1999 年 度 大 韓 電 子 工 學 會 夏 季 綜 合 學 術 大 會 論 文 集 第 22 卷 第 1 號 99/6

PC/Ethernet 기반의 제어용 통신 프로토콜 설계 · 구현

°장태인*, 곽귀일*, 변승현*, 조지용*
*한국전력공사 전력연구원 전력계통연구실 정보통신그룹
대전광역시 유성구 문지동 103-16
E-mail:angeljti@kepri.re.kr

Design and Implementation of a PC/Ethernet-based Communication Protocol for Control Systems

T.I. Jang, K.Y. Gwak, S.H. Byun, J.Y. Cho
Computer & Communication Group, Power System Lab., KEPRI, KEPCO
103-16 MunJi-Dong, YuSeong-Ku, Taejeon, 305-380
Tel:+82-42-865-5947;Fax:+82-42-865-5408;E-mail:angeljti@kepri.re.kr

Abstracts

This paper deals with the design and implementation of an Ethernet-based communication protocol to be used in PC-based I/O interface systems. Recently, the performance of PC systems is being highly improved and Ethernet is used as the stable communication network over the world.

We develop a new protocol driver with the capability of accessing Ethernet directly using NDIS(Network Driver Interface Specification), the network interface standard of Windows O.S. in PC, and install it at the application layer of the protocol structure. Its major roles are the supplement of CSMA/CD algorithm, the effective use of the long data frame of Ethernet, and the real-time transmission of data frames. This paper represents the possibility of the real-time control network and systems based on PCs and Ethernet.

1. 서론

최근 PC의 성능이 비약적으로 발전됨에 따라 기존의 PLC(Programmable Logic Controller) 시스템을 PC를 이용하여 구현한 Soft-PLC가 출현하고, 각종 모니터링 소프트웨어, SCADA(Supervisory Control And Data Acquisition) 소프트웨어 등 기존의 유닉스 운영체제 기반의 시스템에서 운용되었던 제어자동화용 소프트웨어들이 윈도NT 및 95 운영체제를 탑재한 PC 시스템에 이식되어 시장에 나오고 있다.

이더넷은 전세계적으로 널리 설치되어 다양한 목적으로 사용되고 있는 네트워크로서 지난 20여년 동안 그 신뢰성과 안정성이 입증되어 왔으며 FDDI나 ATM 망과 같은 고속망의 출현과 발전에도 불구하고 앞으로도 LAN 환경의 통신망으로서 가장 널리 사용될 것으로 전망된다. 그리고, 전송 속도 100Mbps의 Fast Ethernet의 출현으로 이더넷 환경에서 고속의 통신망을 구축하고 사용할 수 있게 되었다.

본 논문은 실시간 운영체제는 아니지만 가격 대 성능비가 뛰어난 윈도 운영체제를 탑재한 PC와 이더넷을 사용하여 제어용 통신 프로토콜 개발하고자 할 때 고려해야 할 몇 가지 사항을 다룬다.

프로토콜 설계시 가격이 저렴하고 신뢰성이 입증된 이더넷 LAN을 통신망 하부 1, 2 계층에 그대로 사용하고, 그 위 제 3 계층에 이더넷의 CSMA/CD 알고리즘의 보완, 대량의 I/O 데이터 처리, 제어 데이터의 실시간 전송 기능을 갖는 상위 프로토콜을 개발하여 탑재한다.

이더넷의 매체 접근방식인 CSMA/CD는 데이터 전송시의 충돌 및 그 해결 과정의 특성상 매체 접근 시간이 예측 가능하지 못하여 실시간 네트워크에 적용할 수 있는 프로토콜로는 적합하지 못하다. 그러나, CSMA/CD 방식의 비실시간적인 요소가 제거된다면 CSMA를 기반으로 한 이더넷을 실시간 네트워크로서 널리 사용될 수 있기 때문에 상위 프로토콜층에 이 기능을 첨가하며, 그 방식은 통신망 제어 노드를 두어 네트워크에 접속되어 있는 시스템들의 네트워크 이용하는 시간을 제어하는 것이다. 또한, 대부분의 제어용 통신 프로토콜이 사용하는 프레임의 길이는 1~128 바이트인 데 반하여, 이더넷은 기본적으로 64~1518 바이트가 기본이므로 데이터 전송의 오버헤드를 줄이고 이더넷의 긴 프레임의 길이를 효과적으로 이용하도록 데이터를 블록화 하여 전송한다. 그리고, 네트워크를 통한 데이터 전송뿐만 아니라 응용 프로그램의 수행 시간도 엄격히 제어하여 전체 시스템의 리얼 타임성을 증진시킨다.

