

차량내 통신을 위한 EtherCAT 네트워크의 전송지연 및 고장복구 특성 분석

김 동 길*, 조 영 현*, 이 동 익°

Analysis of Transmission Delay and Fault Recovery Performance with EtherCAT for In-Vehicle Network

Dong-gil Kim*, Youngyun Jo*, Dongik Lee°

요 약

IT기술의 발전에 힘입어 지능형 센서 및 지능형 액추에이터 채택이 증가하면서 차량내 통신 네트워크를 통한 데이터 전송은 꾸준한 증가 추세를 보이고 있다. 2015년에는 차량내 통신 네트워크를 통한 전송 데이터량이 2010년 대비 2배 이상 증가하며, 차량제어에 필요한 네트워크 노드 수는 2010년 대비 1.5배 이상 증가할 것으로 전망된다. 이와 같이 차량내 데이터량의 증가가 예상됨에 따라 최근 자동차 산업계에서는 차량용 통신 네트워크로서 EtherCAT, TTEthernet 등 산업용 Ethernet에 대한 관심이 증대되고 있다. 본 논문에서는 차량내 데이터 전송을 위한 산업용 Ethernet으로 주목받고 있는 EtherCAT의 전송지연 특성과 고장복구 특성을 분석할 수 있는 모델을 제안한다. 실험용 EtherCAT 네트워크를 구성하여 제안한 모델의 정확성을 검증하였다.

Key Words : In-Vehicle Network, EtherCAT, Industrial Ethernet, Transmission Delay, Fault Recovery

ABSTRACT

Thanks to progressive development of IT technology, the number of intelligent devices communicating each other through an In-Vehicle Network (IVN) has been steadily increasing. It is expected that the required network bandwidth and network nodes for vehicle control in 2015 will be increased by two times and one and half times as compared to in 2010, respectively. As a result, many researchers in automotive industry has showed a significant interest on industrial Ethernets, such as EtherCAT and TTEthernet. This paper addresses an analysis on transmission delay and fault recovery performance with an EtherCAT network which is being considered as an IVN. A mathematical model based on the analysis is verified through a set of experiments using an experimental network setup.

I. 서 론

현재 자동차 산업은 기존의 기계공학 중심으로 개발되어 온 자동차에 IT를 융합하여 주행안전성과 편의성을 대폭 향상시킨 지능형 자동차의 개발에

집중하고 있다. 지능형 자동차는 전통적인 기계부품들이 지능형 센서와 지능형 액추에이터로 대체되고, 텔레매틱스 기능과 멀티미디어 기능이 강화된다. 그 결과 2015년에는 자동차에서 사용하는 차량내 통신 네트워크(In-Vehicle Network, IVN)를 통한 전송

※ 본 연구는 지식경제부 및 정보통신산업진흥원의 IT융합 고급인력과정 지원사업의 연구결과로 수행되었음(NIPA-2012-H0401-12-1006)

• 주저자 : 경북대학교 IT대학 전자공학부 고신뢰성 임베디드 제어 연구실, eastroad@ee.knu.ac.kr, 학생회원

° 교신저자 : 경북대학교 IT대학 전자공학부 고신뢰성 임베디드 제어 연구실, dilee@ee.knu.ac.kr, 정회원

* LG전자 CTO 부분

논문번호 : KICS2012-05-248, 접수일자 : 2012년 5월 14일, 최종논문접수일자 : 2012년 10월 29일

데이터는 2010년 대비 2배 이상 늘어나며, 차량제어에 필요한 네트워크 노드 수는 2010년 대비 1.5배 이상 증가할 것으로 예상된다^[1].

차량제어를 위한 데이터 전송량이 증가함에 따라 기존의 CAN 중심에서 탈피하여 FlexRay, MOST 등 새로운 네트워크 프로토콜이 적용되고 있다. 그러나 FlexRay 및 MOST의 최고 전송속도는 각각 10Mbit/s, 25Mbit/s로서, 급속히 증가하는 데이터 전송량을 처리하기에는 한계가 있다. 따라서 이러한 전송속도의 한계를 극복하기 위해, 그림1에 제시한 것처럼, 최고 1Gbit/s의 전송속도를 지원하는 Ethernet을 이용하여 차세대 차량내 통신 네트워크를 구축하려는 연구가 진행 중이다^[1-3].

지능형 자동차에 적합한 차량용 네트워크는 저렴한 가격, 빠른 속도, 확정적인 통신 특성, 유지보수성, 확장성 등을 만족해야 한다^[1]. Ethernet은 대역폭 사용률이 높고, 최고 1Gbit/s의 속도로 데이터를 전송할 수 있다. 하지만 Ethernet은 CSMA/CD 기반의 MAC을 사용하기 때문에 통신의 확정성이 낮고, 시간동기를 지원하지 않기 때문에 자동차와 같은 안전이 중요한 시스템(Safety-Critical System)에 적용하기에는 어려움이 있다^[4]. 이러한 Ethernet의 단점을 해결하기 위해 많은 연구가 이루어졌으며, 그 가운데 EtherCAT은 자동화, 로봇 등에서 폭넓게 적용되고 있다^[5]. EtherCAT 프로토콜의 확정적 데이터 전송, 시간동기, 고장대처, 유연성 및 표준 Ethernet과의 호환성 등을 고려할 때, EtherCAT 프로토콜은 차량용 통신 네트워크로 충분한 대안이 될 수 있을 것으로 기대된다. 그 결과 EtherCAT을 이용한 모터 드라이브의 중단간 지연에 대한 연구^[6] 및 PROFINET과의 성능비교에 대한 연구^[7] 등이 이루어지고 있다. 그럼에도 불구하고, EtherCAT을 차량제어에 적용하기 위해서 반드시 필요한 전송지연 특성 및 고장복구 성능분석 등에 대한 연구는 매우 부족한 상태이다.

