

철도용 정보전송 및 네트워크 프로토콜 성능 시뮬레이션

강문호*, 이재호**, 황종규**

* 선문대학교 제어계측공학과, ** 한국철도기술연구원(KRRI)

Performance Simulation of the Data Transmissions and Network Protocols for Railway Signalling

Kang, Moon-Ho*, Lee, Jae-Ho**, Hwang, Jong-Gyu**

* Dept. of Control and Measurement Engineering, Sunmoon Univ., ** Korean Railroad Research Institute (KRRI)

Abstract - This paper addresses a simulation for the Ethernet based data transmission between the CTC(Centralized Traffic Control System) and the SCADA(Supervisory Control and Data Acquisition) system. Frame error rate and throughput are calculated and compared for the two cases that the CTC/SCADA has an extra error control scheme besides the internal CRC32 of the Ethernet and that no extra error control has been done. Simulation program is constructed with the Matlab and the Matlab GUI and then converted into a stand-alone c++ program to be used on any non-Matlab environments.

하기위한 FEC부, ARQ 방식에 따른 처리율(Throughput 또는 Link Utilization) 등 제반 성능들을 시뮬레이션 하기위한 ARQ/LU부, FEC와 ARQ를 결합하여 처리율(Throughput 또는 Link Utilization) 등을 시뮬레이션 하기위한 HARQ/LU부, 이더넷의 처리율과 지연시간 등을 시뮬레이션하기 위한 Ethernet부, 토큰-링의 방식별 처리율을 시뮬레이션하기 위한 Token Ring부, tolerance, 큐 등 일반적인 네트워크의 성능들을 시뮬레이션하기 위한 Tolerance부, 및 네트워크 적용사례에 대한 시뮬레이션을 위한 Exam부 등으로 구성되어, 각각 독립적으로 사용하거나 연계하여 시뮬레이션을 행할 수 있도록 작성하였다.

1. 서 론

컴퓨터를 통한 시뮬레이션은 철도 통신 제반 요소들에 대해 객관적인 검증 방식을 제공하는 한편, 향후 제안되어지는 방식들에 대한 비교분석 나아가 표준방식을 제공하는 효율적인 틀이 될 수 있다. 현재 철도환경에서의 다양한 정보의 수집과 전송을 위해서는 각 통신기간의 정보전송 네트워크화가 요구되고 향후 적용범위가 확대될 것으로 예상되어, 네트워크 기반 정보전송 기법들에 대한 사전 성능 분석 연구가 필요하다.

각 시뮬레이션 파트에 대한 실행은 기본적으로 해당 버튼을 클릭하면 필요한 파라미터 설정 다이얼로그가 생성되고, 다이얼로그상에서 필요한 파라미터들을 설정한 후 실행버튼을 누르면 수행되어 수행된 결과가 메인 화면상에서 플로팅 및 저장되도록 작성하였다.

본 연구에서는 철도 신호설비간 통신 프로토콜의 성능 분석을 위해 Matlab를 이용한 GUI 기반 시뮬레이션 프로그램을 개발하고 개발된 프로그램을 철도용 통신사태에 적용하여 유용성을 보였다. 일반적으로, 개발된 프로그램을 철도 선로변에 설치되는 LDTS와 EIS 사이의 인터페이스를 위한 점대점(peer-to-peer) 정보전송 프로토콜에 적용하여 성능분석을 행한 바가 있다.[1] 이후 프로그램의 성능을 네트워크 기반 정보전송 시뮬레이션이 가능하도록 확장하여, 본 논문에서는 네트워크 기반 철도용 정보전송 사례로서 이더넷(Ethernet)을 이용한 CTC-SCADA 시스템간 정보전송 네트워크를 선정하고, 이에 대한 성능분석 시뮬레이션을 행하였다.[2] 선정된 CTC-SCADA 시스템간 정보전송 사례에서는 Fail-safe 기능을 높이기 위해 별도의 에러제어(CRC16, SW-ARQ)가 CTC-SCADA에서 추가된다.[3],[4] 따라서 이더넷에서 기본적으로 사용되는 CRC32 이외에 CTC-SCADA에서 별도의 에러제어를 행하는 경우와 행하지 않는 경우 각각에 대해 프레임 에러율과 처리율을 구하고 결과를 비교분석하였다.