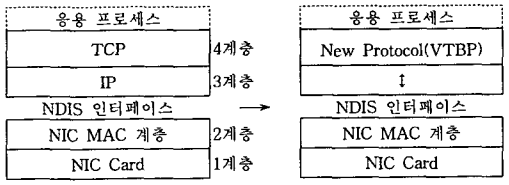
앞으로, 본론에서는 설계한 PC/Ethernet 기반의 프로토콜 구조를 간단히 알아본 다음 프로토콜의 주요 설계 내용, 그리고 시험 시스템 구성 및 성능 분석의 순서로 연구 내용을 기술한다. 결론에서 본 연구의 의의와 향후 전망에 대해서 알아본다.

2. 프로토콜 설계

2.1 프로토콜 구조

윈도 운영체제를 장착한 PC 시스템에서 기존의 상용 이더넷 카드와 드라이버를 프로토콜 하부 1, 2계층으로 그대로 사용할 경우 상위 프로토콜층과의 네트워크 인터페이스가 NDIS(Network Driver Interface Specification)로 표준화되어 있기 때문에 새로운 프로토콜 드라이버를 구현할 때 NDIS 규약을 따르도록 한다.

따라서, 설계한 PC/이더넷 기반의 프로토콜 계층 구조는 아래 그림 2.1 (b)와 같다. 또한, 그림 2.1 (a)는 현재 윈도 운영체제의 네트워크 기본 프로토콜로 가장 널리 쓰이는 TCP/IP를 채용한 프로토콜 구조를 보여주고 있다.



(a) TCP/IP 통신구조 (b) 적용한 통신 프로토콜 구조

그림 2.1 TCP/IP 통신망 계층구조(a)와 적용한 통신망 계층구조(b)의 비교도

그림 2.1 (b)에서 프로토콜 구조의 제 3계층은 기존의 TCP/IP, SPX/IPX 등의 프로토콜 계층을 제거하고 새로운 프로토콜[VTBP:Virtual Terminal Box Protocol]을 제작하여 설치한 것이다. 제어용 네트워크는 상호 연관성이 있는 I/O 데이터를 처리하는 여러 노드들이 하나의 LAN에 접속되어 있기 때문에 TCP/IP나 SPX/IPX 등의 여러 네트워크를 거치는 원거리 데이터 교환용 프로토콜을 제어용 네트워크에 적용하는 것은 비효율적이기 때문이다. 또한, 제어용 데이터를 실시간으로 처리하기 위해서는 프로토콜 구조를 되도록 단순화시킬 필요가 있기 때문에 전체 프로토콜 구조는 3계층으로 구성된다.

2.2 상위 프로토콜의 주요 설계 내용

2.2.1 매체 접근 권한 할당 방식

이더넷의 CSMA/CD 알고리즘은 데이터 전송시의 충돌 및 그 해결 과정의 특성상 매체 접근 시간이 예측 가능하지 못하다. 이를 보완하기 위한 가장 간단한 방법이 네트워크에 액세스하고자 하는 각 노드들에게 차례로 전송 권한을 할당하는 것이다. 특히, 제어용 네트워크에 동작이 정의되지 않는 새로운 노드들이 임의로 추가되는 것은 의미가 없으므로 미리 정해진 순서대로 전송권한을 할당하는 것이 유리하다. 전송권한을 할당하기 위해 여기서는 가장 간단한 통신망 제어노드에 의한 폴링(polling) 방식을 사용한다. 그러나, 단순히 폴링 신호만을 보내는 것이 아니라 자신이 생성하여 송신할 데이터도 함께 실어 보낸다. 이는 이더넷의 최소 프레임의 길이가 64바이트 이상이 되어야 하므로 폴링 신호만 보냈을 경우에는 데이터 전송의 오버헤드가 크기 때문이다.