자동차를 비롯한 네트워크 기반 제어시스템(NCS, Networked Control System)에서 네트워크의 전송지연은 제어관점에서 매우 중요한 요소로써, 과도한 전송지연은 전체 시스템의 성능 및 안정성에 나쁜 영향을 미치는 것으로 알려져 있으며(그림2), 이를 보상하기 위한 제어기 설계기법 연구가 널리 이루어지고 있다^[8]. 일반적으로 네트워크의 전송지연은 프로토콜의 특성에 따라 다양하게 나타나며, MAC, 네트워크에 연결된 노드 수, 메시지 발생 빈도, 그리고 메시지 길이 등과 같은 것에 영향을 받는다.

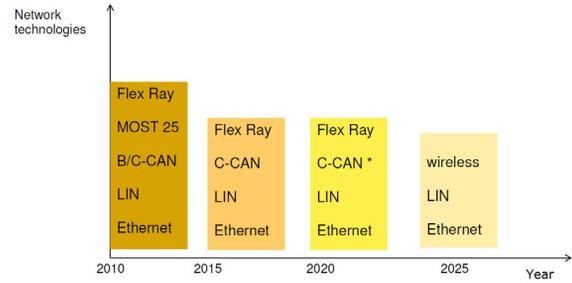


그림 1. 차량용 네트워크 아키텍처 로드맵^[3]
Fig. 1 Roadmap of system architecture in vehicle

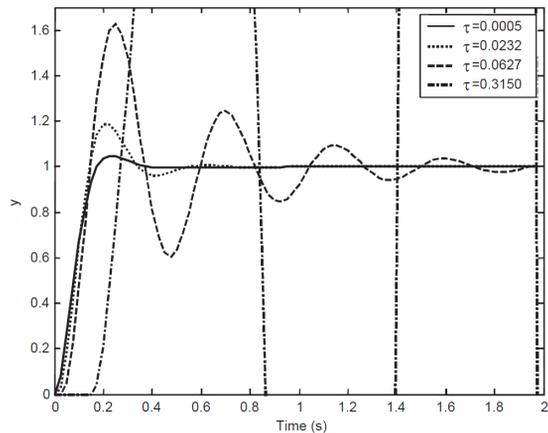


그림 2. 제어루프의 전송지연에 의한 시스템의 성능 저하^[8]
Fig. 2. System performance degradations caused by delay in-the-loop

대표적인 IVN 프로토콜인 CAN은 버스 트래픽이 증가함에 따라 평균 전송지연 시간이 급격하게 증가하는 것으로 알려져 있으며^[9], 차세대 차량용 통신 표준 프로토콜 주목받고 있는 FlexRay의 경우에는 전송지연(최악응답시간) 모델이 보다 복잡하게 다음과 같이 주어진다^[10]:

$$R_m(t) = \sigma_m + w_m(t) + C_m \quad (1)$$

$$\sigma_m = T_{bus} - (ST_{bus} + FrameID_m \times dgMinislot)$$

$$w_m(t) = ST_{bus} + pLatesTx_{N_p} \times dgMinislot$$

$$C_m = Frame\ size(m) / bus\ speed$$

여기서 a_m 은 버스 사이클의 길이에 의해서 발생하는 최대 지연을 의미하고 $w_m(t)$ 는 정적스케줄링 및 동적스케줄링의 메시지 우선순위에 의해서 발생하는 지연을 나타낸다. 마지막으로 C_m 은 데이터 전송지연 시간이다.

대중적으로 널리 쓰이는 프로토콜인 Ethernet의 전송지연 또한 버스 트래픽이 증가함에 따라 기하

급수적으로 증가하는 것으로 알려져 있으며, 일반적인 패킷 스위칭 네트워크에서의 전송지연은 다음과 같이 주어진다¹¹⁾:

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop} \quad (2)$$

여기서 d_{proc} 는 데이터 비트오류 체크 및 링크를 확인하는데 소모되는 시간을 의미하고 d_{queue} 는 버퍼에서 대기하는 시간 및 MAC에서 발생하는 시간을 나타낸다. d_{trans} 는 데이터 전송지연 시간으로 데이터 크기와 전송속도에 의해서 결정된다. 마지막으로 d_{prop} 는 전파지연 시간을 의미한다. 이와 같이 네트워크의 전송지연은 프로토콜의 특성 및 네트워크 구성에 따라 다양하게 나타나므로, 프로토콜의 특성을 정확히 반영한 전송지연 모델에 대한 연구는 매우 중요하다.

본 논문에서는 EtherCAT 네트워크의 전송지연 및 고장복구 모델을 제시하고 실험용 네트워크를 이용하여 제안한 모델의 효용성을 검증하였다. EtherCAT 전송지연 모델은 Prytz에서 연구된 바 있으나⁷⁾, 본 논문에서는 좀 더 명확하고 직관적인 전송지연 모델을 제시하였다. Prytz의 연구에서는 EtherCAT 슬레이브 노드에서 발생하는 지연 시간을 약 1초 정도로 감안하였으나, 본 논문에서는 보다 구체적으로 물리계층에 따른 슬레이브 노드의 전송지연 시간을 제시하였다. 아울러 라인 토폴로지에 대한 전송지연 시간 모델을 직관적으로 제시하였다.

