

필드버스를 이용한 생산자동화 시스템 구축 기술 연구

홍승호*, 박태진**

A Study on the Implementation of Fieldbus-Based Manufacturing Automation Systems

Seung Ho Hong*, Tae Jin Park**

ABSTRACT

Fieldbus provides real-time data communication among field devices in the manufacturing automation and process control systems. In this study, an experimental model of fieldbus-based manufacturing automation system is developed. Experimental model consists of two robots, two conveyor belts, NC machine, PLC, sensors and operator station. These machines are interconnected into the Profibus network, and exchange their data through the services provided by FMS(Fieldbus Message Specification), which is the application layer protocol of Profibus. The experimental model is used to measure the network-induced delay of variable and file data transmitted through FMS services. Network-induced delays are collected and analyzed on each sublayer of Profibus protocol stack. The results obtained from the experiment of this study can be effectively utilized when fieldbus is implemented on the practical manufacturing automation systems.

Key Words : Fieldbus(필드버스), Profibus(프로피버스), Manufacturing automation system(제조 자동화 시스템), Fieldbus Message Specification(필드버스 메시지 규격), Experimental model(실험 모델), Performance analysis(성능 해석)

1. 서론

컴퓨터와 통신기술의 발전에 힘입어 최근 수년간 생산자동화기술은 급속하게 발전하여 왔다. 첨단 생산자동화 시스템은 컴퓨터를 이용하여 단위 공정의 자동화를 이루고, 이들을 다시 수평적, 수직적으로 통합하여 전체 공정을 일관되게 관리함으로써 생산성 향상과 생산비용의 절감은 물론 생산공

정의 설계, 구축 및 유지 관리에 유연성을 제공한다. 첨단의 생산자동화 시스템을 구축하기 위하여서는 각 공정에서 필요로 하는 정보를 적시에 제공할 수 있는 정보의 통합화를 구현하는 것이 선행 과제로 대두된다. 통신망은 자동화 요소들 간에 정보 교환을 가능케 함으로써 첨단 생산자동화 시스템에서 중추 및 신경 기능을 담당하고 있다. 따라서, 궁극적으로 모든 공정의 통합화를 추구하는 첨

* 한양대학교 전자컴퓨터공학부

** (주)나라계전 기술연구소

단의 생산자동화 시스템에 있어서 네트워킹 기술은 가장 핵심이 되는 기술 가운데 하나라고 볼 수 있다.⁽¹⁾

1980년대 초반부터 생산자동화 환경에서 서로 다른 업체로부터 공급된 장비들 간의 통신을 위한 표준화된 네트워킹 시스템으로 MAP(Manufacturing Automation Protocol)이 개발되었다.^(2,3) MAP은 그러나 구조상 OSI(Open Systems Interconnection) reference model⁽⁴⁾에서 제시하고 있는 7계층을 모두 가지고 있다. 이러한 구조에서는 S/W로 구현되는 상위 계층에서 메시지 처리 시간이 커짐에 따라 전체 메시지 지연시간이 크게 증가되며, 따라서 MAP은 생산 현장의 필드에 설치된 각종 필드 장비들 간의 실시간 통신을 지원하기에는 적합하지 않은 시스템으로 인식되고 있다. 또한, OSI 모델은 원래 범용 통신망을 위하여 개발된 것으로, 공장자동화란 특수 환경에서 사용되기에는 프로토콜이 너무 복잡하고 공장자동화 환경에서 필요로 하지 않는 기능들을 많이 가지고 있다. 이러한 문제점들로 인하여 MAP은 80년대에 자동화시스템의 기간(backbone)통신망으로 사용되었으나, 90년대 이후 사용 빈도가 크게 줄어들고 있다.

Mini-MAP⁽⁵⁾은 기존의 MAP의 문제점을 보완하기 위하여 만들어진 프로토콜이다. 즉, 물리 계층과 데이터링크 계층상에 바로 응용 계층 프로토콜로 MMS(Manufacturing Message Specification)를 탑재한 축소된 구조를 갖는다. Mini-MAP은 그러나 데이터링크 계층으로 IEEE 802.4 토큰-버스를 채택하고 있다. IEEE 802.4 토큰-버스는 다른 시스템에 비하여 노드 당 네트워킹 접속 장치의 가격이 매우 비쌌 뿐만 아니라 전송매체로 동축케이블을 사용하기 때문에 설치비용 역시 매우 비싸지는 단점이 있다. 이러한 문제점들로 인하여 현재 MAP 또는 Mini-MAP 제품을 생산하는 회사도 매우 적으며, 따라서 관련 제품을 구하기 매우 어려운 실정이다.

1980년대 후반부터 생산현장에서는 필드에 설치된 각종 제어 및 자동화 관련 장비들에서 생성되는 데이터들의 실시간 통신을 지원하는 동시에 가격이 저렴한 네트워킹 시스템의 필요성이 제기되었으며, 이러한 목적을 위하여 개발된 네트워킹 시스템이 필드버스이다. 필드버스는 필드에 설치된 각종 센서, 단일루프제어기, 소형 PLC, 모터, 밸브, 로봇, CNC 등의 동작기계를 비롯하여 이러한 장비들을 제어하는 다중루프 제어기, 중대형 PLC 등의 자

동화기기에서 생성되는 데이터를 실시간으로 처리하며, 따라서 첨단 생산시스템의 네트워킹 구조상 가장 기본이 되는 네트워킹이다.^(6-8,17)

현재 기술 선진국에서는 필드버스를 이용한 자동화 시스템의 구축이 매우 활발히 진행되고 있다. 그러나, 국내에는 아직 필드버스를 이용한 자동화 시스템의 구축 기술이 축적되지 못한 상태에 있으며, 따라서 자동화 시스템에 필드버스를 도입하는 사례가 아직은 매우 드물다. 즉, 각종 산업 자동화 시스템에 필드버스를 도입하려 해도 관련 지식 및 기술의 부족으로 필드버스 구축 방법에 대하여 알지 못하고 있는 상태이며, 이것이 국내의 자동화 시스템에 필드버스를 도입하는데 있어서 가장 큰 장애 요인으로 작용하고 있다. 이러한 문제점을 해결하기 위하여 본 논문에서는 필드버스를 이용한 첨단 생산자동화 시스템 구축 방안을 제시하고자 한다.