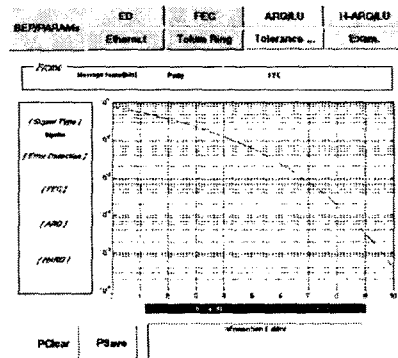


그림 1 시뮬레이션 프로그램 메인화면

2. 통신 프로토콜 시뮬레이션 프로그램

통신시스템 프로토콜 시뮬레이션을 행하기 위하여 그림 1과 같이 Matlab/Matlab GUI를 이용한 윈도우 기반 시뮬레이션 프로그램을 개발하였다.[5]

전체 프로그램은 데이터 시그널링 방식 및 AWGN를 파라미터로 하여 비트에러율을 계산하고 초기 기본 파라미터를 설정하는 BEP/PARAMs부, 에러검지 코드들을 시뮬레이션하기위한 ED부, FEC 코드들을 시뮬레이션

3. 이더넷 기반 CTC-SCADA간 정보전송 시뮬레이션

본 논문에서는 그림1의 시뮬레이션 프로그램을 네트워크 기반 철도용 정보전송 사례인 이더넷 기반 CTC-SCADA 시스템간 정보전송 네트워크에 적용하여 프레임에러율과 처리율에 대한 시뮬레이션을 행한 후, 결과를 분석하였다.

3.1 프레임 에러율

3.1.1 CTC/SCADA에서 에러제어를 하지 않는 경우 이더넷을 이용하여 데이터 전송을 행하는 경우, 그림 2에 보이는 바와 같이 CTC-SCADA간 전송데이터 N 바이트에 대해 통신 프로토콜상의 각 층에 의해서 약 84

바이트의 헤더들이 추가된다[6]. NIC(Network Interface Card)는 최종 프레임인 이더넷 프레임에 대해 CRC32 코드를 이용하여 에러검지를 수행하는데, N(전송 데이터)+72(부가되는 각종의 헤더들) (=N+84-12) 바이트에 대한 에러가 검지된다.(이더넷 프레임의 프리앰블(7바이트), SFD(1바이트), FCS(4바이트), 즉 12바이트는 에러 검지대상에서 제외됨.)

AWGN이 존재하는 BSC(Binary Symmetric Channel)에서 잡음에 따른 수신 메시지의 비트에러율(P_b)은

$$P_b = Q\left(\sqrt{\frac{E_b(1-\rho)}{N_0}}\right) \quad (1)$$

$\rho=0$: orthogonal(uniphase) signalling

$\rho=-1$: antipodal(biphase) signalling

E_b : signal bit energy N_0 : 밴드폭 1Hz당 잡음전력 Q : 에러함수

와 같고[7], CRC32 에러 검지코드에 대한 에러검지 성능은 BSC에 대해서[8]

$$P_{ud} \leq 2^{-k} [1 + (1 - 2P_b)^n - 2(1 - P_b)^n],$$

$$P_c = (1 - P_b)^n$$

$$P_d = 1 - P_c - P_{ud}, \quad P_{fc} = \frac{P_{ud}}{1 - P_d} \quad (2)$$

여기서, P_c : 프레임에러가 없을 확률, P_{ud} : 프레임에러를 검지 못할 확률, P_d : 프레임에러를 검지할 확률, P_{fc} : 프레임에러 발생 확률, P_b : 비트에러 발생 확률, n : 코드워드 비트수, k : 메시지워드 비트수

이 되므로, 식(2)를 이용하여 N(전송 데이터)+72(부가되는 각종의 헤더들)에 대한 각 에러율들을 구할 수 있다.

