통신 프로토콜 검증을 위한 Esterel 정형검증 연구

김성재°, 김진현, 최진영
고려대학교 컴퓨터학과
{sjkim, jhkim, choi}@formal.korea.ac.kr

The Study of Formal Verification using Esterel for Network Protocol

Sung-Jae Kim °, Jin-Hyun Kim, Jin-Young Choi
Dept of Computer Science & Engineering, Korea University

요 약

인터넷의 확산과 네트워크 기술의 발전에 따라 네트워크 메커니즘은 그 설계 및 이해가 더욱 복잡해져 가고 있고, 분산 시스템 환경에서의 역할과 중요성도 날로 더해져 가고 있다. 본 논문에서는 네트워크 메커니즘의 정확성 검증을 위한 정형검증 연구의 일환으로, Reactive system의 모델링 및 검증을 위해 개발된 정형검증 언어인 Esterel을 이용한 Abracadabra 프로토콜의 정형 검증을 통해, 통신 프로토콜에 대한 Esterel의 정형검증 능력을 분석하고 좀 더 복잡한 통신 프로토콜에 대한 정형 검증의 적용 가능성을 타진하고자 한다.

1. 서 론

최근의 컴퓨팅 시스템 환경의 주목할 만한 변화는 시스템 대형화 및 네트워크 기술의 발전에 따른 기존 중앙 처리 시스템의 분산시스템 환경으로의 전환이다. 그러나 시스템간 통신 프로토콜 메커니즘의 안정성 및 효율성에 따라 그 변화의 효용가치는 크게 좌우될 수 있기 때문에 통신 프로토콜의 안정성에 대한 정확한 검증의 필요성이 제기되어 왔으며, 그에 따라 프로토콜의 안정성에 관한 연구 역시 다양하게 진행되어 왔다. 그러나 프로토콜의 안정성 검증을 위한 전통적인 방법인 테스팅 기법으로는 프로토콜 사용 시 발생 가능한 모든 경우에 대한 검증이 불가능하기 때문에, 프로토콜의 안정성 및 효율성을 완벽하게 보장하기 어렵다.

. 본 논문에서는 통신 프로토콜의 정확성 검증을 위한 정형검증 연구의 일환으로, *Reactive system*의 모델링 및 검증을 위해 개발된 함수형 정형검증 언어인 Esterel[1]을 이용한 Abracadabra 프로토콜의 정형검증을 통해 통신 프로토콜에 대한 Esterel의 정형검증 능력을 분석하고 좀 더 복잡한 통신 프로토콜의 안정성 검증에 대한 정형검증 기법의 적용 가능성을 타진하고자 한다.

2. 정형기법

정형 기법(Formal methods)[3]은 수학과 논리학에 기반을 둔 방법으로 하드웨어 시스템이나 정형 소프트웨어 시스템을 명세하거나 검증하는 방법론들이다. 수학적 기호를 사용하여 시스템의 명세를 하고 검증할 특성 또한 논리식을 통해 기술함으로써 시스템의 사용자가 요구하는 특정 조건에 대한 만족 여부를 수학적 성질을 이용하여 검증함으로써 자연어가 내포하는 애매모호함이나 불확실성을 최대한 줄일 수 있다. 즉 복잡한 시스템이 특정 조건을 만족하는지를 검증하여 검증된 시스템에 대한 신뢰성을 가지게 할 수 있는 것이다.

정형 기법은 정형 명세 (formal specification)와 정형 검증(formal verification)의 두 가지로 나눌 수 있는데, 정형 명세는 정형 논리(formal logic) 또는 수리 논리(mathematical logic) 등을 이용하여 시스템이 동작할 환경, 시스템이 만족해야 할 요구 사항, 요구사항을 수행할 시스템 설계등을 기술하는 것이다.

정형 명세는 다시 요구 명세와 설계 명세로 나눌 수 있다. 요구 명세는 시스템이 무엇을 만족해야 하는가를 정의해 놓은 명세이고 설계 명세는 시스템이 어떻게 이루어져 있는가를 나타낸다. 정형 검증은 정형 논리 또는 수리 논리등에서 제공하는 증명 방법을 이용, 정형 명세를 분석하여 시스템의 무모순성 및 완전성을 검증하거나, 설계가 주어진 가정에서 요구사항을 만족하는지를 검증하는 기법이다.

