2004년도 대한전기학회 하계학술대회 논문집 2004.7.14-16

철도신호시스템과 전철전력 SCADA 장치간 프로토콜 설계 및 정형검증

황종규, 이재호, 윤용기
한국철도기술연구원 전기신호연구본부

Design and Formal Verification of Protocol for Interface between Railway Signaling Systems and SCADA Systems

Korea Railroad Research Institute(KRRI)
Hwang Jong-Gyu, Lee Jae-Ho, Yoon Yong-Gi

Abstract - 철도 신호제어장치들은 각자 고유의 기능을 수행하면서 각 장치간 통신링크를 통하여 하나의 신호제어시스템을 구성하고 있다. 특히 철도청에서 통합 CTC 시스템을 구축하면서 신호제어시스템 이외의 SCADA나 여객정보안내 시스템 등과도 인터페이스를 통해 기존의 열차제어 기능만을 수행하는 것에서 타 시스템과의 통신을 통한 종합적인 정보시스템으로 발전하고 있다. 이러한 CTC장치와 외부설비들간의 인터페이스는 철도정보시스템의 발달에 따라 매우 중요한 부분이 되고 있으며, 본 논문에서는 이중 SCADA 장치와의 인터페이스를 위한 프로토콜 구조를 연구하였다. 이에 따라 본 논문에서는 기존의 도시철도, 경부고속철도 등의 프로토콜의 분석을 바탕으로 철도청 통합 CTC와 SCADA 장치간 통신을 위해 설계한 프로토콜 구조를 제시하고, 또한 설계한 프로토콜에 대한 안전성과 필연성을 정형검증(Formal Verification) 결과를 설명한다.

1. 서 론

철도 신호제어장치들은 각자 고유의 기능을 수행하면서 각 장치간 통신링크를 통하여 하나의 신호제어시스템을 구성하고 있다. 이러한 열차제어를 위한 신호제어시스템이 최근 들어 철도 선진국을 중심으로 정보통신 기술의 발달에 따라 기존의 열차제어 기능만의 수행에 더하여 다른 장치들과 인터페이스를 통한 통합적인 정보시스템으로 발전하고 있다. 국내의 경우 현재 구축 중인 철도청의 통합 CTC 시스템도 전력 SCADA 장치, 여객정보안내 시스템, 고속철도 통합정보시스템 등 외부장치들과 통신링크를 통한 인터페이스가 추진 중에 있다. 즉, 열차제어 기능을 수행하는 CTC 시스템이 이러한 외부 설비들과의 인터페이스를 하는 등 각 시스템들이 서로간 인터페이스를 함으로써 철도 시스템 전체의 통합 정보시스템으로 발전하고 있다.

철도청의 주요 노선들의 전철화가 진행되어감에 따라 전차선 섹션별 가압상태를 감시하고 제어하는 전철 SCADA 장치의 중요성이 증가되고 있다. 이러한 전철화에 따라 전차선의 가압상태 정보는 열차의 안전운행을 위해 CTC 장치에서 반드시 필요한 정보이다. 또한 SCADA 장치도 섹션별 운행열차의 수 등이 전차선 제에 매우 중요한 요소가 되고 있다. 이처럼 기존의 열차제어만을 위한 신호제어장치들이 전철화 등의 상황변화와 기술발전에 따라 SCADA 장치 등 외부설비와의 인터페이스 필요성이 점차 증대되고 있다[1].

이러한 CTC 시스템과 인터페이스가 필요한 외부설비들 중 본 논문에서는 철도청 통합 CTC장치와 SCADA 장치와의 인터페이스를 위한 프로토콜 구조를 연구하였다. 이를 위해 기존의 도시철도 시스템들과 경부고속철도 시스템의 프로토콜들을 분석하였다. 이러한 분석들을 토대로 철도청 통합 CTC와 SCADA 장치간 통신을 위한 프로토콜 구조를 제시한다.

일반적으로 비정형적인 방법에 의해 설계된 프로토콜은 모호함과 불완전성 등으로 인한 프로토콜의 오류와 비효율성이 발생할 수 있으므로 사전에 충분히 검정되어야 한다. 이를 위해 본 논문에서는 정형검정하기 위해 형식기법의 하나인 LTS(Labeled Transition System)으로 설계된 프로토콜을 명세화하고 이 모델을 Modal μ-calculus 로직을 적용한 모델체킹 방법을 통해 안전성(Safety)과 필연성(Liveness)을 검정하였다. 본 논문에서는 설계한 CTC와 SCADA 장치 사이의 프로토콜 구조와 정형검정 결과를 간략하게 설명한다.