본 연구의 선행 연구에서는 자동화 장비들을 필드버스 네트워크에 접속시키기 위한 핵심 기술인 필드버스 접속 소프트웨어의 구현 방법을 제시하였다.⁽⁹⁾ 필드버스 접속 소프트웨어는 Profibus^(10,11)의 응용계층인 FMS(Fieldbus Message Specification)를 기반으로 하여 개발되었으며 실시간 다중 작업 프로그래밍 환경 하에서 동작되도록 구현하였다. 본 연구에서는 선행 연구를 통하여 개발된 필드버스 접속 소프트웨어의 상위에서 동작되는 사용자 계층 접속 소프트웨어와, 이를 기반으로 동작되는 필드버스를 이용한 첨단 생산자동화 시스템의 실험 모델을 개발하였다.

본 과제를 통하여 개발된 생산자동화 시스템의 실험 모델은 2대의 로봇, 2대의 컨베이어벨트, PLC, NC 선반 및 오퍼레이터 스테이션 등으로 구성되며 이들은 모두 Profibus NIU(Network Interface Unit)를 통하여 필드버스에 연결된다. 본 연구에서는 Profibus의 FMS 서비스를 통하여 전송되는 파일과 데이터의 전송 지연시간을 계층별로 수집하여 이를 비교, 분석하였다. 본 연구를 통하여 도출된 각종 실험 결과는 추후 산업 현장의 생산 자동화 시스템에서 필드버스를 도입하는데 있어서 매우 유용하게 활용될 것이다.

2. Profibus 특성 및 구조

Profibus는 독일과 유럽에서 필드버스의 표준으

로 제정한 프로토콜로 이미 제품화가 완료되어 각종 자동화 시스템에서 널리 사용되고 있다. 본 장에서는 Profibus의 각 계층별 기능에 대하여 간략히 기술한다. Profibus의 계층 구조는 물리계층, 데이터 링크계층 및 응용계층으로 구성되며⁽¹²⁾, 사용자는 응용계층에서 제공되는 각종 서비스를 이용하여 사용자가 구축하려는 자동화 시스템의 환경에 적합한 사용자 계층의 Profibus 접속 프로그램을 구현하여야 한다.

물리계층은 데이터를 전송 신호로 변환하여 전송 매체를 통해 이를 전송하고, 수신된 전송 신호를 다시 데이터로 변환한다. 버스 토폴리지를 사용하는 Profibus는 리피터를 사용하지 않는 경우에 하나의 세그먼트에 최대 32개의 노드가 접속될 수 있으며, 3개의 리피터를 사용하여 노드 수를 최대 127개까지 확장할 수 있다. Profibus 규격서에는 최대 데이터 전송 속도가 200 m의 버스 길이에서 500 Kbps로 주어졌으나, 최근에는 1.5 Mbps와 12 Mbps의 전송 속도를 갖는 제품도 개발되었다.

Profibus의 데이터링크계층은 FDL(Fieldbus Data Link Layer)이라 하며, 매체접속제어(Medium Access Control) 기능과 논리링크제어(Logical Link Control) 기능을 제공한다. Profibus에서는 매체접속제어 방식으로 토큰-패싱 방식과 폴링 방식을 제공한다. 논리링크제어 기능은 통신을 하려는 노드들 간에 논리적 링크 또는 통신 관계를 설정 및 해제하는 기능을 비롯하여 통신 과정에서 발생할 수 있는 각종 오류를 처리함으로써 노드들 간에 신뢰할 만한 데이터 전송이 이루어지도록 하는 기능을 수행한다.

Profibus 응용계층에는 FMS(Fieldbus Message Specification), LLI(Lower Layer Interface) 및 FMA7(Fieldbus Management Layer 7)이 있다. FMS는 통신 객체와 응용 서비스를 정의하며, LLI는 응용 서비스와 FDL을 접속하는 기능을 가지고 있다. FMA7은 네트워크 관리 서비스를 수행한다. 응용계층에서는 전송할 데이터인 PDU(Protocol Data Unit)를 생성한다. 즉 사용자가 응용계층에게 데이터 전송 서비스를 요청하면 응용 계층에서는 이를 네트워크를 통하여 전송하기 위해서 PDU라는 패킷을 만든다. 일반적으로 패킷은 원하고자 하는 서비스에 대한 정보(서비스 종류 및 파라미터)를 포함하여야 한다. 이것은 Syntax라는 미리 정의된 구조를 갖게 되는데 Profibus에서는 ISO ASN.1(Abstract Syntax Notation One)을 따른다⁽¹³⁾.

FMS는 Profibus의 응용계층 가운데 필드버스의 사용자와 연결되는 부분이다. FMS는 산업 현장에서 사용되는 자동화 기기의 응용 프로그램에 접속되며, 이러한 기기 들은 상호 호환성 및 독립성 등이 보장되어야 한다. 그러므로 FMS를 구현하기 위해서는 객체 지향적인 방법이 사용된다. FMS는 네트워크 사용자에게 다음과 같은 통신 서비스를 제공한다.