EtherCAT 고장복구 모델은 슬레이브 노드가 고장상태에서 복구되어 통신이 가능할 때까지 소요되는 지연시간에 대한 모델로서, 비정상적 동작특성을 나타내는 EtherCAT 슬레이브 노드에 대한 EtherCAT 마스터의 고장대처 메커니즘을 분석하여 도출하였다.

본 논문의 구성은 다음과 같다. 서론에 이어 2장

에서 EtherCAT 프로토콜에 대하여 전반적인 설명을 하고, 이를 기반으로 EtherCAT 프로토콜의 전송지연 및 고장 복구 모델을 제시한다. 3장에서는 EtherCAT 네트워크를 이용한 실험 결과를 제시하고, 마지막으로 4장에서 결론을 맺는다.

II. 본 론

2.1. EtherCAT 프로토콜

EtherCAT은 2002년 독일 Beckhoff사에서 개발되었으며, ETG(EtherCAT Technology Group)를 결성하여 기술을 공개한 개방형 산업용 필드버스이다. 현재 자동화 시스템 분야에서 널리 쓰이고 있으며, ISO 규격(ISO15745-4), IEC 규격(IEC/PAS 62407)으로 인증된 국제 표준이다.

EtherCAT은 스타, 라인, 트리 토폴로지를 지원한다. 물리적인 토폴로지에 관계없이 EtherCAT은 논리적으로 링 토폴로지 형태로 네트워크를 구성하며, 데이터 프레임은 마스터 노드에서 생성되어 토폴로지에 연결된 모든 슬레이브 노드를 거쳐 다시 마스터 노드로 복귀하고, 이로써 하나의 통신 사이클이 완료된다. 그림3은 EtherCAT 네트워크의 간단한 구성 예를 보여준다.

EtherCAT의 데이터 전송은 마스터에서 생성한 데이터 프레임이 슬레이브 노드를 통과하는 동안 해당 슬레이브 노드는 데이터를 실시간으로 읽고 쓰는 방식(on-the-fly)으로 이루어진다. 이는 매우 독특한 데이터 전송 기법으로 각 노드에서 발생할 수 있는 지연을 최소화하는 효과뿐만 아니라, 노드에서 발생하는 지연시간을 확정적인 특성을 가지도록 한다. 이를 위해서는 각 슬레이브 노드에 2개의 표준 Ethernet 포트가 필요하며 실시간으로 데이터를 읽고 쓰기 위한 특수한 하드웨어(EtherCAT Slave Controller)가 필요하다¹²⁾(그림4). EtherCAT의 전송지연 시간을 모델링할 경우 이러한 특성을 반드시 반영하여야 한다. 또한 메시지 폴링(Polling), 시분

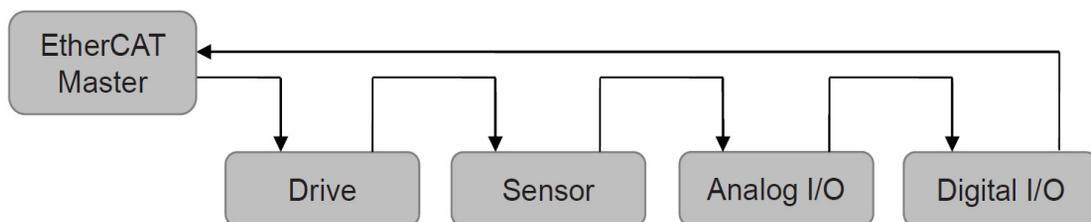


그림 3. EtherCAT 네트워크 구성 예
Fig. 3. Example of EtherCAT network

할(Timeslicing), 브로드캐스트(Broadcast)와 같은 전통적인 통신방식과 비교할 때 EtherCAT의 대역폭 사용량은 아주 높다. EtherCAT 네트워크에 사용할

다. 또한 하나의 표준 Ethernet 프레임 안에 여러 개의 EtherCAT Datagram이 들어갈 수 있기 때문에 분산처리에 효과적이고 네트워크를 효율적으로 사용할 수 있다.

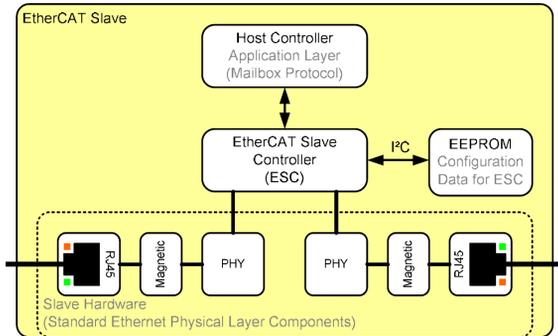


그림 4. EtherCAT 슬레이브 구조^[12]
Fig. 4. EtherCAT slave structure

네트워크 기반의 제어 시스템이 정확하게 동작하기 위해서는 높은 수준의 시간 동기가 매우 중요하다. EtherCAT은 시간 동기 오차를 최소화하기 위해 IEEE 1588 기반의 시간 동기 알고리즘을 사용한다. EtherCAT 슬레이브 노드는 데이터 프레임이 들어올 때와 나갈 때의 시간차이를 측정할 수 있다. 마스터는 이 값들을 이용하여 개별 노드에 대한 전송 지연시간을 계산할 수 있다^[14]. 이러한 방법으로 EtherCAT은 1μs 이하의 낮은 시간동기 오차를 유지한다.