2. 기존 프로토콜 분석

철도청의 경우 대부분은 CTC와 전력 SCADA 장치 사이에 인터페이스가 이루어지고 있지 않지만, 현재 구축되고 있는 경부고속철도나 각 지역의 도시철도 시스템의 경우는 대부분 이들 두 장치사이에 인터페이스를 하고 있다. 철도청의 통합 CTC 시스템과 SCADA 시스템과의 인터페이스를 위한 정보전송방식의 도출을 위해 우선적으로 각 지자체들이 운영하고 있는 도시철도 시스템과 경부고속철도 시스템에서 이들 두 장치사이의 통신 프로토콜을 조사하였다. 대부분의 시스템들이 이들 두 장치 사이의 인터페이스를 위해 시리얼 통신을 사용하고 있었으며, 에러제어 방법이나 전송 메시지 프레임 포맷도 서로 다른 것으로 분석되었다. 특히, 경부고속철도에 적용된 통신방식은 전력 SCADA 장치로부터 CTC 장치로 가압정보는 전송을 하지만 반대방향으로 열차정보는 전송하지 않는 단방향 통신채널을 구성하고 있었다. 두 장치간의 전송정보의 내용도 한 프로토콜에서는 세션별 점유하고 있는 열차의 개수에 대한 정보를, 다른 프로토콜에서는 세션별 열차의 점유유무 정보만을 전송하고 있었다. 하지만 전력 SCADA 장치로부터의 가압정보 전송은 대부분 각 세션별 가압의 유무 상태정보를 전송하고 있었다. 이 중 본 절에서는 대표적인 두 가지 방식에 대해서 설명한다.

2.1 A방식 프로토콜

A방식의 프로토콜은 서울2기 지하철 시스템 등에 사용되는 TTC 장치와 전력 SCADA 장치 사이에 적용되고 있는 프로토콜로서, 표1은 프로토콜에 대한 요약한 것이다.

표 1 A방식 프로토콜

물리계층	RS 232 Asynchronous
전송속도	9600 bps
에러검출	CRC-16($X^{16}+X^{15}+X^2+1$)
프레임 포맷 구성	STX+Length+Seq. No.+OP Code+Data+CRC
전송코드	ACK/NAK(C⇒S), SCADA Info(S⇒C) Train Information(C⇒S)

A방식 프로토콜은 이벤트 발생 시 정보가 전송되는 A 방식과는 달리 각 시스템이 모든 상태를 상태변화의 유무에 상관없이 주기적으로 상대 시스템으로 전송하는 방식을 적용하고 있다. 이러한 주기적인 정보업데이트로 인하여 일반적으로 적용되는 Link Test 메시지가 필요없게 된다.

2.2 B방식 프로토콜

B방식의 프로토콜은 경부고속철도 시스템에 적용되는 프로토콜로서 표2처럼 SCADA 장치로부터 전차선의 가압상태 정보를 CTC 장치가 전송받기만 하는 단방향 통신방식을 사용하고 있으며, MODBUS 프로토콜이 기준 프로토콜로 적용되고 있다.

표 2 C방식 프로토콜

물리계층	RS 485 Asynchronous
전송속도	9600 bps
프로토콜	MODBUS 프로토콜 사용 ·Mater : CTC, Slave : SCADA ·정보전송방향 : 단방향(SCADA⇨CTC)
메시지 흐름제어	SCADA 시스템에서 주기적(2 sec)으로 전차선의 가압상태를 전송
적용	경부고속철도

이 MODBUS 프로토콜은 일반적으로 시스템들은 주장치-종속장치로 인터페이스 되는 부분에 많이 사용되는 프로토콜이다. 즉, 한 장치는 언제나 주가 되고 다른 한 장치는 언제나 종속장치가 되는 기법이다. 즉, 주 장치인 CTC는 종속 장치인 SCADA로부터 전차선의 상태정보만 업데이트하게 되고 CTC의 상태정보는 전송하지 않는 단 방향 정보전송 방식이다.

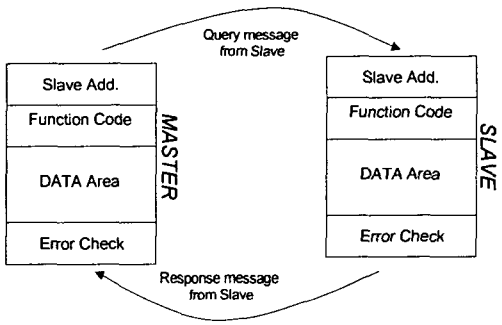


그림 1 MODBUS의 구조

3. 표준프로토콜 구조

앞 절에서 설명한 기존의 CTC와 SCADA 장치사이의 인터페이스는 기본적으로 직렬통신 방식을 사용하고 있었으며, 또한 두 장치사이에 전송되는 메시지고 조금씩 차이가 있었다. 그리고 C방식의 경우는 SCADA에서 CTC로 정보를 제공하지만 반대방향으로는 정보를 제공하지 못하는 방식이 적용되고 있었다.