- VFD 지원(Virtual Field Device Support)
- OD 관리(Object Directory Management)
- 연결 관리(Context Management)
- 변수 접속(Variable Access)
- 사건 관리(Event Management)
- 도메인 관리(Domain Management)
- 프로그램 기동(Program Invocation)
- 접근 보안(Access Protection)

FMS는 클라이언트-서버 관계로 동작되며, 따라서 응용계층의 통신 서비스를 수행하기 위해서는 서비스를 요구하는 클라이언트와 클라이언트의 서비스 요구에 응답하는 서버와의 관계를 설정하여야 한다. Profibus의 응용계층에서 클라이언트와 서버 관계는 통신 객체, 통신 서비스 및 통신 관계 등의 통신 요소들로 구성된다. 통신 객체는 각 노드 사이에서 교환되는 데이터를 말하며, 예를 들어 센서에서 측정되는 데이터를 비롯하여 제어기를 구동하기 위한 프로그램 및 데이터 파일 또는 통신 관련 파라미터 등이 여기에 포함된다. 한 노드는 여러 개의 통신 객체를 가질 수 있으며, 이러한 통신 객체의 목록을 OD(Object Dictionary)라 한다. OD는 Index에 의해 각 객체가 구별된다. 통신 서비스는 이들 객체들에 대하여 해당 노드의 응용 서비스를 수행한다. 통신 관계는 각 노드 간 서비스의 논리적 연결 기능을 수행한다. Profibus에서는 통신 관계의 목록을 CRL(Communication Relationship List)에서 관리한다. 이 목록은 CR(Communication Reference) 순으로 기입되는데 각 CR은 해당 노드의 응용 프로세서들 간의 데이터 교환을 위한 관계를 설정한다. 실제로 Profibus 네트워크를 설계하고 구성할 때 먼저 이들 통신 요소를 정의하고 각 통신 요소간의 관계를 설정해야 한다. 따라서 네트워크 시스템이 매우 복잡한 경우에 이러한 통신 요소를 구성하는 도구를 사용하여 네트워크 시스템의 구성을 단순히 처리할 수 있도록 할 필요가 있다.

3. 생산자동화 시스템 실험 모델 구축

본 연구의 선행 연구⁽⁹⁾에서는 자동화 장비들을 Profibus에 접속시키기 위한 소프트웨어가 개발되었다. Profibus 접속 소프트웨어는 여러 자동화 분야에서 산업용 컴퓨터로 널리 사용되고 있는 PC와, 모듈 형태로 구성되어 독립된 제어 기능을 수행할 수 있는 전용제어기인 IUC (Intelligent Universal Controller)⁽¹⁴⁾에서 동작될 수 있도록 개발되었으며, 여러 개의 응용 및 통신 서비스들이 동시에 수행될 수 있도록 하기 위하여 CTask와 OS-9 등의 실시간 다중 작업 프로그래밍 환경 하에서 동작되도록 구현하였다.

Profibus의 실제 사용자에게는 통신망의 하부 계층 구조와 동작은 중요하지 않고 단지 통신 상대와의 데이터 교환만이 중요한 의미를 갖는다. 그러므로 자동화 시스템의 사용자에게 의하여 개발되는 사용자계층의 응용 프로그램은 하위 계층의 통신 프로토콜들과의 데이터 교환만을 고려해 주면 된다. 즉 사용자 계층의 응용 프로그램은 응용 계층인 FMS와 FMA7 서비스를 이용하여 통신 상대에게 데이터를 전송하거나 수신할 수 있도록 구현하여야 한다. 본 연구에서는 Softing에서 제공하는 FMS와 FMA7⁽¹⁵⁾을 이용하여 생산자동화 시스템의 장비에서 동작되는 응용 프로그램을 구현하였다. 이러한 응용 프로그램은 선행 연구⁽⁹⁾를 통하여 개발된 Profibus 접속 소프트웨어 상에서 동작된다. 본 장에서는 Profibus를 이용한 생산자동화 시스템 실험 모델의 구성과 각 자동화 장비의 동작에 대하여 기술한다.

3.1 자동화시스템 실험모델의 구성

본 연구를 통하여 개발한 자동화 시스템 실험 모델은 Fig. 1과 같이 2대의 로봇, NC 머신, 컨베이어 벨트 및 오퍼레이터 스테이션 기능을 수행하는 PC로 구성된다. 따라서 본 자동화 시스템 실험 모델은 모두 5개의 스테이션으로 구성된다. 본 실험 모델에서 2대의 로봇은 IUC를 통하여 제어된다. IUC는 Profibus에 접속되고 IUC와 로봇구동기 간에는 RS-232C를 통하여 통신한다. 컨베이어 벨트에 부착된 센서와 모터는 Profibus 접속 기능과 PLC 기능이 내장된 Smart I/O를 통하여 Profibus에 접속된다. NC 머신은 PC에 의하여 제어되며, 이러한 PC 제어기는 Profibus-PC 접속 보드인 CP5412⁽¹⁶⁾ 보

드를 통하여 Profibus에 직접 연결된다. PC 제어기와 NC머신은 RS-485를 통하여 데이터를 교환한다. Fig. 2에는 본 연구를 통하여 개발된 자동화 시스템 실험 모델의 사진이 주어졌다.

본 실험 모델에 사용된 자동화 장비들은 모두 교육용 장비들로 그 기능과 성능이 제한되어 있다. 그러나 본 연구의 목적은 자동화 장비들 간에 통신 기능을 부과하는 것이며, 실험 모델에서 교육용 자동화 장비들 간에 통신 기능을 구현하는 데는 문제가 없다. 단, 본 실험 모델에서 사용한 교육용 NC 머신의 경우 PC 제어기의 프로그램이 개방화되어 있지 않으며, DOS 환경에서 단일 태스크로만 동작된다. 따라서 본 연구를 통하여 개발된 통신 소프트웨어와의 접속은 물론 다중 작업 또한 불가능하다. 이러한 문제점으로 인하여 본 모델에서 NC 머신의 구동은 PC 제어기에서 에뮬레이션되는 것으로 처리하였다. 실제 시스템에서는 그러나 Profibus 접속 기능을 지원하는 NC 머신을 사용함으로써 이러한 문제를 쉽게 해결할 수 있다.

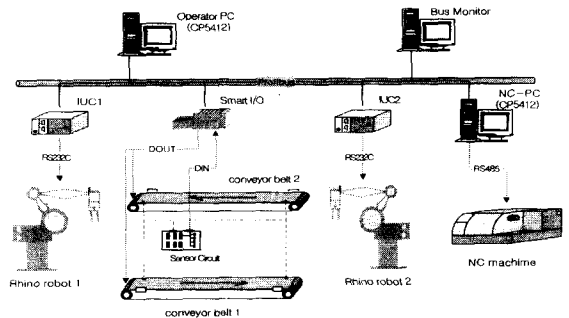


Fig. 1 Configuration of an experimental model of manufacturing automation system



Fig. 2 Picture of the experimental model of manufacturing automation system

본 실험 모델에서 자동화 시스템의 동작 시나리오는 다음과 같으며, 일련의 과정은 OP-PC(Operator PC)에 의하여 모니터링 된다.

