EtherCAT Cycle Time의 성능 분석 및 향상에 관한 연구

이범용*, 박재현*
*인하대학교 정보통신공학부
*e-mail : lbumyong@emcl.inha.ac.kr

A Study on Analyzing and Improving EtherCAT Cycle Time

Bum-Yong Lee*, Jaehyun Park*
*Dept of Information & Communication Engineering, Inha University

요 약

기존 필드버스의 낮은 트래픽 전송률로 인하여 IEEE 802.3 이더넷이 필드버스의 대안으로 제시되었다. 하지만, 이더넷은 CSMA/CD 메카니즘을 사용하기 때문에 실시간 성이 떨어진다는 단점이 있다. 이러한 단점을 개선하기 위하여 EtherCAT 이 개발되었다. 본 논문에서는 EtherCAT의 Cycle Time 성능을 분석하고 개선하기 위하여 M-EtherCAT 을 제안한다. M-EtherCAT 은 EtherCAT 과 달리 각 Telegram이 Slave를 거치지 않기 때문에 EtherCAT 보다 Cycle Time의 최소 주기를 더 줄일 수 있다.

1. 서론

필드버스 시스템은 산업용 시스템의 높은 실시간성의 요구를 충족시키기 위하여 많은 발전이 있어왔다. 하지만, 시간이 지남에 따라 필드 버스 시스템에서도 실시간성만이 아닌 높은 트래픽 전송률이 요구되었지만, 기존의 필드버스 시스템으로는 높은 전송률을 만족시킬 수 없어 이더넷이 필드버스 시스템의 대안으로 제시되었다.

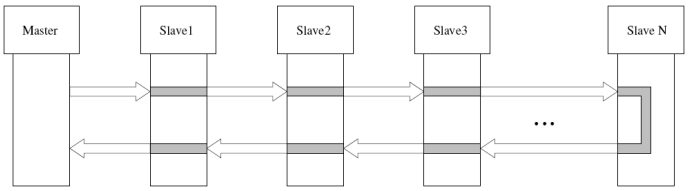
이더넷은 1980년도에 Xerox사에 의하여 개발되어 현재 IEEE 802.3 표준으로 제정되어 꾸준한 발전으로 10Mbps 뿐만 아니라, 100Mbps에서 1Gbps까지의 트래픽 전송률까지 지원되고 있으며, 10Gbps에서 100Gbps까지 지원예정인 고속의 네트워크 프로토콜이다.

하지만, 이더넷은 CSMA/CD 메카니즘에 의한 비결정적인 지연시간에 의하여 실시간성이 떨어진다는 단점이 있다. 이를 해결하기 위하여 여러 산업용 이더넷 프로토콜이 고안되었고, Beckhoff에서 EtherCAT이라는 프로토콜이 고안되었다.

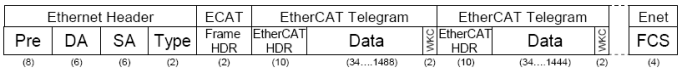
EtherCAT은 Master/Slave 구조로서 Master와 Slave의 Cycle Time에 의하여 성능이 결정된다. 본 논문에서는 EtherCAT의 성능 분석과 성능 향상에 대하여 논한다.

2. EtherCAT

EtherCAT은 그림 1과 같이 Master/Slave의 Daisy-Chain된 구조이다. 데이터 전송 시, Master는 Slave로 프레임을 전송한다. 이 때, Slave는 Master에서 전송한 Command Telegram에 해당하는 데이터를 그림2와 같이 이더넷 프레임의 데이터 영역에 On-the-fly 방식으로 Telegram을 추가하여 이더넷 프레임을 전송한다. 그리고, Daisy-Chain의 마지막에 있는 Slave는 수신된 데이터를 루프백하여 Master에게 데이터를 재전송한다.



(그림 1) EtherCAT 토폴로지



(그림 2) EtherCAT Frame Format

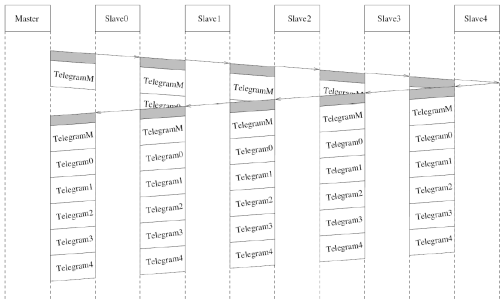
Master에서 Read Command를 브로드 캐스트하고, 각 노드에서는 같은 사이즈의 텔레그램을 추가한다면, Master에서 전송한 프레임이 다시 되돌아오는 시간의 타이밍은 그림3과 같고 되돌아오는 총 시간 $T_{Roundtrip}$은 식 1과 같게 된다.

$$T_{roundtrip} = 2N^* T_{prop} + N^* T_{forward} + (N-1)^* T_{bypass} + T_{header} + (N+1)^* T_{data}$$

식 (1)

여기서, T_{prop}은 전송매체의 지연시간, $T_{forward}$는 노드 내부에서의 지연시간, T_{bypass}는 마지막 Slave로부터 되돌아오는 프레임을 바이패스 시킬 때의 지연시간, T_{header}는

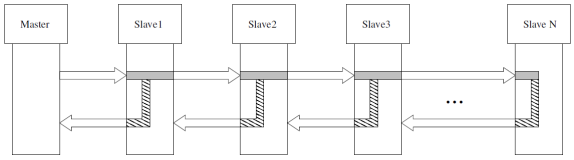
이더넷 헤더의 전송시간, T_{data}는 데이터 전송시간을 뜻한다.