2-1. 정형 기법 도구

Eerel은 Reactive 시스템의 동기적(synchronous) 프로그램을 작성하는데 사용되는 동기적 언어(synchronous language)이며, 프랑스의 Sophia-INRIA 연구소에서 계속적으로 연구 개발되고 있다

Esterel toolset의 시뮬레이션 도구인 XES[3]는 Esterel로 명세된 시스템을 interactive하게 시뮬레이션 할 수 있는 tool로써 특정 입력에 대한 결과를 시각적으로 확인해 볼 수 있다. Esterel의 검증 도구인 XEVE[4]는 Finite State Machine으로 모델링된 Esterel 프로그램을 위한 검증 환경이다. Esterel 컴파일러를 통해 시스템을 설계한 프로그램을 FSM으로 정의하는 latch를 지닌 Boolean equation의 시스템으로 함축적(implicitly)으로 바꾼다. XEVE는 두 가지의 검증 기능을 제공하는데, 첫 번째로 FSM 그래프를 bi-simulation equivalence 방법을 사용하여 FSM 상태를 minimization하며, 이렇게 변환된 FSM을 통해 시스템 행위의 상태 변화를 관찰할 수 있다. 두 번째 검증방법은 출력 신호의 상태를 체크하는 것인데, 어떤 조건하에서 특정 신호의 발생 여부를 관찰함으로써 시스템이 만족해야 하는 조건을 검증할 수 있다.

3. Abracadabra Protocol

AbracadabrA는 K.J. Turner가 정형 명세 언어를 평가하기 위한 기준으로 제시한 통신 예제의 일종으로 Abracadabra Service와 Abracadabra Protocol로 구성되어 있다.

Abracadabra Service는 연결 지향형(Connection-oriented) 서비스로 Connection과 Disconnection만을 다루기 위해 흐름제어와 오류감지 기능을 제한 시킨 프로토콜의 추상화 서비스를 말한다. 이를 위해 Abracadabra Service는 다음과 같은 Primitive를 제공한다.

표 1 Service Primitives

단계	용도	이름	인자
Connection	Request	ConReq	-
	Indication	ConInd	-
	Response	ConResp	-
	Confirmation	ConConf	-
Data	Request	DatReq	SDU
	Indication	DatInd	SDU
Disconnection	Request	DisReq	
	Indication	DisInd	

Abracadabra Protocol은 앞서 설명한 Abracadabra Service 아래 계층에서 데이터 흐름을 제어하고 에러를 감지하면서 신뢰할 수 없는 전송매체를 매개로 데이터를 주고받을 수 있게 해 주는 역할을 한다. Abracadabra Protocol의 모형은 아래와 같다.

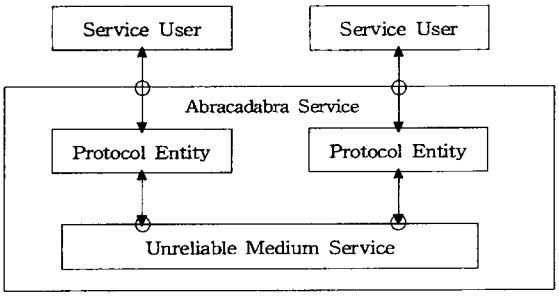

그림 1 Abracadabra Protocol

Abracadabra Protocol이 이용하는 흐름 제어 방법은 기본적으로 Sliding Window Protocol[2]에서 이용하는 방법과 같다. 송신자가 PDU (Protocol Data Unit)에 sequence number를 첨부하며 전송하면 수신자는 그 sequence number를 이용하여 올바른 데이터의 수신 여부를 판단한다. 원치 않는 데이터로 판별되면 해당 패킷을 버리며 올바른 패킷이라면 다음 자료를 요구한다. Abracadabra protocol은 window 사이즈를 2로 제한함으로써 sequence number는 0과 1을 반복하게 되는 일종의 Alternating Bit Protocol[2]의 모습을 띄고 있다.

한편 연결 설정 요구, 자료 전송 요구, 연결 해지 요구에 대해 상대측에서 응답이 없으면 재전송을 시도하고 일정횟수만큼 재 전송을 시도한 후에도 응답이 없으면 연결을 끊어버리고 초기 상태로 돌아간다. 또한 이미 설정된 연결 경로를 통해 데이터를 주고받는 동안 상대편이 연결 설정을 요구하는 부적절한 상황이 발생하면 연결을 끊고 양측 모두를 초기 상태로 돌려놓는다. 이러한 동작은 다음과 같은 protocol data unit을 이용하여 일어난다.