- (1) 로봇 1이 원자재를 컨베이어 벨트 1에 올려 놓는다.
- (2) 센서가 원자재 도착을 감지하여 컨베이어 벨트 1을 구동시키고, 원자재가 컨베이어 벨트의 반대 방향에 도착하면 센서가 이를 감지하여 컨베이어 벨트 1을 정지시킨다.
- (3) 로봇 2가 원자재를 NC머신에 탑재한다.
- (4) NC 머신이 원자재를 가공한다(PC 에뮬레이션).
- (5) 로봇 2가 가공물을 NC 머신으로부터 컨베이어 벨트 2로 이동시킨다.
- (6) 센서가 가공물의 도착을 감지하여 컨베이어 벨트 2를 구동시키고, 가공물이 컨베이어 벨트의 반대 방향에 도착하면 센서가 이를 감지하여 컨베이어 벨트 2를 정지시킨다.
- (7) 로봇 1이 가공물을 컨베이어 벨트 2로부터 내려놓는다.

이러한 일련의 작업을 수행하기 위하여서는 자동화 장비들 간에 필요한 정보를 교환하여야 한다. 본 실험 모델에서 수행되는 장비들 간에 통신 관계 및 FMS 서비스를 정리하면 다음과 같다.

- OP-PC → IUC 1, 2 : 로봇 구동 프로그램 및 데이터를 다운 로드(도메인 관리)
- OP-PC → NC-PC : NC 구동 프로그램 및 데이터를 다운 로드(도메인 관리)
- Smart I/O → IUC2 : 원자재가 도착되었음을

로봇 2에 전달(변수 접속)

- IUC2 → NC-PC : NC 머신 구동 명령 전달 (프로그램 기동)
- NC-PC → IUC2 : NC 머신 상태 정보를 로봇 2에 전달(변수 접속)
- Smart I/O → IUC1 : 가공물이 도착되었음을 로봇 1에 전달(변수 접속)

Fig. 3은 전체 생산 자동화 시스템 실험 모델의 동작을 도식화한 것이다. 자동화 시스템에서 통신 기능이 수행되기 위하여서는 먼저 각 스테이션간에 통신관계가 설정되어야 한다. 이 때 각 스테이션들은 CRL(Communication Relationship List)을 참조해 사용 가능한 서비스, 최대 PDU 길이, OD 개정(Object Dictionary Revision) 등의 통신 요소들을 지정 받는다. 다음에는 각 장비들의 기능 및 동작이 기술된다.

3.2 OP-PC의 기능 및 동작 : station 1

OP-PC는 로봇과 NC 머신의 제어를 위한 도메인을 관리하는 기능을 수행한다. 로봇을 제어하기 위해 OP-PC는 먼저 로봇의 동작에 필요한 좌표값을 관리자로부터 입력받아 Inverse Kinematics를 수행하고, 이를 통해 얻어진 로봇의 각 관절의 회전 각도 값을 FMS의 도메인 서비스를 이용해 로봇 제어기와 연결된 IUC에 전송한다. Profibus에서는 바이트 단위의 데이터통신만이 허용되므로 이렇게 얻어진 데이터들을 다시 전송 가능한 형태로 변환시켜주어야 한다. 따라서 최종 변환된 데이터는 text mode형태의 파일로 저장된다. 이렇게 저장된 파일

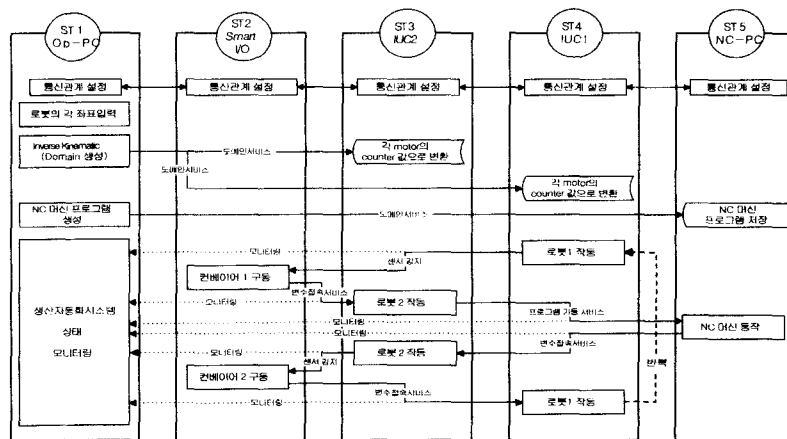


Fig. 3 Operation scenario of the experimental model of automation system

은 관리자에 의해 확인되고 임의 값으로 수정될 수 있어 관리자에 의한 로봇의 임의의 동작도 가능하게 된다.

NC 머신의 프로그램의 다운로드를 위한 도메인 전송에서는 NC 머신 구동 명령코드들로 구성되는 데이터 영역만을 전송하는 방식과 NC 머신 구동 명령 코드와 NC 머신 제어 실행 프로그램을 모두 전송하는 두 가지 방식으로 나누어 수행하였다.

OP-PC는 또한 시스템과 네트워크의 감시 및 관리기능을 수행한다. Fig. 3의 전체 시스템 동작 시나리오에 나타난 바와 같이 각 스테이션들은 그 상태가 바뀔 경우 FMS의 Write Variable 서비스를 통해 각 스테이션의 상태에 해당하는 변수 값을 OP-PC에 전송한다. OP-PC는 각 스테이션으로부터 수신된 변수 값을 참조하여 원자재의 위치 및 시스템의 공정상태, 시간 등을 화면에 출력하고, 이를 통해 얻어진 정보들을 기록파일에 저장한다. 관리자는 모니터링 프로그램에 의해 OP-PC의 화면에 출력되는 정보와 저장된 기록파일을 통해 전체 시스템을 감시하고 관리할 수 있다.