이러한 다른 시스템에 적용되고 있는 기존의 프로토콜들의 분석과 운영기관의 의견수렴 등을 거쳐 두 장치 사이의 전송되어야 할 메시지를 도출하였다. 또한 인터페이스 방식은 Ethernet 기반의 네트워크에 의한 통신방식을 적용하는 것으로 하였다. 하지만 통신서버를 별도로 두고 각각에 방화벽을 설치하고 그리고 라우터를 통해 인터페이스 함으로써 CTC로 대표되는 신호시스템과 전력 SCADA 시스템과는 완전히 분리되도록 하여 네트워크에 의한 인터페이스에 있어서의 보완문제를 해결할 수 있다. CTC 시스템은 전력 SCADA 장치로부터 열차운행에 필요한 급전구간의 상태정보를 수신하기 위해 인터페이스 되며, 불통구간 발생 시 열차운행관리에 활용하도록 하였다. 다음은 본 연구를 통해 제시한 두 장치 사이의 통신 프로토콜의 구조를 간략하게 설명한 것이다.

3.1 링크구성

철도청 통신망을 경유하여 이중화 회선으로 구성되는 전력 SCADA 시스템과의 인터페이스 사양은 표 3과 같으며, 그림 2는 이들 사이의 링크 구성을 나타낸 것이다.

표 3 CTC 통신서버 ⇔ 전력 SCADA간의 인터페이스 사양

항목	규격	상세사양
접속방식	LAN 접속(RJ45)	라우터 사용
전송속도	10/100 Ethernet	100 Mbps
네트워크 모델	Client-Server 모델	Server : 전력 SCADA 장치 Client : 통합사령실
통신 프로토콜	TCP/IP 프로토콜	·통신프로세스 인터페이스
회선 수	2 회선	주 통신라인 및 예비 통신라인

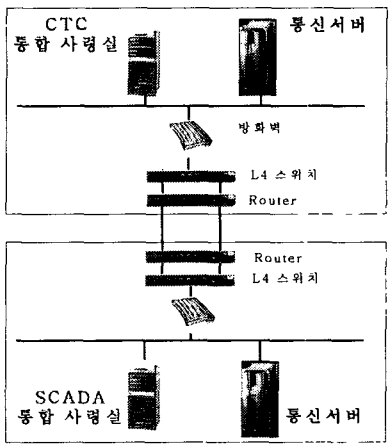


그림 2 SCADA ⇔ CTC 링크 구성

3.2 데이터 패킷 구조

Ethernet을 기반으로 한 두 장치사이의 전송되는 데이터 패킷의 형태는 다음과 같은 구조를 가진다. 송수신 메시지의 에러검사를 위해서 CRC-16 코드를 사용하였으며, 또한 두 장치사이에 정보전송은 CTC나 SCADA 장치에서 상태정보가 변경될 경우마다 전송되도록 하지 않고, B 방식과 같이 모든 상태 정보를 주기적으로 전송되도록 하였다. 또한 CTC에서 SCADA 장치로 전송되는 메시지는 열차의 점유된 전차선의 섹션 정보와 열차번호를 동시에 전송되도록 하였다. 그리고 반대로는 섹션별 전차선의 가압상태 정보를 CTC로 전송되도록 하였다.

Ethernet 정보	IP Header	TCP Header	장치간 송수신 데이터	Ethernet 정보

그림 3 데이터 패킷 기본구조

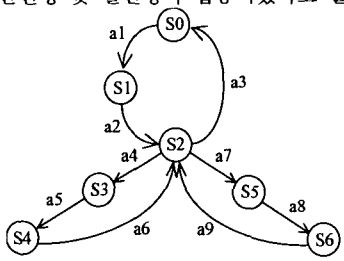
Byte	의미	비고
1	STX	Message Header
2	CRC를 제외한 Message information 부분의 길이	
3	SEQ_NO(Sequence number)	Message Information
4	OP_CODE(Message function code)	
5 . . .	Signification Data	
n-1	CRC-LOW	
n	CRC-HIGH	

그림 4 송수신 데이터 필드의 구성

3. 모델체킹에 의한 프로토콜 정형검정

통신 프로토콜이 적절한 기능을 수행하기 위해서는 프로토콜 각 해당 상태의 Deadlock나 Livelock 및 비정상적 도달(Reachibility)과 같은 잠재적인 설계에러가 없어야 한다. 모델체킹(Model Checking) 방법은 프로토콜의 안전성과 필연성 특성을 보다 구체적으로 검정하기 위한 방법으로 이를 위해서는 프로토콜의 행위특성을 나타내는 형식언어로 명세화를 한다. 이러한 명세화된 프로토콜을 Modal μ-calculus로직을 이용하여 검정하고자 하는 상태 및 행위 특성을 모델링 한 후 이 로직을 별도의 알고리즘을 이용해 정형검정하게 된다. 본 논문에서는 형식명세를 위해 LTS를 적용하였고, 또한 정형검정을 위해 Solve 알고리즘을 적용하였다[2][3].