3.1.2 CTC/SCADA에서 에러제어를 하는 경우

Fail-safe 기능을 높이기 위해 NIC에서의 CRC32와는 별도로 CTC-SCADA에서 최종 수신데이터에 대한 에러 검지 및 흐름제어 기능이 추가된다. CTC-SCADA에서 전송되는 전체 데이터는 헤더부 3바이트와 데이터부 4+N 바이트로 구성되고(그림 2), 헤더부의 데이터길이 필드와 데이터부의 데이터-필드를 포함한 4+N 바이트에 대해 CRC16코드를 이용하여 에러 검지를 행한다.

NIC에서 일차로 이더넷 프레임에 대해 CRC32 코드를 이용하여 에러검지를 수행하므로, CTC-SCADA에서 CRC16에 의한 별도의 에러검지시 프레임에러가 발생할 확률은 CRC32 코드에서 프레임에러가 발생하는 확률(전(priori)확률조건으로 한다. 따라서, 비트에러가 전 프레임에 걸쳐서 랜덤하게 발생하는 경우, CRC32 코드에서 프레임에러가 발생된 후에 CRC16에 의해서도 프레임 에러가 발생될 확률, P_{fc} 는 각각의 프레임에러가 겹쳐져서

$$P_{fc} = P_{fc32}P_{fc16} \quad (3)$$

여기서, P_{fc16} : CRC16에서 프레임에러 발생확률, P_{fc32} : CRC32에서 프레임에러 발생 확률

를 최소값으로 갖게 된다. 이는 CRC16을 추가함에 따라, 프레임 에러율을 떨어뜨릴 수 있게 됨을 나타낸다.

3.2 프레임 처리율

CTC-SCADA에서의 흐름제어는 단순한 SW-ARQ방식으로, 3회까지 전송이후 ACK가 수신되지 않으면 데이터 전송을 중지하고 경고신호를 발생시킨다. 에러가 발생하

는 경우, SW-ARQ에서 평균 메시지 전송수는

$$N_R = (1 - P_d) + 2P_d(1 - P_d) + 3P_d^2(1 - P_d) + \dots$$

$$= \frac{1}{1 - P_d} \quad (4)$$

P_d : 에러 검지코드의 에러 검지 확률

이 되고, 처리율은 다음과 같이 결정된다.

$$\eta_{SW} = \frac{k}{(n + n')N_R} \quad (5)$$

n' = (2*전과제연시간 + 디코딩, 제어정보 전송 등 기타 지연시간) * 데이터레이트

여기서, n 은 코드워드 비트수, k 는 메시지워드 비트수를 나타내고, n' 는 메시지 전송을 제외한 전체 지연시간 동안의 전송 가능한 비트수를 나타낸다. 식(4)에서 P_d 는 CTC-SCADA에서 CRC16에 의한 에러 검지 확률로, NIC에서의 CRC32 코드에서 프레임에러가 발생되는 확률을 전제조건으로 하므로, CTC-SCADA에서 프레임 에러가 검지될 확률, P_d' 는

$$P_d' = P_{fc32}P_d \quad (6)$$

P_{fc32} : CRC32의 프레임에러 발생 확률, P_d : CRC16의 에러 검지 확률

이 되고, 이를 식(4)의 P_d 대신에 삽입하면 평균 메시지 전송회수는

$$N_R' = \frac{1}{1 - P_d'} \quad (7)$$

이 된다. 한편, 이더넷 프로토콜에 의해 최대 처리율이 다음과 같이 정해지므로[9]

$$\eta_{max} = \frac{t_p}{t_p + t_{prop} + 2et_{prop}} = \frac{1}{1 + (1 + 2e)a}$$

$$= \frac{1}{1 + (1 + 2e)(Rd/VL_p)} \quad (a = t_{prop}/t_p) \quad (8)$$

최종적인 처리율식은 식(5)에서 (8)로부터 다음과 같이 결정된다.