3.3 Smart I/O의 기능 및 동작: station 2

Smart I/O는 Profibus의 통신 기능을 제공하는 일종의 PLC이다. Smart I/O는 컨베이어 벨트의 양단에 부착되어있는 4쌍의 적외선 센서로부터 발생되는 신호를 디지털 입력 값으로 읽어들이며 물체의 위치를 인식하고, 디지털 출력 값을 조절함으로써 컨베이어 벨트의 동작을 ON/OFF제어하는 기능을 수행한다. 또한 Write Variable 서비스를 이용하여 로봇에 물체가 도착하였음을 알리는 기능도 수행한다.

컨베이어 벨트 1의 시작부분에 물체가 도착하면 발광부로부터 발생되는 적외선을 차단하게 되어 디지털 입력의 첫 번째 비트 값이 1로 바뀌게된다. 이로 인해 Smart I/O는 NC 머신으로 운반되어야 할 물체가 도착했음을 인지하고 컨베이어 벨트 1의 모터에 연결되어 있는 디지털 출력 값을 조절하여 컨베이어 벨트 1을 구동시킨다. 물체가 NC 머신 측 로봇에 도착하면 센서에 의해 디지털 입력의 두 번째 비트가 1이 되고, 컨베이어 벨트를 정지시킨다. 이 때 Smart I/O는 로봇 2와 연결된 IUC 2에 물체가 도착했음을 알리는 Write Variable 서비스를 수행한다. 컨베이어 벨트 2도 동일한 방법으로 동작된다.

3.4 IUC의 기능 및 동작: station 3

IUC는 Profibus의 네트워킹 기능을 갖춘 embedded controller로 로봇을 움직이므로써 컨베이어 벨트로부터 물체를 이동시키거나 컨베이어 벨트에 물체를 위치시키는 기능을 수행한다. IUC와 로봇 구동기는 RS232C 통신을 통하여 로봇 구동 신호를 전달한다. Smart I/O의 Write Variable 서비스를 통하여 가공을 원하는 원자재가 컨베이어 벨트의 NC 머신 측에 도착했음을 알리는 신호를 수신하면 IUC 2는 로봇을 이용해 물체를 컨베이어 벨트로부터 NC 머신으로 이동시킨다. 로봇의 동작이 완료되면 IUC 2는 FMS의 프로그램 기동 서비스를 이용해 NC 머신의 PC 제어를 구동시킨다. 다시 NC 머신으로부터 가공이 종료되었음을 알리는 Write Variable 서비스를 수신하면 IUC 2는 로봇을 움직여 물체를 NC 머신에서 컨베이어 벨트 2로 이동시킨다. 컨베이어 벨트에 의해 물체가 반대편 로봇 측에 도달하게 되면 IUC 1은 이를 알리는 Smart I/O의 Write Variable 서비스를 수신하게 된다. IUC 1은 로봇을 움직여 컨베이어 벨트위의 가공물을 처리 지점으로 옮긴 후 대기 위치로 귀환한다.

3.5 NC-PC의 기능 및 동작: station 5

본 연구에서는 프로그램 기동 서비스를 통한 NC 제어기 에뮬레이션 프로그램의 원격 제어를 위해 두 가지 방식을 제시한다. 첫 번째 방식은 NC-PC에 응용 프로그램을 탑재시키고 OP-PC로부터 도메인 서비스를 통해 NC 구동 명령 코드(데이터 영역) 만을 전송하여 NC 제어를 에뮬레이션하는 '해석 프로그램' 방식이다. 두 번째 방식은 NC 구동 명령 코드(데이터 영역)와 NC 에뮬레이션 실행파일(프로그램 영역)을 각각 도메인 서비스를 통해 OP-PC로부터 NC-PC로 전송하여 저장하고 이를 통해 NC 제어를 에뮬레이션하는 '실행 프로그램' 방식이다. 이 경우 NC 제어기 에뮬레이션 프로그램은 OP-PC에서 미리 NC-PC에 적합한 형태로 컴파일되어 실행 가능한 파일로 생성된다. 두가지 방식 모두 IUC 2로부터 수신되는 프로그램 기동 서비스에 의해 원격 제어된다. NC 구동 명령 코드는 OP-PC에서 관리자에 의해 문서편집기 등을 이용해 작성되어 파일로 저장된 후 도메인 서비스를 통해 NC-PC로 전송된다.

4. 메시지 지연시간 측정 실험 결과 및 고찰

본 연구에서는 생산자동화 시스템의 실험모델에서 자동화 장비들을 동작시키기 위해 FMS에서 제공하는 변수 접속 서비스와 도메인 서비스 및 프로그램 기동 서비스를 이용하는 경우에, 이러한 서비스의 수행과정에서 발생하는 메시지의 지연시간을 측정함으로써 생산자동화 시스템에서 Profibus 네트워크의 성능을 분석한다.

4.1 변수 접속 서비스 메시지 지연시간

Profibus의 데이터 링크 계층에서는 우선 순위 데이터 전송 기능을 제공한다. 선행 연구⁽⁹⁾의 해석 결과에 따르면 네트워크 파라미터들 가운데 TTRT의 변화가 메시지 지연 시간에 가장 큰 영향을 미치는 것으로 나타났다. Profibus에서 토큰은 논리 링(logical ring)을 따라 네트워크 내의 노드들을 차례로 방문한다. 각 노드는 토큰이 도착할 때까지 기다리고 있다가 토큰을 수신하면 대기하고 있던 데이터를 전송한다. Profibus는 high와 low의 두 가지 우선 순위를 제공한다. 각 노드에서 우선 순위 데이터의 전송은 네트워크 파라미터인 TTRT와 실제 토큰 순환 시간인 RTRT(Real Token Rotation Time)에 의하여 결정된다. 여기서 RTRT란 토큰이 논리 링을 한번 순환하는데 실제로 소요되는 시간으로 이는 매번 불규칙하게 변한다. 각 노드는 토큰을 수신한 후 TTRT와 RTRT 값을 비교한다. 만일 RTRT가 TTRT보다 큰 경우에는 high 우선 순위 메시지를 한 개만 전송하며, 반대의 경우에는 TTRT가 만료될 때까지 대기하고 있는 데이터들 가운데 높은 우선 순위 데이터부터 차례로 전송한다.

