Formal Verification Network-based Protocol for Railway Signaling Systems

Jong-Gyu Hwang, and Jae-Ho Lee

Korea Railroad Research Institute(KRRI), Korea
(Tel : +82-31-460-5438; E-mail: {jghwang, jhlee1}@krri.re.kr)

Abstract: According to the computerization of railway signaling systems, the interface link between the signaling systems has been replaced by the digital communication channel. At the same time, the importance of the communication link is more pronounced than in the past. In this paper, new network-based protocol for Korean railway signaling has designed between CTC and SCADA system, and the overview of designed protocol is briefly represented. Using the informal method for specifying the communication protocol, a little ambiguity may be contained in the protocol. To clear the ambiguity contained in the designed protocol, we use LTS model to design the protocol for this interface link between CTC and SCADA, the LTS is an intermediate model for encoding the operational behavior of processes. And then, we verify automatically and formally the safety and the liveness properties through the model checking method. Especially, the modal μ-calculus, which is a highly expressive method of temporal logic that has been applied to the model checking method. It will be expected to increase the safety, reliability and efficiency of maintenance of the signaling systems by using the designed protocol for railway signaling in Korea.

Keywords: : Formal verification, Modal- μ -calculus, protocol for railway signaling, LTS(Labeled Transition System)

1. INTRODUCTION

There are many computerized equipment in railway signaling systems such as CTC (Centralized Traffic Control) system, EIS (Electronic Interlocking System), ATC (Automation Train Control) system, and LDTS (Local Data Transmission System) and so on. Lots of information is exchanged among the computerized railway signaling systems. The importance of protocol for railway signaling systems was increased by increase of exchange of information among computerized signaling systems. For this reason, it is required the standard communication protocol with high reliability for railway signaling systems [1][2]. Among of them, in this research, we concentrate the interface link between CTC and SCADA (Supervisory Control and Data Acquisition) system. The CTC system is a major railway signaling system. This is located at central control and monitoring center, and total trains are operated by this CTC system. SCADA system has a role for control and monitoring of railway power systems such as catenary system, railway power stations and etc. The very important information has to be exchanged between above two systems.

It is expected that the communication protocols designed by experts could have brought about some ambiguities. Provided that there were some ambiguities in the designed protocol, and that the protocols were applied to vital railway signaling systems, the ambiguities might provoke fatal faults in the control of signaling systems or accidents. To clear the ambiguity contained in the designed protocol, we apply a formal method to the designed protocol and verify the safety and liveness properties with the model checking method. In this research, we use LTS (Labeled Transition System) model to design the protocol for this interface link between CTC and SCADA, the LTS model is an intermediate model for encoding the operational behavior of processes. And then, we verify formally the safety and the liveness properties through the model checking method. Especially, the modal μ -calculus, which is a highly expressive method of temporal logic that has been applied to the model checking method[3]-[5].

The overview of designed network-base protocol for Korean railway signaling systems and formal verification results will be presented in this paper. It is expected that there will be an increase in safety, reliability and efficiency in terms of the maintenance of the signaling system by using the designed protocol for Korea railway signaling systems.

