

통합 필드버스 게이트웨이 설계

박진원⁰ 이준형 이희찬 김명균
울산대학교 컴퓨터·정보통신공학과

{zwsnic⁰,comfuny, dino}@cnlab.ulsan.ac.kr, mkkim@ulsan.ac.kr

Design of Integrated fieldbus gateway

Zin-Won Park⁰ Joon-Hyung Lee Hee-Chan Lee Myung-Kyung Kim
Dept. of Computer Architecture and Information Technology, University of Ulsan

요 약

오늘날 공장 자동화에 이용되는 필드버스 네트워크는 여러 가지가 있으며 이들은 각각의 미디어 접근 방식과 프로토콜을 사용하고 있다. 서로 다른 종류의 네트워크는 완전히 분리되어 있기 때문에 네트워크 관리자는 이를 관리하는데 많은 어려움을 겪고 있다. 본 논문에서는 이러한 이기종의 필드버스 네트워크를 통합할 수 있는 통합 필드버스 게이트웨이를 설계하였다. 통합 필드버스 게이트웨이는 이더넷 프로토콜을 이용하여 다른 네트워크로 데이터를 전달할 수 있고 프로토콜간의 변환을 수행하여 분리된 네트워크를 통합하고 통합된 환경으로 제어 및 모니터링을 가능하게 한다.

1. 서 론

오늘날 공장과 같은 생산 현장에서는 공장 자동화를 위한 컴퓨팅 디바이스들이 사용되고 있으며 이러한 디바이스들은 모두 네트워크로 연결되어 동작한다. 공장 자동화에 사용되는 네트워크는 여러 종류가 있지만 그 중 많이 사용되는 것으로는 CAN, Profibus, DeviceNet, Lonworks 등이 있으며 이들은 서로 다른 미디어 접근방식과 프로토콜을 사용하고 있어 각각의 특성 및 장단점을 가지고 있다. 각 네트워크의 관리 시스템들은 네트워크에 독립적으로 존재하며 디바이스들에게 제어 명령을 내리거나 데이터를 수집하여 저장하고 분석한 다음, 다음 제어 명령을 내리는데 활용된다. 뿐만 아니라 통계 정보를 수집하여 라인의 작업 상태를 모니터링 하는 데에도 사용되고 있다. 하지만 하나의 생산라인에 서로 다른 네트워크가 구성되어 있는 경우도 있는데 이런 경우 각 네트워크를 관리하는 관리 시스템이 별도로 구성되어 있어 서로 다른 네트워크에서 수집한 정보를 통합하기 힘들뿐만 아니라 통합을 위한 새로운 네트워크를 구성해야하는 어려움이 있다. 따라서 본 논문에서는 이러한 서로 다른 네트워크로 구성된 필드버스를 이더넷을 이용하여 통합하는 통합 필드버스 게이트웨이를 설계하였다.

본 논문의 구성으로는 2장에서는 각 네트워크가 어떠한 특성을 가지고 있는지 필드버스의 특성을 알아본다. 3장에서는 통합 필드버스 게이트웨이의 프로토콜 변환 기능의 설계 방법에 대해 설명한 다음 4장에서는 결론을 맺고 향후 연구 방향에 대해서 논한다.

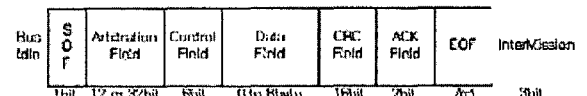
2. 필드버스의 종류 및 특성

본 장에서는 여러 가지 필드버스의 종류 중에서 본 연구에서 고려한 CAN과 Profibus에 대해서만 다룬다. 또한 게이트웨이 설계에서 고려한 데이터 링크 계층에 대해서만 다룬다.