본 실험의 Write Variable 서비스를 통한 Variable Access 메시지의 지연시간 측정에서는 TTRT 변화에 따른 1바이트 메시지의 전송 지연시간을 측정하였다. 본 실험에서는 Fig. 4에 나타난 바와 같이 Smart I/O와 IUC간의 일-대-일 Write Variable 서비스 메시지 지연시간을 측정하였으며, 데이터 전송 속도는 500 Kbps이다. 본 실험에서 물리 계층과 LLI 계층에서의 메시지 지연시간은 Softing에서 제공하는 버스모니터라는 tool을 이용해 네트워크 상에 발생하는 프레임의 발생시간 차이를 측정하였다. FMS 계층에서 Write Variable 서비스의 메시지 지연시간의 측정은 서비스 프리미티브를 처리하는 함수의 실행순간에 임의의 신호를

직렬 포트에 발생시키도록 하여, 이 신호들의 시간차를 오실로스코프를 이용해 측정하였다.

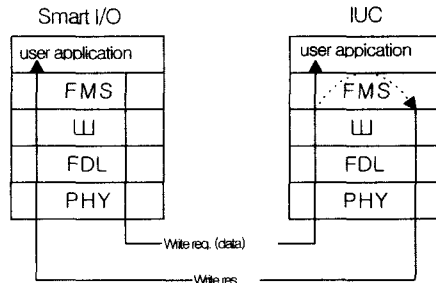


Fig. 4 Write Variable service between smart I/O and IUC

본 실험에서는 메시지의 지연시간을 Profibus의 계층 구조에서 PHY(Physical), LLI(Lower Layer Interface), FMS(Fieldbus Message Specification)의 세 단계로 나누어 분석하였다. 각 단계에서의 데이터 지연 시간은 다음과 같이 정의된다.

PHY 지연시간은 ① 토큰을 수신한 Smart I/O가 1바이트의 데이터 프레임을 물리 계층을 통하여 IUC로 전송하고 이에 대한 Ack가 Smart I/O의 물리 계층에 수신 완료된 시간과 ② IUC가 토큰을 수신하여 데이터 프레임에 대한 response 프레임을 물리 계층을 통하여 Smart I/O로 전송하고, 이에 대한 Ack가 IUC의 물리 계층에 수신 완료한 시간의 합이다. ①과 ②는 각각 서로 다른 token transaction에 의하여 수행된다.

LLI 지연시간은 Smart I/O의 LLI 계층에서 FDL 계층으로 데이터 전송을 요구하는 시점부터, Smart I/O로부터 프레임을 수신한 IUC의 LLI가 생성한 response 프레임이 Smart I/O의 LLI 계층에 완전히 수신된 시점까지의 시간이다. Smart I/O와 IUC에서 프레임 전송은 각기 다른 token transaction에 의하여 수행되며, LLI 지연 시간에는 Smart I/O에서 IUC로 토큰이 전달되는 시간과 IUC의 FMS 계층에서 메시지를 처리하는 시간이 모두 포함된다.

FMS 지연시간은 Write Variable 서비스를 수행하는 Smart I/O의 FMS 계층에서 LLI 계층으로 데이터 전송 request 프리미티브를 발생하는 시점부터, 데이터를 수신한 IUC가 생성한 response 프레임이 Smart I/O의 FMS 계층에 완전히 수신된 시점까지의 시간이다. 여기에는 IUC의 FMS 계층에서 수신

된 메시지를 처리하고, IUC가 토큰을 수신하는데 소요되는 시간이 모두 포함된다.

Fig. 5에는 TTRT의 변화에 대하여 PHY, LLI, FMS 계층에서의 평균 메시지 지연시간을 비교한 그래프가 나타나 있다. 그래프에 나타난 바와 같이 PHY 계층 데이터 지연 시간은 TTRT 변화에 대하여 영향을 받지 않는다. 이는 TTRT가 FDL 계층 파라미터이므로 그보다 하위 계층인 물리 계층에는 영향을 미치지 않기 때문이다. FDL 계층을 포함하는 LLI 계층에서는 그러나 TTRT 변화에 대하여 메시지 지연시간이 민감하게 영향을 받는다. TTRT 값이 증가할수록 각 노드에서는 데이터를 전송할 확률이 증가되며, 따라서 메시지 전송 지연 시간은 감소한다. 그림에서 LLI 계층의 평균 지연 시간이 증감을 반복하는 것은 LLI 지연시간이 IUC의 FMS 계층에서 처리시간을 포함하며, 이 처리 시간이 불규칙하게 변하기 때문이다. 그래프를 보면 FMS 계층의 지연 시간도 TTRT 변화에 대하여 영향을 받는 것으로 나타났다. 이는 본 실험에서 정의된 FMS 메시지 지연 시간이 FMS 계층에서 메시지를 처리한 후 IUC가 토큰을 수신하는데 소요되는 시간까지를 포함하였기 때문이다. 즉, TTRT 값의 변화에 따라 IUC가 토큰을 수신하는데 소요되는 시간이 달라질 수 있기 때문이다. Fig. 5에서 보면 TTRT 값을 적절히 설정함에 따라 Variable Access 서비스에 대한 데이터 지연시간을 반으로까지(55.4 msec에서 27.9 msec) 줄일 수 있는 것으로 나타났다.