2. NETWORK-BASED PROTOCOL FOR RAILWAY SIGNALING SYSTEMS

2.1 Configuration of interface link

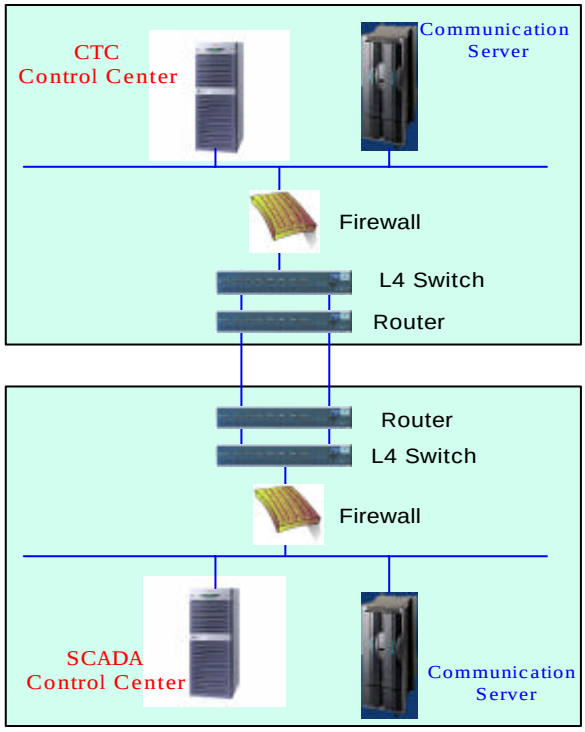


Fig. 1 Configuration of SCADA ✍ CTC link

Recently according to the computerization of railway signaling systems such as CTC, EIS, ATC and so on, the importance of protocol for railway signaling systems is increased by increase of exchange of information among computerized signaling systems. For this reason, it is required

the protocol with high reliability for railway signaling systems. Among of them, in this research, we concentrate the interface link between CTC and SCADA. The CTC system is a major railway signaling system and SCADA system has a role for control and monitoring of railway power systems such as catenary system and etc.

Few years ago, there was no interface link between these two systems, so these two systems have been operated separately. However the interface between SCADA and CTC has been important according to the upgrade of train operating speed such as Korean high-speed train (KTX), because CTC system for train control has been needed the catenary's information for safety operation of high-speed trains. Fig. 1 shows the configuration of interface link between these two systems by Ethernet LAN. SCADA systems send the states information such as powered catenary sections or not, and reversely CTC systems sends the location information or train number of running trains to SCADA systems. Follows are summarized information for interface between these two systems.

- CTC ⇒ SCADA
  - ?CTC system state information message : currently occupied section no. and trains no. of running trains
  - ¡Control message : ACK/NAK
- SCADA ⇒ CTC
  - ¡SCADA system state information message : catenary section ID and state information of each section(powered section of not)
  - ¡Control message : ACK/NAK

2.2 Structure of designed protocol

As the above-mentioned configuration of interface link between CTC and SCADA, Ethernet LAN is applied to this link. Therefore the network-based communication protocol is required to the interface of this links. So we have designed new protocol for this interface link by network-based structure. In this section, the designed protocol for railway signaling will be described concisely.

The designed protocol has used the well-known TCP/IP protocol for Ethernet LAN as a transport and network layer protocols. Fig. 2 shows the configuration of TCP/IP protocol-based message format on the Ethernet. So the functions for transport and network layers are not defined, but only the transmitted data field is defined to the new designed protocol. This protocol will be applied to vital railway signaling systems for safely train control, so it is significant to reflect the function for improvement of reliability and safety at protocol design. Therefore the 'Transmitted Data' field is very important step at TCP/IP-based protocol design for safety-critical system such as railway signaling systems.

Ethernet Header	IP header	TCP header	**Transmitted Data**	Ethernet tailor

Fig. 2 Configuration of message format.

Length	Field	Remarks
1 byte	**STX**	Message Header
2 bytes	**Data Length**	Message Information
1 byte	**Sequence Number**	
1 byte	**Message Type**	
N bytes	**Data**	
2 bytes	**CRC**	

Where
- STX: start of frame
- Data Length: message length from Sequence Number to data (maximum 255 bytes)
- Sequence Number: 0x00 ~ 0xFF
- Message Type: transmitted message types
- Data: variables according to information
- CRC: CRC-16(X^{16} ? X^{15} ? X^{2} ? 1)

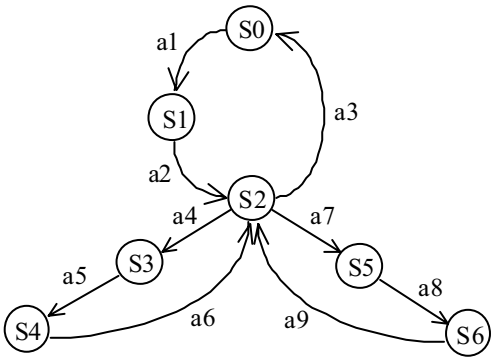
Fig. 3 Structure of message data field.