2.1 CAN 2.0x

CAN에는 몇 가지 버전이 있는데 본 절에서는 CAN 2.0A와 CAN 2.0B 버전에 대해서만 다룬다. CAN은 버스 구조의 토폴로지로 구성된다. CAN에는 노드의 주소를 표시하는 필드가 따로 없어 메시지 ID에 대한 필터링을 통하여 원하는 노드만이 수신하도록 하고 있다. CAN의 아비트레이션 필드(Arbitration Field)는 낮은 값을 갖는 메시지가 높은 우선순위를 가지는 우선 순위 필드이기도 하며 네트워크에서 유일한 메시지임을 구분하는 메시지ID 필드이기도 하다. CAN에서는 노드가 주기적으로 메시지를 발생시켜 전체 네트워크로 브로드캐스팅하고 메시지ID를 보고 이를 필요로 하는 노드만 읽어들이는 통신 방식과 특정 노드가 다른 노드들로 메시지를 요청하고 요청을 받은 노드들이 이에 응답하는 통신 방식을 사용한다. CAN은 하나의 프레임에 전송할 수 있는 데이터는 0~8바이트로 제한되어 있다. CAN 2.0A 표준에서 아비트레이션 필드는 11비트의 길이를 가지고 있어 최대 2¹¹가지를 표현할 수 있으나 CAN 2.0B 표준에서는 아비트레이션 필드가 27비트로 확장되어 있다. CAN의 메시지 전송 방식은 전송하고자 하는 메시지의 우선순위가 낮을 경우 경쟁에서 자주 뒤지게 되므로 오랫동안 전송할 수 없게되는 문제점을 가지고 있으나 이를 해결하기 위한 스케줄링 방법 등이 많이 연구되고 있다[1, 2].

[그림 1]은 CAN 2.0x의 프레임 구조를 보여주고 있다.



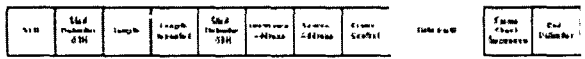
[그림 1. CAN 프레임 구조]

2.2 Profibus

Profibus는 버스 및 트리 구조의 토폴로지로 구성될 수 있으며 가상 토큰을 이용한 토큰 패싱(Token-passing) 방식을 사용하여 통신을 한다. 네트워크는 하나 이상의 마스터와 슬레이브들로 구성되어 있으며 토큰은 마스터들 사이에서만 이동한

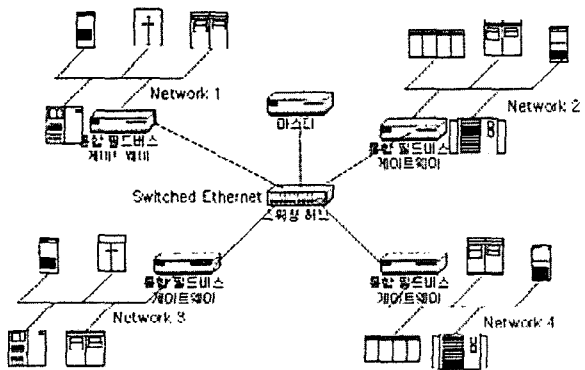
다. 그리고 통신은 토큰을 가지고 있는 마스터만이 할 수 있다. 실시간성을 보장하기 위하여 마스터가 토큰을 보유할 수 있는 토큰 보유 시간(Token holding time)이 정해져 있어 마스터는 이 시간 내에 통신을 마쳐야 한다. 슬레이브들은 자율적으로 데이터를 전송하지 않고 토큰을 가진 마스터가 데이터를 요청하면 이에 대해 응답하는 수동적인 통신만을 한다. 네트워크에 마스터가 2개 이상이 존재할 경우 마스터는 다음 토큰을 가질 마스터와 자신에게 토큰을 넘겨 줄 마스터의 주소를 유지하고 있다. 따라서 이전 마스터로부터 토큰을 받고 자신이 사용할 수 있는 토큰 보유 시간 이내에 통신이 끝날 경우 다음 마스터에게 즉시 토큰을 넘겨줄 수도 있다. Profibus의 프레임은 시작 구분(Start delimiter) 필드를 가지고 있는데 이 필드를 통하여 프레임의 종류를 결정한다. Profibus 표준에 정의된 프레임 종류로는 no data 프레임, 고정 데이터 프레임, 가변 데이터 프레임, 토큰 프레임이 있으며 최대 246바이트의 데이터를 전송할 수 있다. 주소 필드는 목적지 주소와 출발지 주소가 있는데 각각 7비트를 사용할 수 있기 때문에 최대 127가지의 주소 표현이 가능하다. 데이터의 전송은 토큰을 가졌을 때에만 가능하기 때문에 미디어에서 충돌이 발생하지 않으며 메시지에 대한 우선 순위는 응용프로그램이 자율적으로 지정한다. Profibus에 관한 많은 연구들은 토큰 보유 시간을 지정하는 스케줄링 방법과 실시간 데이터와 비실시간 데이터를 스케줄링 방법에 대해 이루어지고 있다[3].