그림 5는 앞 절에서 설명한 프로토콜을 6개의 상태와 8개의 행위로 구성되어진 LTS로 명세화한 결과를 나타낸 것이다. 이렇게 프로토콜의 모델로부터 얻어진 다음과 같은 Modal μ-calculus 로직이 검정되, 주어진 프로토콜은 안전성 및 필연성이 입증되었다고 할 수 있다.



S0 : idle	S1 : ack_awaited	
S2 : TCP_con_astab	S3 : T1=P1	S4 : resp_awaited
S5 : T2=P2	S6 : ack_awaited	
a1 : TCP_con_req	a2 : ack	a3 : release
a4 : operate_T1_timer	a5 : train_state-msg	a6 : ack
a7 : operate_T2_timer	a8 : SCADA_state_msg	

그림 5 프로토콜의 LTS 모델링

$$\nu Z.[\mu Y.A \vee (<->tt \wedge [-]Y)) \wedge [-]Z,\ A=\{S0,\ldots\}$$ 식 (1)

제시된 식(1)과 같은 로직을 Solve 알고리즘에 의해 모델체킹하였다. 다음 그림은 Solve 알고리즘에 의한 모델체킹 전과 후의 상태를 비교한 것으로 이 결과로부터 제시된 프로토콜에 Deadlock나 Livelock 상태가 없음을 확인할 수 있다.

X	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
S0	0	0	0	0	0	1	1	1
S1	0	0	0	0	0	1	1	1
S2	0	1	0	0	0	1	1	1
S3	0	0	0	0	0	1	1	1
S4	0	0	0	1	0	1	1	1
S5	0	0	0	0	0	1	1	1
S6	0	0	0	0	0	1	1	1

C	X_3	X_4
S0	2	1
S1	2	1
S2	2	3
S3	2	1
S4	1	0
S5	2	1
S6	2	1

M[1]=<<S2, X2>, <S4, X4>, <S0, X6>, <S1, X6>, <S2, X6>, <S3, X6>, <S4, X6>, <S5, X6>, <S6, X6>>
M[2]=< >

(a) 초기상태

X	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
S0	1	0	1	1	1	1	1	1
S1	1	0	1	1	1	1	1	1
S2	1	1	1	1	1	1	1	1
S3	1	0	1	1	1	1	1	1
S4	1	0	1	1	1	1	1	1
S5	1	0	1	1	1	1	1	1
S6	1	0	1	1	1	1	1	1

C	X_3	X_4
S0	0	0
S1	0	0
S2	0	0
S3	0	0
S4	0	0
S5	0	0
S6	0	0

M[1]=< >
M[2]=< >

(b) 알고리즘 수행결과

그림 6 Solve 알고리즘에 의한 모델체킹

4. 결 론

본 논문에서는 최근 들어 인터페이스 필요성이 증대되고 있는 CTC 장치와 SCADA 장치 사이의 통신 프로토콜 구조를 제시하였다. 이 제시된 프로토콜을 기반으로 하여 장치간의 점유한 열차번호 정보, 전차선 구간의 가압정보 등이 교환됨으로 인해 CTC에서는 보다 안전하게 열차운행 제어가 가능하게 되고, 또한 SCADA 장치에서는 열차의 운행 상태 정보를 통해 보다 효율적인 시스템의 운용이 가능해 질 수 있을 것으로 기대된다.

이러한 설계한 프로토콜들은 철도신호시스템에 적용되어지므로 반드시 안전성과 필연성이 검증되어야 한다. 이를 위해 본 논문에서는 정형기법에 의해 프로토콜의 안전성 및 필연성 검정한 결과를 간략하게 나타내었다. 이를 통해 제시된 프로토콜의 안전성을 확인할 수 있었다.

[참 고 문 헌]

[1] 철도기술연구개발사업 연구보고서, "신호설비 유지보수 효율화를 위한 정보전송방식 기술연구", 한국철도기술연구원, 2003.

[2] 서미선, 김성운, 황종규, 이재호, "LTS로 명세화된 철도신호제어용 프로토콜 검정 및 적합성시험", 한국철도학회 추계학술대회 논문집, 2003.

[3] D. Schwabe, 'Formal Techniques for the Specification and Verification of Protocol', Ph.D Thesis, Univ. of Califonia Los Angeles, Apr. 1981.