표 2 Protocol Data Unit

목적	이름	인자	관련 Primitve
Connection Request	CR	-	ConReq,ConInd
Connection Confirmation	CC	-	ConResp, ConConf
Data Transfer	DT	SDU	DatReq, DatInd
		Sequence Number	
Ack	AK	Sequence Number	-
Disconnection Request	DR	-	DisReq, DisInd
Disconnection Indication	DC	-	-

Abracadabra Protocol은 연결설정, 자료전송, 연결해지, 에러발생의 상태에 있을 수 있으며 상태의 전이는 현재 상태와 protocol data unit, service primitive에 따라 달라지게 된다. 아래의 그림을 보면 protocol data unit과 service primitive에 따른 상태 변화가 어떻게 일어나는지를 알 수 있을 것이다

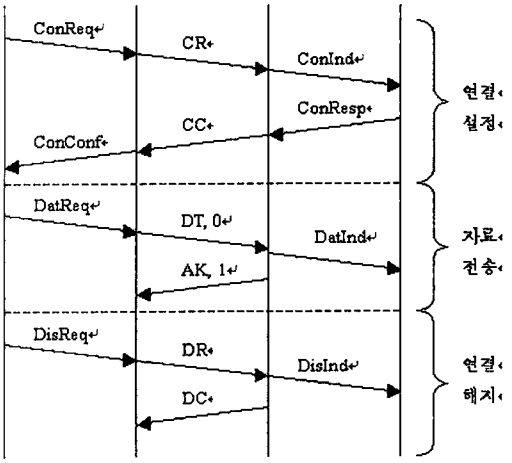

그림 2 Abracadabra의 상태 변화

4. Esterel을 이용한 Abracadabra의 모델링

본 논문에서는 4개의 모듈을 사용해 Abracadabra를 명세했으며 각 모듈의 이름과 역할은 다음과 같다..

1. User
   Abracadabra Service를 이용하는 모듈로 Service Primitive를 발생시키고 처리한다.

2. Station
   Abracadabra Protocol을 나타내며 가장 핵심적인 모듈이다. 사용자들로부터 요구 받은 행동을 실제 수행하는 Protocol Entity. Connection 및 Disconnection, Data transfer 중 나타나는 흐름 제어 및 오류를 감지한다.

3. TransCode
물리 계층을 나타내는 모듈이다. Station 모듈에서 받아온 패킷을 Unreliable Medium 모듈로 보내고 반대로 Unreliable Medium모듈로부터 패킷을 받아 Station 모듈로 넘겨준다.

4. UM
TransCode 모듈로부터 패킷을 받아 반대편 Transcode 모듈로 전송하는데 이 과정에서 패킷을 잃어버릴 수 있다. 즉 신뢰성 없는 전송 매체를 명세한 모듈이다. 이 모듈에서 일어나는 패킷 손실은 Station 모듈에서 일어나는 재 전송의 원인이 된다.

각 모듈간의 연결은 Signal을 Broadcast 하는 방법을 사용했으며 상위에 위치하는 모듈이 하위 모듈을 Run함으로써 채널의 개념을 대신하였다. 따라서 서로간에 직접 연관이 있지 않은 모듈들의 Signal에는 영향을 받지 않는다.

Abracadabra 프로토콜 계층은 전송 매체와 같은 물리 계층을 매개로 하여 통신을 하고 서비스 계층은 프로토콜 계층을 매체로 하여 통신을 한다. 따라서 Abracadabra 시스템을 검증하기 위해서는 물리 계층을 매개로 한 프로토콜 계층간의 통신을 검증한 후 프로토콜 계층을 매개로 한 서비스 계층간의 통신을 검증하면 된다. 구체적인 Esterel 코드는 지면 관계상 생략한다.

5. XEVE를 이용한 검증

본 논문에서는 가상의 사용자 A,B를 설정하여 다음과 같은 Safety에 대한 검증을 수행해 보았다.