(그림 3) EtherCAT의 데이터 전송 타이밍

3. M-EtherCAT

EtherCAT의 모든 텔레그램은 Master와 Slave 사이에만 데이터 교환이 이루어고, Slave 사이에 필요한 데이터 교환에도 Master가 Gateway의 역할을 하여 데이터 교환이 이루어진다. 하지만, 현재 EtherCAT 다른 Slave에서 사용되어 지지 않는 텔레그램을 다음 Slave 노드에게 전송한 후, 루프백하여 불필요한 지연시간이 생기게 된다.



(그림 4) 수정된 EtherCAT의 전송방식

이를 해결하기 위하여, 그림4와 같이 M(Modified)-EtherCAT의 각 Slave노드에서는 Master로부터 프레임을 수신 시, 다음 Slave 노드로의 이더넷 프레임 전송 시 자신이 전송할 EtherCAT 텔레그램을 추가시키지 않고, Master 방향으로 EtherCAT 텔레그램이 저장된 이더넷 프레임을 전송한다.

이렇게 하게 되면, Master에서 마지막 Slave노드까지 이더넷 프레임을 전송하는데 까지 걸리는 시간이 단축되게 된다.

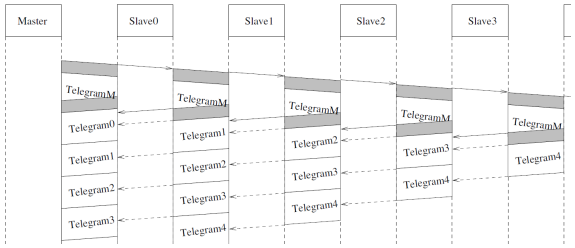
Master에서 Read Command를 브로드 캐스트하고, 각 노드에서는 같은 사이즈의 텔레그램을 추가한다면, Master에서 전송한 프레임이 다시 되돌아오는 시간의 타이밍은 그림5과 같고 되돌아오는 총 시간 $T_{roundtrip}'$은 식2와 같게 된다.

$$T_{roundtrip}' = 2(T_{prop} + T_{forward}) + T_{header} + (N+1) * T_{data}$$

식 (2)

4. 성능 비교

EtherCAT과 수정된 EtherCAT의 Cycle Time의 차이 T_{gap} 은 식3과 같다.



(그림 5) M-EtherCAT 데이터 전송 타이밍

$$T_{gap} = T_{roundtrip} - T_{roundtrip}' = 2(N-1)T_{prop} + (N-2)T_{forward} + (N-1) * T_{bypass}$$

식 (3)

식 3에서 알 수 있듯이 수정된 EtherCAT은 노드의 수가 증가할수록, 기존의 EtherCAT 보다 더 짧은 Cycle Time을 갖게 된다.

5. 결론

본 논문은 산업용 이더넷에 사용되는 EtherCAT의 성능을 분석하고, 성능을 개선시키기 위한 방법에 대하여 논하였다. 기존의 EtherCAT의 경우 전송 매체의 전송시간과 노드 내부에서 지연시간이 노드의 개수에 비례하여 Cycle Time이 증가하였지만, M-EtherCAT의 경우 노드 내부에서의 지연시간과 전송매체의 지연시간이 일정하였다. 그러므로 M-EtherCAT을 사용한다면 노드 수가 증가함에 따른 Cycle Time의 증가율을 줄여 EtherCAT의 전송률이 증가할 수 있을 것이다.

참고문헌

[1] Beckhoff, "EtherCAT Communication Specification", Beckhoff, 2004.

[2] Lucia Seno 외 1인, “A Simulation Approach to a Real-Time Ethernet Protocol : EtherCAT”, Padova Univ, 2008.

[3] Gianluca Cena 외 2인, “An Arbitration-based Access Scheme for EtherCAT Networks”, Torino, 2008.

[4] Markus Schumacher 외 2인, “A new Approach for Increasing the Performance of the Industrial Ethernet System PROFINET”, OWL University of Applied Sciences, 2008.

[5] Sven Gestegard Robertz 외 2인, “Industrial robot motion control with real-time Java and EtherCAT”, Lund University, 2007.

[6] MAX FELSER, "Real-Time Ethernet - Industry Prospective", IEEE Member, 2005.

[7] Gunnar Prytz, "A performance analysis of EtherCAT and PROFINET IRT", ABB AS Corperate Research Center, 2008.

[8] Juergen Jasperneite 외 2인, “Limits of Increasing the Performance of Industrial Ethernet Protocols”, LuH Univ, 2007.