아래의 [그림 2]는 Profibus의 가변 데이터 프레임 구조를 보여주고 있다. 각 필드는 데이터 필드를 제외하고는 모두 1바이트의 크기를 가진다.



[그림 2. Profibus 프레임 구조 : 가변 데이터 프레임]

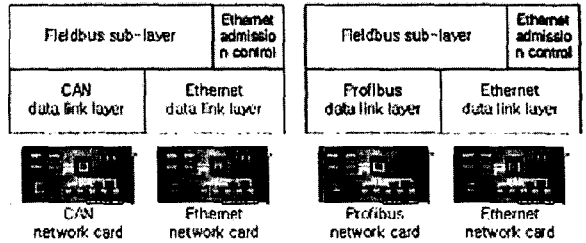
3. 통합 필드버스 게이트웨이 설계



[그림 3. 통합 필드버스 게이트웨이를 이용한 구성]

[그림 3]은 통합 필드버스 게이트웨이를 이용하여 네트워크를 스위치 이더넷으로 연결한 구성을 보여주고 있다. 본 논문에서 제안하는 통합 필드버스 게이트웨이는 CAN-Ethernet, Profibus-Ethernet 인터페이스를 가진 두 가지의 형태이다.

[그림 4]는 통합 필드버스 게이트웨이의 모듈 구조를 보여주고 있다. 본 절에서는 이더넷을 이용한 데이터 전달 방법과 프로토콜 변환과정을 위한 설계 방법에 대해 설명한다.



(a) CAN 게이트웨이 (b) Profibus 게이트웨이

[그림 4. 통합 필드버스 게이트웨이 모듈구조]

3.1 프로토콜 변환

각 필드버스에서 수신한 데이터는 이더넷 프레임으로 변환이 되어 다른 게이트웨이로 전달되고 수신된 이더넷 프레임은 다시 목적지 필드버스 프레임으로 변환이 된다. 이 때 CAN의 경우 메시지ID와 이더넷 주소, Profibus의 경우 노드 주소와 이더넷 주소의 매핑 테이블이 사용되는데 그 예는 [표1]에서 볼 수 있다.

[표 1. 주소 매핑 테이블의 예]

(a)CAN 게이트웨이 (b)Profibus 게이트웨이

| CAN 게이트웨이 | | Profibus 게이트웨이 | |
|------------|----------------|----------------|----------------|
| Message ID | Ether. Addr. | Node Addr. | Ether. Addr. |
| 0x789 | 0x000001010789 | 0x7A | 0x00000202007A |
| 0x58E | 0x00000101058E | 0x7B | 0x00000202007B |
| 0x6A0 | 0x0000010106A0 | 0x7C | 0x00000202007C |
| 0x4EF | 0x0000010104EF | 0x7D | 0x00000202007D |