Fig. 3 shows that the 'Transmitted Data' field of message format consisted two parts: message header and message information parts. In message header part, 'STX' and 'Data Length' fields are included for the function of start of transmitted message frame and bytes length of information data part. And there are real transmitted fields in the message data part such as 'Sequence Number', 'Message Type', 'Data' and 'CRC' field. As you know, it is not required the 'Sequence Number', 'Message Type' and 'CRC' field in well-known data field for TCP/IP-based protocol, because similar function for above-described several fields are already implemented in TCP and IP protocol. However those duplicated fields in the aspect of functions with some fields in TCP and IP protocol are added to improve the reliability and safety of designed protocol performance because the designed protocol have to applied to vital railway signaling systems. According to those specially added fields, the designed protocol is able to apply vital control systems which required higher reliability such as railway signaling systems [3].

3. FORMAL VERIFICATION OF DESIGNED PROTOCOL SPECIFIED IN LTS

It is expected that the communication protocols designed by experts have some ambiguities. Provided that there are some ambiguities in the designed protocol by experts and the protocol is applied to a vital railway signaling system, these may provoke the fatal faults in the control of signaling or accidents. Therefore, the communication protocol for vital systems such as the railway signaling system has to be correctly verified by the formal method.

For our research, we use LTS (Labeled Transition System) to design the protocol for railway signaling, the LTS is an intermediate model for encoding the operational behavior of processes. And then, we verify automatically and formally the safety and the liveness properties through the model checking method. Especially, the modal μ-calculus, which is a highly expressive method of temporal logic that has been applied to the model checking method. Figure 4 also shows the LTS modeling with function model for formal specification.



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
a9 : ack	

Fig. 4 LTS modeling for designed communication protocol.

3.1 Overview for verifying correctness

As a highly focused part in the development of the protocol, verifications for protocol specification are complementary techniques that are used to increase the level of confidence in the system functions as prescribed by their specifications. That is, it must verify the nonexistence of deadlock and livelock states, abnormal reachability, and potential design errors in the designed protocol. The basic approach to verifying whether our designed protocol satisfies a property by using the model-checking concept involves the following steps:

- The communication protocol is expressed as a FSM and a set of labeled transition remarks.
- The protocol to verify is expressed as a LTS model from pre-worked FSM model.

An algorithm, called a model checker, is used to verifying whether the model expressed as a LTS satisfies the properties of correctness. There are two kinds of properties, which are as follows:

- Safety property is the state in which bad states and actions never happen. That is, the system never enters an unacceptable state. Some well-known examples of safety properties in communication protocol are the absence of deadlock or livelock.
- Liveness property is the state in which good states and actions eventually happen. That is, predefined states and actions necessarily take place. This is defined as reachability.

3.2 Modal μ-calculus as a modal-checking

Modal μ-calculus of Kozen[3] is a a powerful logic for expressing temporal properties by using the least and greatest fixed-point operators that can express the safety and liveliness properties of communication protocols. In modal μ-calculus, formulas consist of atomic propositions, $\wedge$ (conjunction), $\vee$ (disjunction), [ ](necessity), < >(possibility), ν(greatest fixed point) and μ(least fixed point). And generalized formulas for modal μ-calculus are as follows:

$\varphi ::= tt \mid ff \mid Z \mid \varphi 1 \wedge \varphi 2 \mid \varphi 1 \vee \varphi 2 \mid [k]\varphi \mid <k>\varphi \mid \nu Z.\varphi \mid \mu Z.\varphi$ (1)

where

- ν & μ: Operators for expression of least and greatest fixed-point, respectively
- $\wedge$ & $\vee$: Local connectives
- [] & <> : Modal operators with meaning of necessity and possibility
- φ : Formulas for process characteristics

Safety property is based on the exclusion of deadlock and livelock states, and liveness property is the characteristic of satisfaction of reachibility and liveness in communication protocol. If the following modal mu-calculus formulas are true respectively, it means that the safety and liveness characteristics of designed communication protocols have been verified correctly.