2.2. 전송지연 및 고장복구 모델

2.2.1. 전송지연 모델

EtherCAT의 전송지연 시간을 모델링하기 위해, EtherCAT 전송지연 시간을 다음과 같이 정의하였다:

- 마스터에서 전송한 데이터 프레임이 슬레이브 노드를 거쳐 다시 마스터로 돌아올 때까지 소요 시간,
- 전방향 지연시간(t_f)과 후방향 지연시간(t_b)의 합으로 표현.

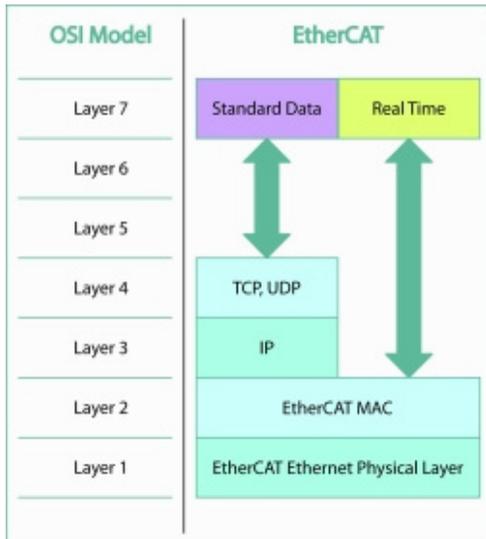


그림 5. EtherCAT OSI 계층
Fig. 5. EtherCAT OSI Layer

앞서 설명한 것과 같이 EtherCAT은 마스터 노드에서 생성된 데이터 프레임이 각 슬레이브 노드를 통과하고 다시 마스터 노드로 복귀함으로써 하나의 통신 싸이클이 완료된다. 따라서 EtherCAT의 전송 지연 시간을 모델링하기 위해서는 각 슬레이브 노드에서 발생하는 지연시간(노드지연)이 매우 중요하며 그 외의 프레임 오버헤드, 전송지연, 그리고 전파지연 등을 고려해야 한다. 프레임 오버헤드는 순수한 전송 데이터를 제외한 데이터 프레임을 완성하기 위해서 필요한 데이터에 의해 발생하는 전송 지연의 일종이다. 전송지연은 전송하고자 하는 데이터의 페이로드 크기와 프로토콜의 전송속도에서 비롯된 지연이고, 전파지연은 노드와 노드 사이의 거리에 따른 지연이다. 노드지연은 데이터 프레임이 EtherCAT 슬레이브 노드를 거치면서 발생하는 지연으로, 노드에서 사용하는 물리계층에 의존한다. EtherCAT의 물리계층은 LVDS(EBUS) 및 IEEE

수 있는 허브나 스위치의 개수에는 명시적인 제약이 없으며 최대 65535개의 노드를 연결하여 네트워크를 구성할 수 있다.

EtherCAT은 IEEE 802.3에 따른 표준 Ethernet 프레임을 사용한다. EtherCAT 프레임과 Ethernet 프레임을 구분하기 위해 EtherCAT 프레임은 Ethernet Header의 EtherType을 0x88A4로 사용한다^[13](그림5). 따라서 EtherCAT은 일반 Ethernet 장비와 함께 하나의 네트워크를 구성할 수 있다. 그리고 EtherCAT 마스터는 Ethernet용 표준 NIC를 사용하기 때문에 저렴하게 네트워크를 구성할 수 있

802.3 표준 물리계층이 있으며, 각 물리계층에 따른 노드지연은 $0.3\mu s(t_{LVDS})$ 와 $1.2\mu s(t_{MII})$ 으로 알려져 있다¹⁴⁾. 따라서 EtherCAT 네트워크의 전송지연 시간 (t_l) 모델은 다음과 같이 주어진다:

$$t_l = t_{WD} \times l_W + t_B \times l_F + \sum_{i=1}^n t_{MII} + \sum_{j=1}^m t_{LVDS} + t_O \quad (3)$$

여기서 $t_{WD}(5ns/m)$ 및 l_W 은 각각 전파속도와 케이블 길이이고, $t_B(100Mbps$ 전송속도 적용시 $80ns/byte)$ 및 l_F 는 전송속도와 데이터의 페이로드 크기이다. $t_{MII}(1.2\mu s)$ 및 n 은 각각 IEEE 802.3 물리계층을 사용한 노드의 지연시간과 슬레이브 노드의 수를 나타낸다. $t_{LVDS}(0.3\mu s)$ 및 m 은 각각 LVDS 물리계층을 사용한 노드의 지연시간과 슬레이브 노드의 수를 나타낸다. 그리고 $t_o(40byte \times 80ns/byte = 3.2\mu s)$ 는 Ethernet 데이터 패킷 오버헤드에 따른 값이다.

FlexRay의 전송지연 모델과 달리, EtherCAT 네트워크의 전송지연 모델은 식(3)과 같이 상대적으로 간략히 표현된다. 식(3)처럼 전송지연 시간이 확정으로 주어지는 것은 매우 중요한 장점으로, 이는 네트워크 기반 제어시스템의 설계를 용이하게 한다. 한편, Ethernet 전송지연의 가장 큰 요소를 차지하는 MAC이 EtherCAT의 전송지연에는 영향을 미치지 않는 것은 EtherCAT의 프로토콜의 데이터 프레임 기반의 통신방식 및 on-the-fly 특성 때문이다.