2.2.2 데이터 샘플링 주기

각 노드가 생성하는 데이터에 대한 샘플링 주기 T_1 은 네트워크를 통한 데이터의 전달 시간뿐만 아니라, 각 노드가 처리해야하는 데이터 스캔, 처리/출력 등의 부하도 고려하여 식 (2.1)과 그림 2.2의 관계를 만족하도록 정한다. 이 때, 각 데이터의 전송 주기와 데드라인은 같다고 가정한다.

$$T_1 \geq T_{scan} + T_{network} + T_{out} \qquad (2.1)$$

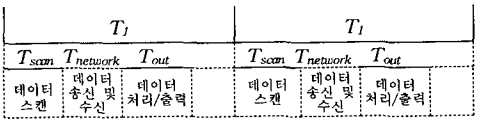


그림 2.2 데이터 샘플링 주기

식 (2.1)과 같이 샘플링 시간을 정할 경우는 T_1 시간 이내에 어떤 하나의 노드가 스캔한 데이터를 네트워크 상의 다른 노드에게 모두 전송할 수 있다. 통신망 제어 노드는 정기적으로 타이밍 프레임을 네트워크에 브로드캐스트 하여 각 노드의 프로그램 수행시간에 대한 동기를 유지한다.

2.2.3 데이터 블록의 송신 및 수신

2.2.3.1 데이터 블록

통신망 제 1, 2계층에 표준 LAN(여기서는 이더넷)을 채용하였으므로 표준 LAN 카드가 제공하는 긴 데이터 프레임(1500바이트)과 넓은 전송 대역폭(10Mbps)을 효과적으로 이용하도록 데이터를 블록화하여 처리한다. 이렇게 함으로써 대용량 I/O 포인트 데이터를 소수의 전송 횟수에 모두 교환할 수 있게 되어, 대용량 제어 시스템에서도 주어진 샘플링 시간 안에서 효과적인 실시간 통신 서비스를 제공할 수 있게 된다.

2.2.3.2 데이터 스캔, 송수신, 처리/출력 알고리즘

데이터의 스캔, 송신 및 수신, 처리/출력의 개시는 통신망 제어용 노드에서 발생시키는 브로드캐스트 타이밍 프레임이다. 데이터 블록의 송신 및 수신 방식은 통신망 제어 노드에 의한 I/O 처리 노드의 폴링이다. 전체 알고리즘은 다음과 같다.

① 통신망 제어 노드는 MCT(Macro Cycle Time, T_1)를 위한 타이머를 초기화하고, 데이터 스캔 타이밍 프레임을 브로드캐스트 한다.

② 정해진 시간, T_{scan} 후 통신망 노드는 NCT(Node Cycle Time)을 위한 타이머를 초기화 한 후 각 I/O 처리 노드에게 차례로 Sending Request Frame을 전송하여 매체 사용권을 부여한다. 이 때, 통신망 제어 노드 자신이 송신할 데이터를 이 Request Frame에 실어 보낼 수 있다.

③ 사용권을 부여받은 I/O 처리 노드는 자신의 송신 버퍼의 내용을 버퍼 Response Frame에 실어 네트워크에 브로드캐스트 한다. 이 버퍼 프레임은 모든 노드에게 알려지므로 필요한 노드에서는 이 프레임을 수신하여 값을 갱신, 참조할 수 있다.

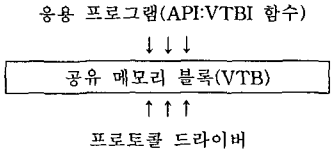
④ 통신망 제어 노드는 이 버퍼 Response Frame을 수신 한 후 NCT를 다시 초기화하고, 다음 노드에게 Sending Request Frame을 전송하여 매체 사용권을 부여한다. 만약, NCT가 종료될 때까지 Response Frame이 도착하지 않으면 그 노드는 문제가 있는 것으로 간주하고 다음 노드의 폴링에 들어간다.