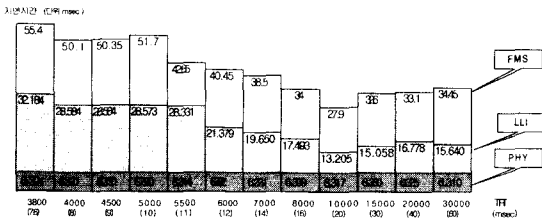


Fig. 5 Average delay at the FMS, LLI and PHY layers with the change of TTRT in the Variable Access Service

4.2 도메인 서비스 메시지 지연시간

도메인 서비스의 경우에는 일반적으로 도메인이 여러 개의 세그먼트로 분리되어 전송된다. 따라서 메시지 지연 시간이 도메인 자체의 크기뿐만이

아니라 세그먼트의 크기에도 영향을 받는다. 본 실험에서는 전체 도메인의 크기와 각 세그먼트의 크기 변화에 따른 메시지 전송 지연시간을 측정하였다. Fig. 6에는 OP-PC와 IUC 간에 도메인 서비스 과정이 나타나 있다. 메시지 지연시간은 PHY, LLI, FMS의 세 계층으로 나누어 분석하였다.

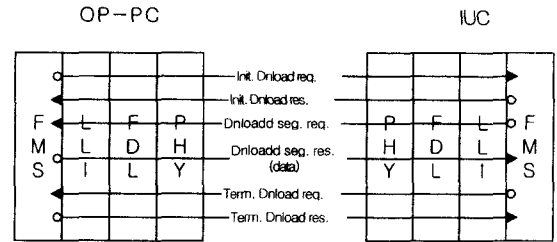


Fig. 6 Domain Download service between OP-PC and IUC

도메인 서비스의 PHY 지연시간은 Write Variable 서비스의 경우와 동일하며, 단지 IUC 측의 요구에 의하여 OP-PC에서 전송되는 데이터가 Write Variable 서비스의 1바이트 데이터가 아니라 세그먼트 단위로 나뉘어진 도메인 데이터라는 점이 다르다

LLI 지연시간은 initiate download 서비스에 의한 도메인 서비스의 초기화가 끝난 후 IUC의 LLI에서 OP-PC의 LLI에 세그먼트 데이터를 요구하는 시점부터 OP-PC의 LLI 계층에서 전송된 마지막 세그먼트 데이터가 IUC의 LLI 계층에서 수신 완료된 시점까지의 시간이다. 여기에는 OP-PC의 FMS에서 세그먼트 처리시간이 포함된다. OP-PC로부터 계속해서 전송되어질 세그먼트 데이터가 있을 경우 response 프레임의 more follows flag는 true 값을 가지고 전송된다. 이 경우 IUC와 PC는 모든 도메인 데이터의 전송이 완료될 때까지 세그먼트 전송 과정을 반복한다. 세그먼트 데이터는 각기 다른 token transaction으로 수행된다.

도메인 서비스는 서비스의 초기화를 위한 initiate download 서비스와 실제 도메인 데이터를 전송하는 download segment 서비스 및 도메인 서비스를 종료하는 terminate download 서비스로 이루어진다. FMS 지연시간은 OP-PC의 FMS가 initiate download 서비스를 통하여 도메인 서비스의 초기화를 완료하고, IUC가 download segment 서비스를 통하여 OP-PC에게 도메인 전송을 요구하여 마지막

세그먼트 데이터까지를 수신한 후, terminate download 서비스를 통하여 전체 도메인 서비스를 종료하는데 까지 소요되는 시간이다.

Fig. 7에는 도메인을 100 byte로 세그먼트 하였을 경우에 도메인 크기 증가에 따라 LLI 계층에서 메시지 세그먼트의 의 평균 지연 시간의 변화가 나타나 있다. ①은 IUC에서 PC로 세그먼트 데이터의 전송을 요구하는 서비스의 지연시간이고, ②는 PC로부터 IUC에 첫 번째 세그먼트 데이터를 전송하는 서비스의 지연시간으로 실제 메모리상의 데이터를 도메인 서비스를 이용해 전송하는 시간뿐만 아니라 PC의 FMS에서 세그먼트 처리시간도 포함된다. ③은 전체 도메인 데이터의 전송이 완료되지 않은 경우, 즉 more follows 가 있을 경우 다음 세그먼트의 전송을 요구하는 서비스의 지연시간이다. ④는 첫 번째 세그먼트의 전송을 제외한 나머지 세그먼트의 평균 지연시간이다. Fig. 7에서 보는 바와 같이 ④의 지연시간은 ②의 지연시간에 비하여 상대적으로 매우 적다. 이는 FMS의 도메인 서비스에서 more follows가 있을 경우 첫 번째 세그먼트 처리 시 후속 세그먼트에 대한 처리를 일괄적으로 수행하기 때문에 후속 세그먼트의 처리시간이 단축된다.

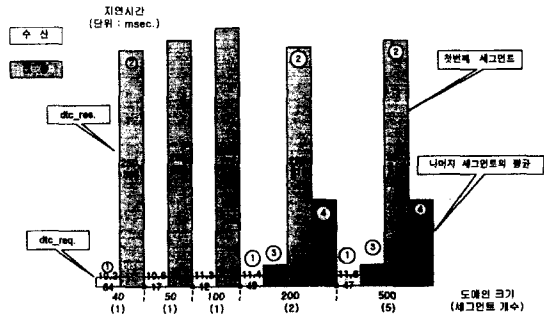


Fig. 7 Average LLI message delay in the Domain service

Fig. 8에는 도메인을 100 바이트의 세그먼트로 나누었을 경우에 도메인 크기 증가에 대하여 PHY, LLI, FMS 계층에서의 평균 메시지 지연시간을 비교한 그래프가 나타나 있다. Fig. 8에서 보는 바와 같이 도메인의 크기에 가장 민감하게 영향을 받는 계층은 LLI 계층이고, 나머지 PHY와 FMS 계층에서 메시지 지연시간은 LLI 계층 비하여 큰 변화를 보이지 않는다. 이는 PHY 계층에서의 변화