각 게이트웨이의 이더넷 인터페이스는 연결된 필드버스에 필요한 수만큼의 이더넷 주소를 가지고 있다. CAN 게이트웨이의 경우 CAN 네트워크에서 사용되는 메시지ID 수만큼, Profibus 게이트웨이의 경우 Profibus 네트워크의 노드 수만큼의 이더넷 주소를 가지고 있다. 따라서 수신한 데이터의 목적지 이더넷 주소와 대응하는 메시지ID나 Profibus 목적지 주소를 이용하여 프레임 변환을 한다. CAN에서 수신한 메시지의 ID가 [표1(a)]에서와 같이 0x0789일 경우 이 메시지의 데이터는 이더넷 페이로드(payload) 필드에 들어가고 이더넷 송신자 주소는 0x000001010789로 된다. 목적지 이더넷 주소는 3.2절에서 설명할 라우팅 테이블에 의해 결정된다. 예를 들어 수신측 이더넷 주소가 [표1(b)]에서 예를 든 0x00000202007A라고 할 경우 수신측 게이트웨이는 매핑 테이블에 의해 목적지가 0x7A인 Profibus 프레임을 구성하여 Profibus 네트워크에 데이터를 전송하게 된다. 이러한 동작은 [그림 4]의 Fieldbus sub-layer에서 이루어진다.

3.2 라우팅 테이블 구성

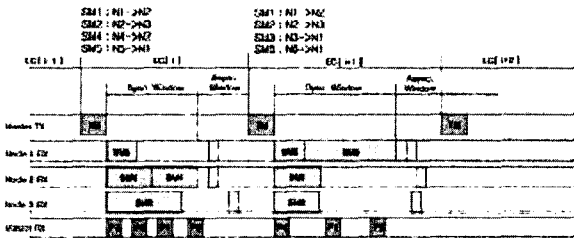
필드버스에서 발생한 메시지를 어느 게이트웨이로 전송해야 할지 결정하는 라우팅 테이블은 아래와 같은 과정을 거쳐 얻을 수 있다. CAN 게이트웨이가 셋업될 때 CAN에서 사용하는 모

은 메시지ID를 이용해 요청 메시지를 이더넷에 브로드캐스팅을 하고 이에 대한 응답이 수신될 경우 해당 메시지를 전달할 노드가 있다는 것을 알아차린 후 응답한 목적지 이더넷 주소와 게이트웨이의 송신 주소의 쌍을 라우팅 테이블에 추가한다. 이러한 과정은 셋업되는 시점과 CAN 네트워크에 새로운 메시지 ID가 추가될 경우에만 수행된다. 반면 Profibus 게이트웨어의 경우 이더넷 주소와 Profibus 목적지 주소 매핑 테이블은 매뉴얼로 구성한다.

3.3 스위치 이더넷 스케줄링

polling message = { #msg, DSF, DLC, PERI, PRIO, DLJ }, #1..ADCA
 DSF : 데이터 수, DLC : data length in byte, PERI : period, PRIO : priority
 DLJ : deadline, ADCA : number of address
 TM = { (#msg, START, Di C, Cj, DUTM) }, #1..ADCA
 RTART : 시작 시간
 C : maximum transmission time
 DUTM : duration of this TM bit-time

(a) Message format



(b) Bus 상의

[그림 5. 스위치 이더넷 스케줄링]

CAN과 Profibus에서 전달되는 데이터 중에는 실시간 전송을 요구하는 데이터가 존재할 수 있다. 따라서 네트워크간 데이터를 전달하는 이더넷 역시 실시간 전달을 보장해야만 한다[4]. [그림 4]의 Ethernet admission control 모듈은 스위치 이더넷을 이용하여 데이터를 전달하고 스위치 이더넷에서 실시간 전달을 보장하지 못할 수 있는 경우 스케줄링을 통하여 실시간 전달을 보장할 수 있도록 한다. 스위치 이더넷에서는 스위치와 노드 사이에는 충돌(collision)이 발생하지 않는다. 다만 서로 다른 노드에서 출발한 데이터가 같은 목적지로 동시에 전송될 경우 버퍼를 이용하여 전송을 보장하지만 버퍼의 한계를 초과하는 데이터가 집중될 경우 데이터가 유실될 수 있다. 따라서 스위치 이더넷에 마스터/슬레이브 구조의 스케줄링을 이용한다. 마스터는 게이트웨이를 사이에서 선출하여 정해지며 슬레이브들이 요청하는 폴링 메시지(polling message)를 수신하여 목적지가 같은 요청들간에 순서를 정하고 TM(Trigger Message)을 이용하여 이를 알려주는 역할을 한다. 슬레이브들은 TM에 따라 데이터를 전송하고 새로이 전송할 데이터가 발생할 경우 다시 요청메시지를 전달한다. 이더넷은 송수신 채널이 따로 있기 때문에 수신을 하면서도 요청 메시지를 송신할 수 있다. TM은 실시간 전송을 보장하기 위해 정해진 최대값을 넘지 않는 범위내에서 가변적인 간격으로 전송된다. 또한 대역폭을 초과하는 요청이 수신될 경우 우선 순위에 따라 스케줄링에서 제외시킬 수도 있다. [그림 5(a)]는 요청 메시지와 TM의 구성을 나타내며 [그림 5(b)]는 스위치 이더넷에서의 버스 상