⑤ 통신망 제어 노드는 마지막 노드에 대한 폴링을 마치고 NCT가 종료되면 데이터 처리/출력 타이밍 프레임을 브로드캐스트 한다.

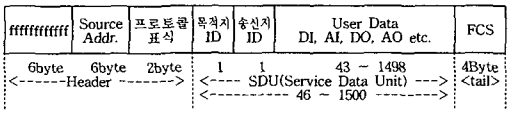
⑥ 통신망 제어 노드는 MCT 타이머가 종료되면 다음 주기의 MCT를 시작하여 ①에서 ⑥의 과정을 반복 수행한다.

2.3 응용 프로그램과의 데이터 교환

프로토콜층과 응용 프로그램 상호간은 미리 정의된 메모리 블록[VTB:Virtual Terminal Box]을 공유하면서 데이터를 교환한다. 그림 2.3은 이에 대한 개념도로서 응용 프로그램에서는 전용의 API[VTBI:Virtual Terminal Box Interface] 함수를 이용하여 공유 메모리 블록의 특정 위치에 접근하여 내용을 읽거나 기록한다.



(a) 데이터 교환



(b) 공유 메모리 블록의 내부 구성 예

그림 2.3 공유 메모리를 이용한 데이터 교환 개념도

프로토콜층과 응용 프로그램은 각 노드의 프로토콜층에서 운용하는 프로그램 수행 단계에 대한 상태 정보를 서로 공유하여, 프로그램 수행 동기를 맞춘다.

3. 시험 시스템 구성 및 성능 분석

이 절에서는 앞 절에서 설계한 내용을 토대로 발전소 시뮬레이터 시스템의 I/O 인터페이스 통신망 구축을 목표로 시험 시스템을 구축하고, 그 성능을 평가한다.

3.1 전송 데이터 용량 및 전송 주기

시험 시스템이 다룰 I/O 포인트 용량은 B화력 발전소 시뮬레이터 I/O 인터페이스 시스템이 다루는 I/O 포인트를 기준으로 AI 포인트를 추가하여 아래의 표 3.1과 같이 가정한다. 여기서 디지털 신호는 1비트, 아날로그 신호는 16비트를 사용한다.

표 3.1 전체 I/O 포인트 용량

신호종류	입력		출력			합 계
	AI	DI	AO	DO	LO/RO	
포인트수	635	1894	635	226	2502	5,892
바이트수	1,270	236.75	1,270	28.25	312.75	3,117.75
바이트합	1,506.75		1,611			

전체 I/O 포인트를 여러 개의 노드로 나누어 처리하면 노드의 수에 따라 전송에 필요한 바이트 수가 달라지지만 이더넷의 헤더, FCS(Frame Check Sequence), 프로토콜 설계시 사용한 그 밖의 오버헤드 등을 포함 노드당 약 500바이트 정도를 처리한다고 가정하여 4개의 노드로 분산 처리한다.

전체 I/O 포인트의 샘플링 주기는 B화력 발전소 시뮬레이터 데이터의 최소 갱신 주기인 83ms이다.

3.2 시험 시스템 구성

시험 시스템은 현장 계기 패널에 설치된 계기들로부터 데이터를 획득하여 외부 연결 노드(그림 3.1의 시뮬레이션 컴퓨터)를 통해 다른 계통에 데이터를 넘겨주거나, 역으로 외부 연결노드로부터 나오는 데이터를 현장 계기까지 전송하는 역할을 하는 I/O 인터페이스 시스템이다. 시험 시스템은 그림 3.1과 같이 다수의 PC(통신망 제어노드:1, 감시 노드:1, I/O 처리노드:4, 외부연결노드:1)로 구성되어 있으며, 채용한 운영체제는 윈도 95 및 NT이다.