Safety01 -한번 접속이 이루어진 상태에서 들어오는 또 다른 접속 요구는 무시한다

Safety02 - 두 사용자 A,B 가 동시에 ConReq를 보낸 경우에는 양 Station의 상태가 동시에 Estab 상태로 전이되며 사용자 A와 B는 Primitive ConConf를 받게 된다.

Safety03 - 사용자 A가 보낸 데이터는 언젠가 반드시 사용자B 에게 도달하게 된다.

위의 Safety Property들에 대한 각각의 LTL logic formula는 다음과 같다.

```
Always
{ ((not STATE_DISCONNECTED & A_CONREQ) |
  (not STATE_DISCONNECTED & B_CONREQ)) ->
  (not B_CONIND | not A_CONIND) }

Always
{ (not STATE_DISCONNECTED & (A_CONREQ & B_CONREQ)) ->
  ((A_ESTAB & B_ESTAB) & (A_CONCONF & B_CONCONF) ) }

Always
{ (A_DATREQ -> B_DATIND & SENDSEQ ) }
```

위의 Satety에 관한 검증은 Esterel toolset의 검증 도구인 XEVE를 이용해서 진행했으며 observer를 통한 출력신호의 유무여부를 통해 Safety property를 위반하는 모든 경우의 수를 검사하였으며, 검사 결과 해당 Safety property를 위반하는 경우는 발생하지 않음을 (Never Emitted) 그림 3을 통해 알 수 있다.

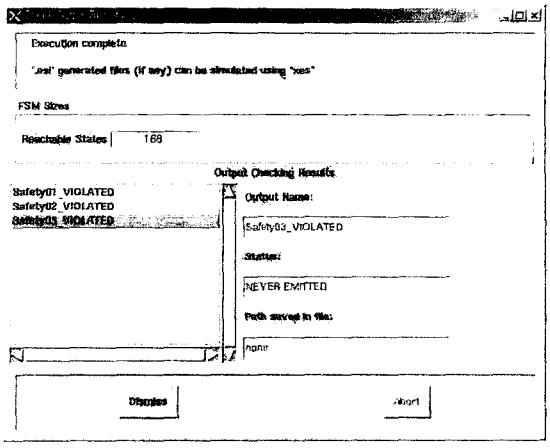

그림 3 XEVE를 이용한 Property의 검증

6. 결론 및 향후 과제

네트워크 환경의 변화로 인해 다양한 통신 메커니즘이 제안되고 있지만 전통적인 테스팅 기법으로는 프로토콜에서 발생 가능한 모든 상태에 대한 검사가 불가능하다.

본 논문에서는 정형검증 기법을 이용한 프로토콜의 안정성 연구의 일환으로, Reactive system의 모델링 및 검증을 위해 개발된 Synchronous language인 Esterel과 XEVE를 이용해 Abracadabra 프로토콜이 만족해야 할 상기의 Safety property를 위반하지 않음을 보였다. Esterel은 기존의 통신 프로토콜 검증 언어인 SDR이나 Estelle등과 비교해 보았을 때 c 와 유사한 함수형 언어로써 프로그램 작성에 있어서의 편리성을 제공하고 있으며, 통신 프로토콜에 있어서의 검증능력 역시 여타의 검증 도구와 비교해 보았을 때 떨어지지 않는다는 사실을 본 연구 결과를 통해 확인해 볼 수 있었다.,

향후 과제로는 Abracadabra의 정형검증을 통해 분석된 Esterel의 정형검증 능력을 TCP/IP 프로토콜의 분석에 적용해 보고자 한다.

7. 참고문헌

[1] Gerard Berry. *Esterel V5 Language Primer v5_21 release 2.0.* April 6.2000
[2] Kenneth J.Turner. *Using Formal Description Techniques.* John Wiley & Sons, 1993
[3] Edmund M.Clake, Orna Grumberg, Doron A. Peled.. *Model Checking.* The MIT press.1999
[4] G. Berry and The Esterel Team. *The Esterel v5_21 System Manual.* INRIA. France. March 11.1999
[5] Amar Bouali. XEVE –"an Esterel Verification Environment(Version v1_3) " . INRIA.France.1997
[6] Joost-Pieter Katoen, *Protocol validate*
[7] B.Berard, *Systems and Software Verification,* Springer. 1999
[8] Doron A. Peled, Software Reliability Methods, Springer, 2001