, 다양한 검증 속성을 대상으로 SPIN 을 이용한 정형검증을 수행하였다.

FBlock Framework 계층의 FBlock 등록 및 해지에 관한 검증을 하기 위하여 그림 1과 같은 모델을 구현하였다. Register 와 Unregister 프로세스는 임의의 시간에 임의의 정보를 가진 FBlock 정보를 regTableThread 로 전송한다. 저장된 FBlock 에 대한 정보를 관리하는 regTableThread 로 각각 등록 및 해지 요청을 한다. regTableThread 에서 등록 및 해지에 대한 요청이 올바르게 처리되는지 확인하기 위하여 다음과 같은 속성들을 확인하였다.

- 현재 등록을 시도하려는 서비스 함수의 속성이 함수들을 관리하는 테이블에 올바르게 저장되었는가?
- 해지 신청이 요청됐을 경우, 현재 저장되어 있는 해당 함수가 올바르게 해지되었는가?

FBlock 을 등록하는 과정에서는 오류를 나타내지 않았지만 FBlock 을 해지하는 과정에서 발생할 수 있는 오류를 발견하였다. FBlock 을 해지하려 할 때, 요청이 들어온 FBlock 의 속성과 실제 저장된 테이블에서 해지되는 FBlock 의 속성이 동일한지 여부를 검증하는 과정에서, 이를 확인하기 위한 assert 문을 위반하여 프로세스가 강제 종료됐다.

* 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음. (NIPA-2010-C1090-0903-0004, NIPA-2010-C1090-1031-0003)

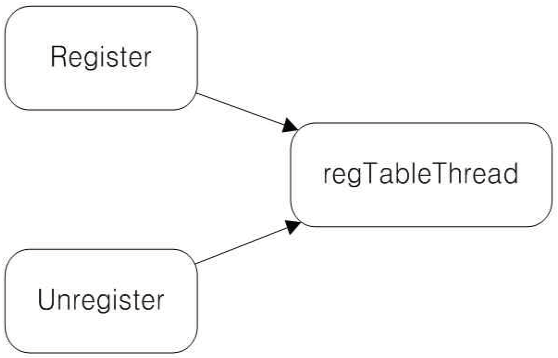

그림 1 FBlock Register & Unregister model

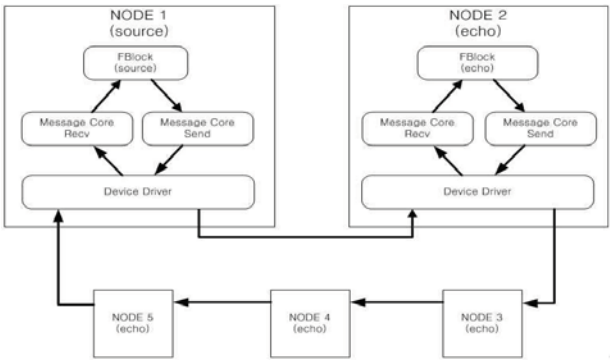

그림 2 u-OMNiPro echo model (5 nodes)

두 번째로 이론적으로 무한대의 데이터 전송이 가능하도록 설계된 u-OMNiPro 의 데이터 전송을 검증하기 위하여 그림 2와 같은 5 개의 노드를 링 형태로 구성한 모델을 설계하였다. 그림 2의 Message Core 는 상위 FBlock 으로부터 받은 데이터를 다른 노드로 송신할 때 크기에 따라 일정한 패킷으로 나누어 보내주는 역할을 한다. 또한 데이터를 수신할 때 나눠진 패킷을 하나로 합쳐 FBlock 으로 전달해 주는 역할을 한다. 위와 같은 기능이 오류 없이 처리되는지 검증하기 위하여 다음과 같은 속성을 확인하였다.

- echo NODE 로부터 되돌아온 메시지가 source NODE 에서 생성한 원본 메시지와 동일한가?
- 첫 번째 패킷이 전송되었다면 언젠가는 반드시 마지막 패킷이 보내지는가?

source 노드에서 생성한 데이터를 다른 노드로 전송할 경우, 송신한 데이터와 되돌아온 데이터의 크기를 비교했을 때 두 데이터의 크기가 다르다면 assert 를 발생시키게 된다. 이 실험에서 데이터의 크기가 하나의 패킷으로 갈 수 없는 경우에는 나눠진 모든 패킷에 데이터 크기를 명시해 문제가 없었다. 하지만, 전송되는 데이터가 단일 패킷으로 전송될 수 있는 경우에는 해당 패킷에 데이터 크기를 명시적으로 설정해주지 않는 특징으로 인해 assert 가 발생되었다. 이는 잠재적인 오류가 있을 수 있는 가능성이 있다고 판단하였으나, 개발자와 확인 후 데이터의 크기를 설정해 주는 부분이 모델링 부분 밖에 설정되어 있음을 확인하였다. 해당 부분에 대한 가정을 모델에 추가하여 오류가 없음을 확인하였다. 또한 데이터를 분리하여 전송할 경우 마지막 패킷이 언젠가는 전송되어야 함을 보장해주기 위하여 LTL (Linear Temporal Logic) property 를 이용하여 확인하였으며, 검증 결과는 모든 상황에서 해당 속성이 만족됐다.

3. 결론

본 논문에서는 MOST 를 플랫폼의 제약이 없도록 구현한 u-OMNiPro 1.0을 정형명세하고 정형검증을 수행한 과정 및 결과를 소개하였다. 정형검증 결과, 기존의 인스펙션과 테스트로는 발견할 수 없었던 두 개의 오류가 발견되었으며, 현 버전의 u-OMNiPro 2.0 에서는 수정이 완료되어 문제가 해결되었음을 확인했다. 현재 차량용 인포테인먼트 시스템의 네트워크 구축을 위해 개발 중인 k-OMNiPro (kernel-level OMNiPro) 를 대상으로, 본 연구에서 수행한 정형검증을 통해 신뢰성과 안전성에 검사를 수행할 예정이다.

참고문헌

[1] Mu-Youl Lee, Sung-Moon Chung, Hyun-Wook Jin, “Design and Implementation of MOST Network Service over POSIX“, Journal of IEMEK, Vol. 3, No 1, March 2010

[2] Holzmann, G. J. “The Model Checker SPIN“, IEEE Transactions on Software Engineering, Vol. 23, No. 5, May 1997