- Safety for states : $\nu Z.\varphi \wedge [-]Z$
- Safety for behavior : $\nu Z.\varphi [k]ff \wedge [-]Z$
- Liveness for states : $\mu Z.\varphi \vee (<->tt \wedge [-]Z)$
- Liveness for behavior : $\mu Z.\varphi <->tt \wedge [k]Z$

3.3 Verification of designed protocol

In this section, we will verify the designed protocol using the modal μ-calculus fomula described to equation (1), which means we will examine the correctness properties of the protocol. For example, the equation (2) modal μ-calculus fomula has to be "true", if the LTS of the designed protocol consists of the non-existence of deadlock and livelock.

$?Z.\ (?Y.A\ ?(<->tt\ ?\ [-]Y))\ ?\ [-]Z\ ,\ A=\{S0\}$ (2)

$B_1 \equiv \min\{X_1 = X_2\ ?\ X_3$	$B_2 \equiv \max\{X_7 = X_1\ ?\ X_8$
$X_2 = A$	$X_8 = [-]X_7\}$
$X_3 = X_4\ ?\ X_5$	
$X_4 = [-]X_1$	
$X_5 = <->X_6$	
$X_6 = tt\}$	

Fig. 5 Max block, min block

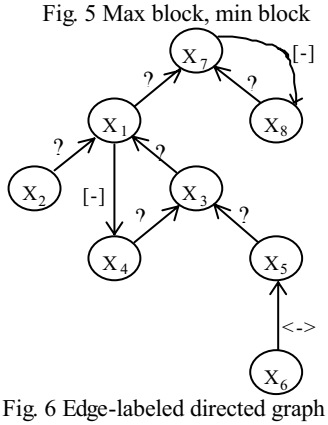


Fig. 6 Edge-labeled directed graph G

The Solve as a model-checking algorithm is applied to this equation (2). From this process the verifying results can be obtained. Figure 5 shows the max and min blocks generated from the equation (2) by using least and greatest fix-points and Figure 6 shows the relation between the variables' transitions described as an edge-labeled directed graph G.

Figure 7(a) shows the bit-vector, counter and array, which are initialized by means of the solution algorithm. And Figure

7(b) shows the resulting bit-vector, counter, and array. We can decide the deadlock and livelock properties from the results.

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) Initial states

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) Resulted states

Fig. 7 Bit-vector, counter and array M[i]

A deadlock can be detected at components of bit-vector. All bit-vectors from resulted bit-vector, with the exception of component X_2 relate to atomic proposition A, set to '1', so we can find the absence of a deadlock in the LTS model of the designed protocol. Therefore, the livelock property can be detected through the components of the counter. All counters from resulted counter are set to '0', so we can find the absence of lovelock in the LTS model. From these results, the above described LTS model for the designed protocol is satisfied with the "?Z. (?Y.A ?(<-> tt ? [-]Y)) ? [-]Z , A={S0}" formula. This means that the protocol is verified as the proper model for satisfying the correctness properties.

REFERENCES

[1] J. G. Hwang and J. H. Lee, 'Performance analysis of data link protocol for railway signaling', Proceeding of World Congress on Railway Research, pp. 1427-154, Oct. 2003.

[2] J. G. Hwang and J. H. Lee and et al., 'A New Data Link Protocol for Korea Railway Signaling Systems', KIEE Int'l Trans. on EMEC, Vol.3-B, No.4, pp. 195-201, 2003..

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

[4] D. Kozan, 'Results on the prepositional μ-calculus', Theoretical Computer Science, 27:333-354, December 1983.

[5] R. Cleaveland, 'Tableau - Based Model-Checking in the Propositional μ-Calculus', Acta Informatica 27 : 725-727, 1990.