2.2.2 고장복구 과정

EtherCAT 마스터는 데이터 프레임의 Working Counter(WC)값을 이용하여 고장발생 여부를 판별한다. 마스터는 슬레이브 노드에 전송하는 데이터의 종류에 따라서 정상적인 통신이 이루어졌을 때의 WC값을 알고 있으며, 만약 예상한 값과 다른 WC값을 수신하면 마스터는 슬레이브 노드에 고장이 발생한 것으로 간주한다. 마스터는 일부 슬레이브 노드에 고장이 발생하더라도 정상적으로 동작하는 슬레이브 노드에는 계속해서 데이터를 전송한다. 그리고 고장이 발생한 노드가 복구되었는지를 확인하기 위해 마스터는 고장이 발생한 노드와 인접한 노드의 데이터링크 상태 레지스터를 주기적으로 확인한다.

EtherCAT 네트워크에 연결된 슬레이브 노드가 고장에서 복구되면, 복구된 노드와 인접한 노드의

물리 계층은 복구된 노드와의 케이블 연결 상태를 확인한다. 정상 노드는 케이블 연결이 확인되면, 데이터링크 상태 레지스터의 Loopback 설정을 수정하고 마스터 노드의 응답을 기다린다. 데이터링크 상태 레지스터를 주기적으로 확인하는 마스터 노드는 데이터링크 상태 레지스터의 값이 수정된 것을 확인하고, 수정된 값이 영구적으로 수정된 것인지 확인하기 위해 특정시간(t_w) 동안 데이터링크 상태 레지스터를 계속 읽어온다. 특정시간 후에도 데이터링크 상태 레지스터의 값이 수정된 값으로 남아 있다면 마스터는 슬레이브 노드가 고장에서 복구되었음을 확인하고, 고장이 발생한 노드와 그 뒤에 연결된 노드의 초기화를 진행한다. 슬레이브 노드 초기화는 여러 사이클에 소모되며, 마스터 노드는 파이프라인 방법을 이용하여 한 사이클씩 순차적으로 슬레이브 노드를 초기화를 진행한다. 초기화에 필요한 사이클 횟수는 마스터 구현방식에 따라 다르며, 본 논문에서 사용한 TwinCAT 마스터의 초기화 사이클은 총 28사이클로 이루어져 있다.

2.2.3. 고장복구 모델

앞에서 분석한 고장복구 과정을 토대로 고장에서 복구된 후 통신이 재개될 때까지 걸리는 고장복구 지연 시간(t_r) 모델을 다음과 같다.

$$t_r(n, m, s) = t_{ld}(s) + t_w(m) + t_{si}(n, m) \quad (4)$$

$$t_{si}(n, m) = (\text{number of init cycle} + n) \times \text{cycle time}$$

n = 슬레이브 노드의 논리적 번호

m = 마스터 노드 종류

s = 슬레이브 노드 종류

여기서, t_{ld} 는 슬레이브 노드에서 인접한 노드의 케이블 연결을 확인하기 위한 시간으로 슬레이브 노드의 구현에 따라 변동되고, t_w 는 마스터 노드가 슬레이브 노드의 데이터링크 레지스터 값을 확인하기 위한 소요 시간으로 마스터 노드 구현 방식에 의존한다. 마지막으로 t_{si} 는 슬레이브 노드의 초기화를 위한 시간으로, 초기화 사이클 횟수와 통신주기가 곱으로 주어지고, n 은 네트워크에 연결된 노드의 논리적인 번호를 의미한다.

t_{ld} 값과 t_w 값은 각각 슬레이브 노드 구현 방식 그리고 마스터 노드 구현 방식에 의존할 것이라는 것을 알 수 있다. 본 논문에서 사용한 마스터 노드와

슬레이브 노드는 각각 TwinCAT 및 FB1111-0142 이다. 그러나 t_{ld} 값과 t_w 값에 대한 정보를 확인할 수 없어 wireshark를 이용한 패킷 분석을 통하여 t_{ld} 값과 t_w 값을 추정하였다. 그 결과는 t_{ld} 값은 대략 5초 정도이며 측정값의 신뢰도를 위해 50여 차례 실험을 수행한 결과, t_{ld} 는 $N(5, 0.2^2)$ 의 정규분포를 보였다. 또한 t_w 값은 대략 1초 정도이며, t_w 는 $N(1, 0.01^2)$ 의 분포를 보였다. 이 결과는 TwinCAT 및 FB1111-0142에 대한 것으로 이외에 다른 종류의 마스터 또는 슬레이브 노드를 사용할 경우 값은 변동이 있을 것으로 예상된다.

식(4)에서 확인할 수 있듯이, EtherCAT의 고장복구 지연 시간은 물리 계층의 성능, 통신주기, 복구 시간을 측정할 노드의 위치, 마스터 구현에 따른 초기화에 필요한 사이클의 수, 그리고 데이터 링크 상태 레지스터를 확인하는 데 걸리는 시간에 의존해서 변동된다.