$$\eta_{SW} = \eta_{max} \frac{k}{(n + n')N_R'} \quad (9)$$

3.3 시뮬레이션

3.3.1 프레임 에러율

CTC/SCADA에서의 전송하는 데이터의 바이트수를 5->50->500 바이트로 증가 시키면서 CTC/SCADA에서 에러제어를 하지 않는 경우와 하는 경우에 대해, AWGN 증가에 따른 프레임에러율에 대한 시뮬레이션을 행하였다. 바이트수를 5->50->500 바이트로 증가시키는 경우, 총72바이트의 헤더바이트가 부가되어(프리앰블(7바이트), SFD(1바이트), FCS(4바이트) 제외) 이더넷 프레임은 총 77->122->572 바이트가 되고, 이에 대한 CRC32가 수행된다. 따라서 코드워드 바이트수는 CRC32 4바이트를 포함하여 (77->122->572) + 4가 되고, 메시지 바이트 수는 77->122->572가 된다.

Biphase 신호를 가정하는 경우, 식(1)과 (2)를 이용하여 구한 CRC32 코드에 의한 이더넷 프레임의 프레임에러율을 보이면 그림 3과 같다. CTC/SCADA에서 에러제

어를 하여, 메시지의 데이터 필드의 바이트수를 5->50->500로 증가시키는 경우, 각각에 대해 메시지 길이에 해당하는 2바이트와 시퀀스번호, 메시지 형태와 관련된 2바이트가 부가되어 CRC16이 행하여진다. 즉 코드워드 바이트 수는 (5->50->500) + 6가 되고, 메시지 바이트 수는 CRC16 2바이트를 제외하면 (5->50->500) + 4가 된다. 그림 4는 메시지의 바이트를 5->50->500로 증가시키며 전송하는 경우, CTC/SCADA의 CRC16과 이더넷의 CRC32를 모두 고려한 프레임 에러율을 보인다. 식(3)에 의해 각 경우에 대한 프레임에러가 곱해져서 전체적으로 프레임에러가 낮아지게 된다.

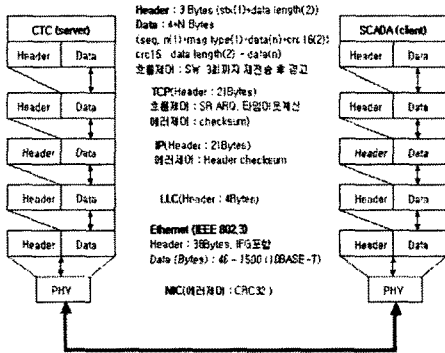


그림 2 CRC와 SCADA사이의 데이터 전송 프로토콜 개략도

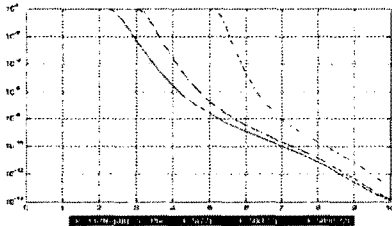


그림 3 CTC/SCADA에서 에러제어를 하지 않고, 메시지의 바이트를 5->50->500로 증가시키며 전송하는 경우, CRC32에 의한 이더넷 프레임 에러율

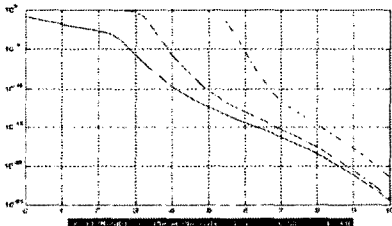


그림 4 메시지의 바이트를 5->50->500로 증가시키며 전송하는 경우, CTC/SCADA의 CRC16과 이더넷의 CRC32를 모두 고려한 프레임 에러율

3.3.2 프레임 처리율

CTC/SCADA에서 전송 데이터의 바이트수를 5->50->500 바이트로 증가시키는 경우, 앞 절에서 설명한 바와 같이 코드워드 바이트수는 CRC32 4바이트를 포함하여 (77->122->572) + 4가 되고, 메시지 바이트 수는 77->122->572가 된다. CTC/SCADA에서의 전송하는 데이터의 바이트수를 5->50->500 바이트로 증가시키면 앞절에서 설명한 바와 같이 총72바이트의 헤더 바이트가 부가되어 이더넷 프레임은 총 77->122->572 바