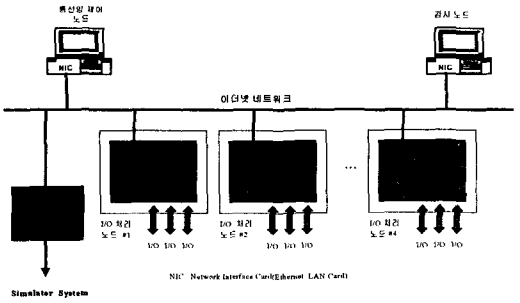


그림 3.1 시험 시스템 구성도

통신망 제어 노드는 네트워크에 접속되어 있는 모든 노드에게 각 단계에 맞는 타이밍 프레임을 전송하여 노드들의 프로그램 수행 단계를 동기화 시킨다. I/O 처리 노드는 장착된 DI, DO, AI, AO 등의 카드들로부터 신호를 획득하거나 알고리즘을 수행하여 이들 카드로 출력을 보내는 역할을 한다. 감시 노드는 각 노드의 정상 동작 여부나 특정 I/O 포인트의 추이를 감시하면서, 동시에 통신망 제어 노드가 고장났을 때 백업 역할을 수행한다. 그리고 외부 연결노드는 I/O 인터페이스 시스템과 독립된 다른 네트워크로 연결된 노드로서 이 노드는 윈도 NT로 구성되어 있으며, I/O 인터페이스 네트워크 상의 데이터를 이용하거나 다른 외부 네트워크상의 데이터를 각 I/O 처리 노드로 보낼 수 있다. 통신망 제어 노드는 외부 노드의 역할을 겸할 수 있으나 여기서는 외부 노드의 데이터 전송 시기만

을 조정하도록 한다. 감시 노드를 제외한 각 노드에서는 키보드를 제거하여 사용자에 의한 키보드 및 디스크 액세스를 방지하며, 통신망 제어 노드는 리얼타임 클록을 이용하여 데이터 스캔, 송신/수신, 처리/출력 타이밍을 맞추어 준다.

3.3 데이터 전송시 각 노드의 내부 처리 시간

표 3.2는 하나의 노드가 접속되어 있는 네트워크로부터 데이터 프레임을 받아들인 시점에서부터 생성한 데이터 프레임을 네트워크 상으로 내보낼 때까지의 내부 처리 시간인 데이터 버퍼 교환 시간 C_i를 측정해 보기 위해 2대의 동일한 PC간 상호 버퍼 교환 횟수를 측정한 수치이다.

표 3.2 두 노드간 상호 버퍼 교환 횟수(586 333A, μs)

프레임 크기 (헤더, FCS 제외)	초당 프레임수 (평균)	프레임당 처리시간	초당 프레임수 (최저)	프레임당 처리시간
46byte	7,260.81	137.73	7,171	139.45
486byte	1,649.01	606.42	1,629	613.87
986byte	879.74	1,136.70	869	1,150.75
1,500byte	593.84	1,683.96	586	1,706.48

두 노드 사이의 연결은 10Base2 케이블을 이용하여 연결하였으며, 거리는 10m 정도이다. 10Mbps 이더넷의 1bit 당 점유 길이는 20m 정도이므로 두 노드 사이의 순수한 데이터 전송시간은 무시할 만 하므로, 전체 시간이 노드 내 버퍼 교환에 걸린 시간이라고 가정한다. 여기에는 9.6 μs의 이더넷 interframe-gap이 포함한다. 여기서, C_i는 최대 약 1.7 ms(1500바이트 프레임) 이나, 이 수치는 PC의 성능에 따라 가변적이다.

3.4 성능 예측 및 실험 결과

3.4.1 데이터 전송시간 예측

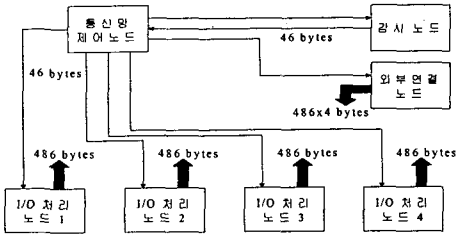


그림 3.2 시험 시스템의 데이터 전송 메카니즘

각 노드가 동일한 성능을 가질 경우 데이터 송신 및 수신 과정에 걸리는 $T_{network}$의 예측치는 정상적인 동작의 경우 그림 3.2를 참고하면, 식 (3.1)에서 약 8.1 ms로 예측된다.