III. 실험

3.1. EtherCAT 실험 네트워크

본 논문에서 제안한 모델을 검증하기 위해서 실험용 EtherCAT 네트워크를 구현하였다. 실험용 네트워크는 TwinCAT을 마스터 프로그램으로 사용하고, 슬레이브 노드는 FB1111-0142 보드를 이용하여 제작하였다. FB1111-0142 보드는 RJ45 커넥터, 마그네틱, Ethernet PHY 칩은 MICREL사의 KSZ8721BL을 사용하였고, ET1100[15] 칩을 데이터링크 계층으로 사용하였다. 어플리케이션 계층은 ET1100칩의 디지털 I/O를 이용하였으며, 어플리케이션 계층에서 발생하는 오버헤드를 최소화하여 EtherCAT 프로토콜 고유의 통신특성을 파악하였다. 데이터의 송수신을 확인하기 위해 LED를 연결하였고, FB1111-0142 보드를 구동하기 위한 주변회로를 설계하여 PCB로 제작하였다(그림6).

3.2. 전송지연 실험

3.2.1. 전송지연 실험 구성

EtherCAT 마스터는 통신 주기마다 각 슬레이브 노드의 LED를 점멸하는 메시지를 전송한다. 1번 노드와 28번 노드의 출력을 오실로스코프에 연결하여, 두 노드의 LED가 켜지는 시간의 차이를 단방향 전송지연 시간으로 간주하였다(그림7).

3.2.2 전송지연 실험 결과

식(3)에 의하면, 1번 노드와 28번 노드 사이의 단방향 전송지연 시간(t_f)은 다음과 같이 주어진다:

$$t_f = \frac{1}{2}t_l \cong t_{WD} \times l_w + \sum_{i=1}^n \frac{t_{MMI}}{2} \cong \sum_{i=1}^{28} \frac{t_{MMI}}{2} = \sum_{i=1}^{28} 0.6 = 16.8us.$$

그림8은 전송지연 실험의 측정결과를 보여준다. 그림8의 ① 시점에서 노드 1의 LED가 켜지고 ②

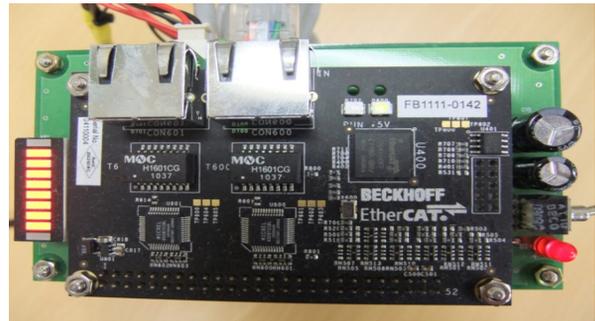


그림 6. EtherCAT 슬레이브 노드
Fig. 6. EtherCAT slave node

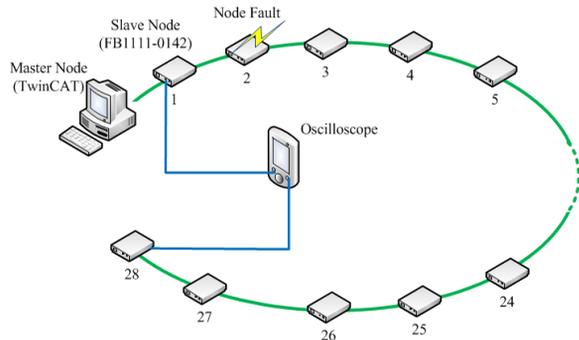


그림 7. EtherCAT 실험 구성
Fig. 7. EtherCAT experimental setup

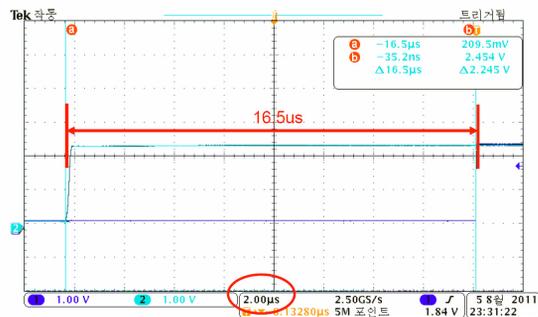


그림 8. 전송지연 시간 측정 결과
Fig. 8. Measurement result of transmission delay

시점에서 노드 28의 LED가 켜진다. 따라서 실험용 네트워크를 이용한 실험 결과는 약 16.5 μ s이며, 전송시간 모델과의 오차는 0.3 μ s(1.8%)이다. 식(3)에서 제시한 모델이 EtherCAT 네트워크의 전송 지연 시간 특성을 잘 나타내고 있음을 확인할 수 있다.

3.3. 고장복구 지연 실험

3.3.1. 고장복구 지연 실험 구성

EtherCAT 슬레이브 노드의 고장은 EtherCAT 통신 칩 고장, 케이블 단선, I/O 오류, 어플리케이션 계층과 통신 불능 등으로 분류할 수 있다. 본 실험에서는 슬레이브 노드의 전원이 차단되는 것을 고장으로 가정하였고, 노드의 전원이 복구되는 것을 고장에서 복구되는 것으로 가정하였다.

슬레이브 노드가 정상적으로 동작할 경우 마스터 노드의 명령에 의해서 슬레이브 노드의 디지털 출력은 LOW가 되도록 설정하고, 슬레이브 노드와 마스터 노드의 통신이 불가능할 경우 디지털 출력은 HIGH가 되도록 설정한다. 그리고 단방향으로 연결된 EtherCAT 네트워크에서 임의의 슬레이브 노드에서 고장이 발생하면, 고장이 발생한 노드 뒤에 연결된 노드도 통신이 불가능하게 되며, 이는 디지털 출력을 통하여 확인할 수 있다. 실험은 다음과 같이 수행하였다.