이트가 되는데, 여기에서 MAC 주소와 목적지 어드레스 각각 6바이트, MAC 형식에 해당하는 2바이트를 제외하면 이더넷 프레임의 Payload가 된다. 따라서 Payload는 63->108->558 바이트가 된다. Payload가 63->108->558 바이트이고, 링크길이가 1Km인 경우 10Base-T 이더넷의 최대 처리율, η_{max} 를 식(8)로부터 구하면 각각 0.751, 0.806, 0.944가 된다. 그림 5는 식(9)를 이용하여 처리율을 계산한 후 도시한 그림이다. 계산시 지연요소 (n')에는 전파지연과 관련한 성분만 첨가하였다. 시뮬레이션결과 전송메시지가 500 바이트인 경우 Eb/No가 5dB 이하에서는 처리율이 거의 0으로 매우 낮은 값을 가지나, Eb/No가 6dB 이상으로 증가하면 처리율이 급격히 증가하여 0.9이상이 됨을 알 수 있다.

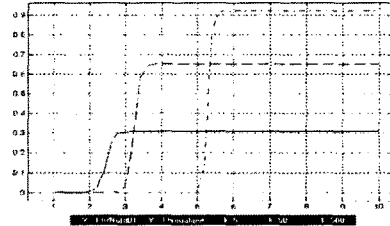


그림 5 메시지의 바이트를 5->50->500로 증가시키며 전송하는 경우, CTC-SCADA-이더넷 네트워크 처리율

4. 결론

본 논문에서는 네트워크 기반 철도용 정보전송 사례로서 이더넷을 이용한 CTC-SCADA 시스템간 정보전송 네트워크를 선정하고, 이에 대한 성능분석 시뮬레이션을 행하였다. 이더넷에서 기본적으로 사용되는 CRC32 이외에 CTC-SCADA에서 별도의 에러제어를 행하는 경우와 행하지 않는 경우 각각에 대해 프레임 에러율과 처리율을 구하고 결과를 비교 분석하였다. 시뮬레이션 결과, CTC/SCADA와 이더넷의 순차적인 에러검지 기능에 의해 전체적인 프레임에러가 현저하게 낮아짐을 확인하였고, 전송메시지가 500 바이트인 경우 Eb/No가 5dB 이하에서는 SW-ARQ 처리율이 거의 0으로 매우 낮은 값을 가지나, Eb/No가 6dB 이상으로 증가하면 처리율이 급격히 증가하여 0.9이상이 됨을 알 수 있었다.

[참고 문헌]

- [1] "GUI 기반 철도용 통신 프로토콜 시뮬레이션 프로그램 개발", 철도기술개발 연구사업 2차년도 연구보고서, 한국철도기술연구원, 2003.
- [2] "열차집중제어장치(CTC)와 전철전력 원격감시제어장치(SCADA)간 정보전송방식(Protocol)", 시험규격, 한국철도기술연구원, 2004.
- [3] S. B. Wicker, *Error Control System for Digital Communication and Storage*, Prentice Hall, pp. 99-128, 1995.
- [4] R. E. Ziener, *Introduction to Digital Communication*, Prentice Hall, pp. 504-510, 2001.
- [5] "Using Matlab Version 6", Math Works, 2000. 2.
- [6] IEEE 802.1D 1998, *Local Area Network MAC(Media Access Control) Bridges*, may 25, 1998.
- [7] B. Sklar, *Digital Communications Fundamentals and Applications*, Prentice Hall, pp. 104-136, 2001.
- [8] T. Kasami, T. Klove, and S. Lin, "Error detection with linear block codes", *IEEE Trans. Inform. Theory*, vol. IT 29, pp. 131-136, Jan. 1983.
- [9] G. Keiser, *Local Area Networks*, McGraw-Hill, New York, 2nd ed., pp. 147-148, 2002.