$$\begin{aligned} T_{network예측} &= T_{I/O처리} + T_{감시} + T_{외부연결} \\ &= N\times(2T_p + T_{f1} + T_{f2} + C_{max1} + C_{max2}) \\ &\quad + (2T_p + 2T_{f1} + 2C_{max1}) \\ &\quad + \{2T_p + T_{f1} + T_{f2} + C_{max1} + C_{max2} \\ &\quad + (N-1)(9.6 + T_p + T_{f2})\} \\ &= 4\times 1265.52 + 439.1 + 2590.62 \\ &= 8091.8\ [\mu s] \end{aligned} \quad (3.1)$$

여기서, 통신망 제어노드가 송신하는 프레임은 46 바이트(이더넷 헤더, FCS 제외)이고, I/O 처리노드는 486 바이트의 데이터 프레임을 송신한다. I/O 처리노드는 모두 4 개(N=4), 이더넷 end-to-end propagation time T_p = 22.5 μs, 이더넷 프레임 전송시간 T_{fl} = 20.8(μs) + 0.8L(L=46바이트) = 57.6 μs, T_{f2} = 409.6(L=486) μs, 버퍼 교환 시간 C_{max1}(L=46) = 139.45 μs, C_{max2}(L=486) = 613.87 μs 이다.

NCT 예측치는 I/O 처리 노드 1265.52 μs, 감시노드 439.1 μs, 외부 연결 노드 2590.62 μs이나, 리얼타임 클록의 해상도(여기서는 2048회/초)를 고려하면 실제 NCT는 각각, 3/2048, 1/2048, 그리고 6/2048 초이다.

3.4.2 실험 결과

실제 실험은 T_{scan} = T_{out} = 0 으로 설정하고, T_l = $T_{network}$ = 19/2048 초로 설정한 후, 데이터 프레임의 송신과 수신 과정을 반복하면서 네트워크 상에서의 프레임 충돌여부를 관찰하였다. 실험 결과 네트워크에서 송수신 데이터가 프레임의 충돌 없이 무사히 전송됨을 네크워크 분석기를 통해 확인하였다.

한편, 전체 샘플링 주기 T_l = 83ms에서 T_{scan} 및 T_{out}에 약 반인 43ms를 할당하고 나머지 시간 40ms를 $T_{network}$에 할당한다고 가정하면 N=24 개까지 I/O 처리노드의 확장이 가능하다.

4. 결론

이 논문은 PC 기반의 제어 시스템에 사용할 이더넷 기반의 제어용 통신 프로토콜의 설계와 구현에 관한 내용을 다루었다. 프로토콜 설계 및 구현에 있어서 통신망 프로토콜의 하부계층에 이더넷을 채용하고, 그 상위에 NDIS를 이용하여 이더넷에 직접 접근할 수 있는 프로토콜을 새로이 개발하여 설치하였다.

분산된 시스템에서 응용 프로그램의 알고리즘 수행 시간과 프로토콜층의 데이터 전송 시간을 타이밍 프레임을 기준으로 처리하고, 이더넷 CSMA/CD 알고리즘을 보완하여 데이터 전송의 리얼타임성을 증진시켰다.

비교적 저렴한 PC 시스템과 상용 이더넷 카드의 채용은 PC의 비약적인 성능 개선과 이더넷이 가진 10Mbps의 빠른 전송속도 및 1,500여 바이트의 긴 데이터 프레임의 효과적인 이용은 고속·대량의 I/O 포인트를 다루는 다양한 분야의 I/O 인터페이스 시스템 구축에 용이성을 제공할 것으로 생각된다.

참고문헌

[1] A. Baker, "The Windows NT Device Driver Book", Prentice Hall, 1995.

[2] K.Y. Gwak, T.I. Jang, J.Y. Cho, and S.W. Lee, "A Development of Fieldbus Network for Power Plant Simulator", ICEE, 1998

[3] Sanjay Dhawan, "Networking Device Driver", Van Nostrand Reinhold, ITP A Division of International Thomson Publishing Inc. 1995

[4] 홍승호, "전력설비의 분산제어를 위한 필드버스 네트워크 구축 기술 연구", 전기학회 논문지, 제46권, 제4호, 1997