- 노드 2의 전원을 차단하여 고장 인가,
- 일정 시간 후, 노드 2에 전원을 인가하여 고장 복구,
- 고장이 복구된 후, 노드28의 통신이 재개될 때까지의 시간을 오실로스코프로 측정.

EtherCAT 고장복구 시간은 마스터 노드의 데이터 링크 상태 레지스터 변경 대기 시간과 초기화 사이클 수에 따라 달라진다. 실험에 사용한 TwinCAT의 데이터 링크 상태 레지스터 변경 대기시간은 평균 1초이고 EtherCAT 슬레이브 노드의 초기화는 28사이클이 소모된다. 그리고 표준 Ethernet에서 케이블 연결을 확인하는데 소모되는 시간은 5초 정도였다. 또한 고장복구 지연 시간은 통신주기와 측정할 노드의 위치에 따라 달라진다. 본 실험에서는 통신주기를 0.1초와 0.05초로 달리하면서 실험을 수행하였고, 고장복구 시간은 노드 3과 노드 28에서 각각 측정하였다.

3.3.2 고장복구 지연 실험 결과

그림9는 통신주기가 0.1초일 때, 노드 28의 고장복구 시간을 측정한 결과이다. 그림9에서 ㉑는 노드 2가 고장에서 복구된 시점이고 ㉒는 노드 28의 통신이 정상적으로 이루어진 시점이다. 따라서 노드 28의 고장복구 시간은 10.5초이다. 제안한 모델을 이용한 계산 결과 일치하는 것을 확인할 수 있다. 그림10은 통신주기가 0.1초일 때 노드 3의 고장복구 시간을 측정한 결과이다. 실험을 통해 측정한 값은 8초로서, 제안한 모델을 이용한 계산한 결과, 제안한 모델을 이용한 추정시간과 일치한다. 그림11은 통신주기가 0.05초일 때 노드 28의 고장복구 시간을 측정한 결과이다. 실험에서 측정한 값은 7.85초이고, 제안한 모델을 이용하여 계산한 결과는 7.75초로 오차는 0.1초이다. 이러한 오차는 반복 실험으로 확인한 t_{td} 및 t_w 의 표준편차 범위에 들어오는 것으로 제안한 모델이 효과적임을 확인할 수 있다. 아울러, 실험 결과를 표1에 정리하였다.

표 1. EtherCAT 고장복구 시간 실험 결과
Table 1. Experimental result of fault recovery time

Communication Cycle (s)	Node Number	Fault Recover time(s) - Estimated	Fault Recover time(s) - Experiment
0.1	28	10.5	10.5
0.1	3	8.0	8.0
0.05	28	7.75	7.85

IV. 결 론

IT기술이 발전함에 따라 지능형 센서 및 액추에이터의 사용이 증가하면서 차량내 통신 네트워크를 통한 데이터 전송량은 꾸준한 증가 추세를 보이고 있다. 이에 대응하여 FlexRay, MOST 등 새로운 프로토콜이 적용되고 있으며, 한편으로는 1Gbit/s의 전송 속도를 지원하는 Ethernet을 이용한 차세대 차량내 통신 네트워크를 구축하려는 연구도 진행 중이다. 본 논문에서는 차세대 차량내 통신 네트워크로 고려되고 있는 EtherCAT의 전송지연 및 고장복구 지연 모델을 제시하였다. 고장복구 지연 모델은 EtherCAT 마스터의 동작 메커니즘을 분석한 결과를 토대로 도출하였으며, 통신주기, 물리계층의 성능, 고장복구 지연시간을 측정할 노드의 위치 등을

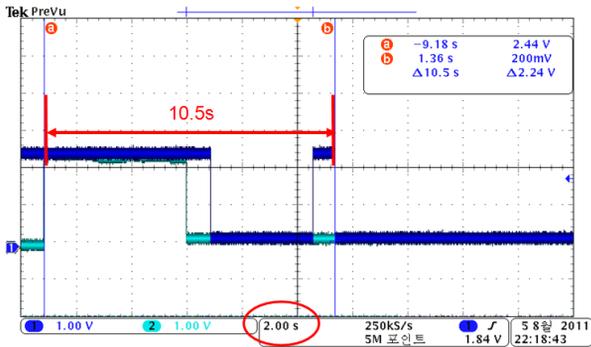


그림 9. 노드 28의 고장복구 시간(통신주기 0.1초)
Fig. 9. Fault recovery time on node 28 with 0.1 cycle time

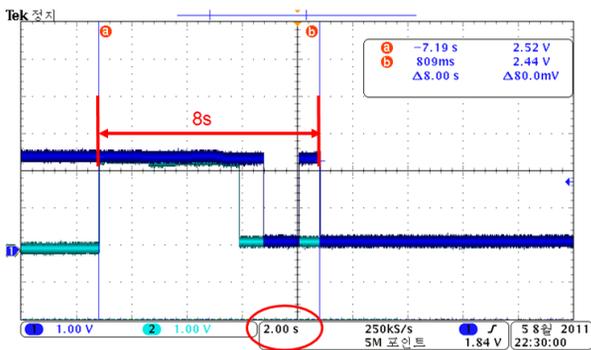


그림 10. 노드 3의 고장복구 시간(통신주기 0.1초)
Fig. 10. Fault recovery time on node 3 with 0.1 cycle time