태의 한 예를 보여주고 있다.

3.4 통합 필드버스 게이트웨이 동작 과정

앞에서 살펴본 통합필드버스의 동작 과정은 예를 들어 설명하겠다. [그림 3]에서 Network1은 CAN이고 Network4는 Profibus라고 가정한다. CAN의 노드A는 Profibus의 노드B에게 메시지를 전송 하고자 한다. 노드A가 전송한 데이터는 CAN측 게이트웨이가 수신하고 자신의 주소매핑 테이블을 검색하여 목적지 게이트웨이의 이더넷 주소를 찾아 이더넷 프레임으로 변환한다. 목적지 이더넷 주소를 알아내면 게이트웨이는 이더넷 마스터에게 전송할 데이터가 있다는 것을 폴링 메시지를 통하여 알린다. 마스터는 노드들로부터 수신한 폴링 메시지를 바탕으로 스케줄링을 하고 스케줄링 결과로 TM를 구성하여 브로드캐스팅한다. CAN측 게이트웨이는 자신이 스케줄링된 시점을 TM을 읽고 알 수 있으며 해당 시점에 데이터를 전송한다. 데이터를 수신한 Profibus측 게이트웨이는 자신이 가지고 있는 주소매핑 테이블을 검색하여 해당 데이터를 수신할 목적지 노드B의 주소를 찾고 Profibus 프레임으로 구성하여 Profibus 네트워크에 전송한다.

4. 결론 및 향후 계획

본 논문에서는 서로 다른 필드버스 네트워크를 연결하는 통합 필드버스 게이트웨이에 대해서 알아보았다. 통합 필드버스 게이트웨이는 네트워크간의 데이터 전달을 위하여 프로토콜 변환을 수행하며 본 논문에서는 CAN과 Profibus간의 변환 과정에 대해서 알아보았다. 본 논문에서 제시하는 통합 필드버스 게이트웨이는 프레임을 이더넷 프레임으로 변환하여 다른 게이트웨이로 전달하도록 하였고 실시간 전송을 위하여 스케줄링을 통한 전송을 한다. 따라서 종중의 혹은 이기종의 원격지 네트워크로 데이터를 실시간으로 전달하는 것이 가능할 뿐만 아니라 원격 제어 및 모니터링도 가능하다.

향후 프레임 변환시 [이더넷 주소]-[Profibus주소] 매핑 테이블을 자동으로 생성할 수 있는 방법을 연구할 것이며 임베디드 리눅스로 구현하여 다양한 테스트를 수행할 것이다.

5. 참고문헌

[1]Cena, G.; Valenzano, A., "An improved CAN fieldbus for industrial applications", IEEE Transactions on , Volume: 44, 1997
 [2] Almeida, L.; Pedreiras, P.; Fonseca, J.A.G., "The FTT-CAN protocol: why and how", IEEE Transactions on , Volume: 49, 2002
 [3] Tovar, E.; Vasques, F., "Real-time fieldbus communications using Profibus networks", IEEE Transactions on , Volume: 46, 1999
 [4] Pedreiras, P.; Almeida, L.; Gai, P., "The FTT-ethernet protocol: merging flexibility, timeliness and efficiency", Real-Time Systems, 2002. Proceedings, 2002