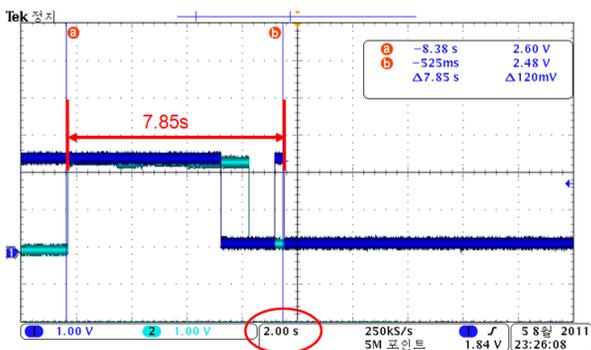


그림 11. 노드 28의 고장복구 시간(통신주기 0.05초)
Fig. 11. Fault recovery time on node 28 with 0.05 cycle time

고려하였다. 제안한 모델의 효용성을 검증하기 위해 실험적 네트워크를 구성하여 실험을 수행하고 그 결과를 제시하였다.

본 논문에서 간단한 네트워크를 구성하여 제안한 모델을 실험으로 검증하였으나, EtherCAT이 시간동기를 기반으로 정해진 스케줄링에 따라 메시지를 전송하는 특성을 감안하면 더욱 복잡한 구성의 네

트워크 환경에서도 제안한 모델이 효과적일 것이라고 예상할 수 있다. 본 논문에서 제안한 모델을 이용하여 EtherCAT 네트워크의 전송지연 및 고장복구 시간을 보다 정밀하게 예측할 수 있으며, 이를 토대로 보다 효과적인 차량제어용 네트워크 설계가 가능할 것으로 기대된다.

References

- [1] R. Bruckmeier, "Ethernet for Automotive Applications", *Freescale Technology Forum*, Orlando, June 2010.
- [2] M. Jochim, "Ethernet in the Car", *SAE/TTA-Group Open Forum on Time-Triggered Technologies*, Texas, Nov. 2010.
- [3] G. Reichart, "AUTOSAR and beyond", *ISEF*, Michigan, March 2010.
- [4] H. Kopetz, *Real time systems - Design Principle for Distributed Embedded Application*, Kluwer Academic Publishers, 1997.
- [5] EtherCAT Technology Group, Available: <http://www.ethercat.org>.
- [6] M. Sung and K. Kim, "Implementation and Analysis of a Networked Motor Drive using Real-Time Ethernet," *Journal of KIISE: Computing Practices and Letters*, vol. 17, no. 8, pp. 457-462, Aug. 2011.
- [7] G. Prytz, "A performance analysis of EtherCAT and PROFINET IRT", *Proc. of Emerging Technologies and Factory Automation*, pp. 408-415, Sep. 2008.
- [8] Y. Tipsuwan and M. Chow, "Control methodologies in networked control systems", *Control Engineering Practice*, vol. 11, no. 10, pp. 1099-1111, Oct. 2003.
- [9] K. Ha, M. Kim, K. Lee, and S. Lee, "Performance Evaluation of Network Protocol for Automated Transfer Crane System," *Journal of Control, Automation, and Systems Engineering*, vol. 11, no. 8, pp. 709-716, Aug. 2005.
- [10] T. Pop, P. Pop, P. Eles, Z. Peng, and A. Andrei, "Timing Analysis of the FlexRay

Communication Protocol”, *Proc. of 18th IEEE Euromicro Conference on Real-Time Systems*(ECRTS'06), pp. 203-216, July 2006.

- [11] A. Leon-Garcia and I. Widjaja, *Communication Networks 2nd Ed.*, MacGraw Hill, 2003.
- [12] EtherCAT Technology Group, “EtherCAT Slave Implementation Guide v1.1.6”, 2009.
- [13] ETG.1000, EtherCAT Specification, 2010. Available: <http://www.ethercat.org>.
- [14] EtherCAT Technology Group, “Topology variants and calculation of frame delay time”, Available : <http://www.ethercat.org>.
- [15] Beckhoff, “ET1100: EtherCAT Slave Controller”, 2010.

이 동 익 (Dongik Lee)



1987년 8월 경북대학교 전자공학과 공학사
 1990년 2월 경북대학교 전자공학과 공학석사
 1990년 2월~1997년 8월 국방과학연구소 연구원
 1997년 9월~2002년 4월 영국

세필드대학교 자동제어시스템공학과 공학박사
 2002년 1월~2005년 3월 영국 DRTS Ltd 공동설립
 2005년~현재 경북대학교 전자공학부 부교수.
 <관심분야> 네트워크 기반 제어, 시스템 안전, 지능형 자동차, 수중로봇

김 동 길 (Dong-gil Kim)



2006년 2월 경북대학교 전자컴퓨터공학부 졸업
 2008년 2월 경북대학교 전자공학과 석사
 2008년 3월~현재 대학교 전자공학과 박사과정
 <관심분야> 네트워크 기반 제어, 필드버스, 고장대처 제어

조 영 현 (Younghuyn Jo)



2010년 8월 경북대학교 전자컴퓨터공학부 졸업
 2012년 2월 경북대학교 전자공학과 석사
 2012년 3월~현재 LG 전자 CTO부분
 <관심분야> 임베디드 시스템