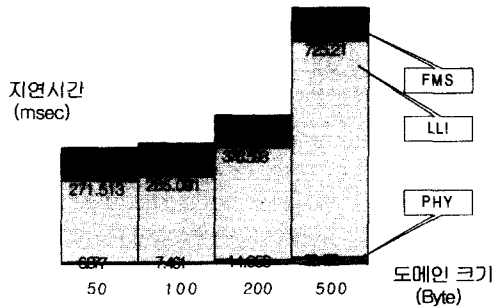


Fig. 8 Average delay at the FMS, LLI and PHY layers with the change of domain size(segment size: 100 bytes)

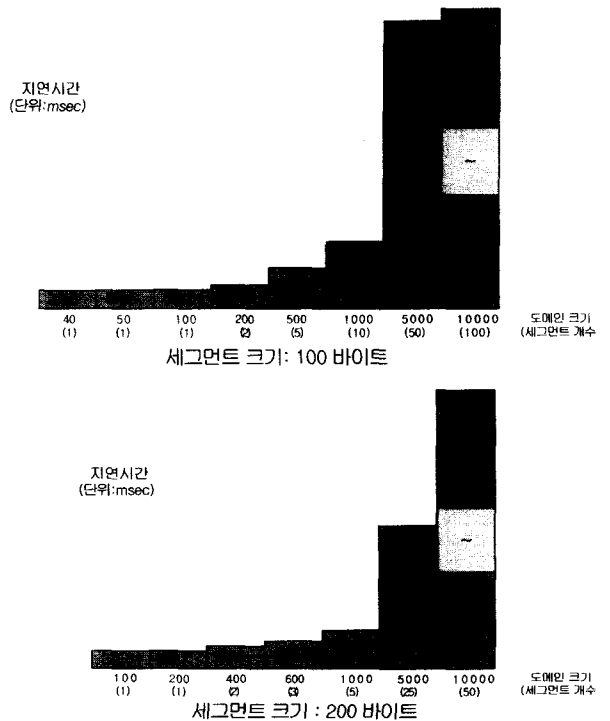


Fig. 9 Average delay at the FMS layer with the change of segment size

는 단순히 세그먼트를 전송하는 시간의 차이에 기인하고, FMS 계층에서는 Initiate_download와 Terminate_download 서비스만이 추가되는데 반하여 FMS의 도메인 서비스에서 more follows 세그먼트에 대한 처리가 LLI 계층 메시지 지연시간에 포함되었기 때문이다. Fig. 8에서 보는 바와 같이 도메

인 크기의 증가에 대하여 메시지 지연 시간은 지수 함수로 증가하는 경향을 보이고 있다.

Fig. 9에는 도메인을 100 바이트와 200 바이트의 세그먼트로 나누었을 경우에 도메인 크기에 대하여 FMS 계층에서의 평균 메시지 지연시간을 비교한 그래프가 나타나 있다. Fig. 9에서보면 도메인의 세그먼트 길이가 클수록 데이터 지연 시간이 감소하는 것으로 나타났다. 따라서 FMS의 도메인 서비스에서 데이터 지연 시간을 줄이기 위하여서는 세그먼트의 길이를 적절히 선정하여야 할 필요가 있다.

4.3 프로그램 기동 서비스 메시지 지연시간

프로그램 기동 서비스에는 Create_Program_Invocation, Delete_Program_Invocation, Start, Stop, Reset, Resume, Kill 등의 서비스가 포함된다. 본 실험에서는 프로그램 기동 서비스 중에서 IUC로부터 NC-PC로 프로그램의 원격 기동을 요청하는 Start 서비스 메시지에 대한 PHY, LLI, 사용자 응용 계층에서의 지연시간을 측정하였다. Fig. 10에는 Start 서비스 시 각 계층에서 OP-PC와 NC-PC 사이에 메시지가 교환되는 과정이 나타나있다.

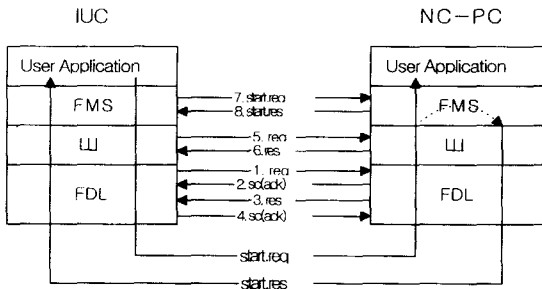


Fig. 10 Start service between IUC and NC-PC

PHY 계층에서 Start 서비스 메시지의 지연시간은 4.1절의 변수 접속 서비스의 경우와 동일하며, 단지 전송되는 프레임에 데이터 영역이 없이 단지 서비스 영역만 전송된다는 점이 다르다.

LLI 지연시간은 IUC의 LLI에서 NC-PC에 프로그램의 원격 기동 서비스를 요구하는 시점부터, 이에 대한 응답으로 NC-PC가 전송한 response 프레임을 IUC의 LLI 계층에서 완전히 수신하는 시점까지의 시간이다. 이들은 각기 다른 token transaction에서 이루어지며, LLI 지연 시간에는 IUC에서 NC-PC로 토큰이 전달되는 시간과 NC-PC의 FMS 계층에

서 메시지를 처리하는 시간이 모두 포함된다.

사용자 응용 계층 메시지 지연시간은 IUC의 응용 프로그램에서 NC-PC로 프로그램의 원격 구동을 요구하는 시점부터 NC-PC에서 NC 제어기 에뮬레이션 프로그램이 반응하기 시작하는 시점까지의 지연시간으로 정의된다.

실행 프로그램 방식에서는 도메인 서비스를 통해 미리 20 Kbyte 가량의 실행 프로그램 파일을 도메인 서비스를 이용하여 OP-PC로부터 NC-PC로 미리 전송하여야 하며, 이로 인해 약 23700 msec의 지연시간이 소요되게 된다. 따라서 실행 프로그램 방식은 해석 프로그램 방식에 비해 전체 시스템 자원의 낭비를 초래할 수 있다. 그러나 실행 프로그램 자체를 통신망으로 전송할 수 있어 여러 응용 프로세스에 대한 프로그램 기동 서비스의 유연성을 향상시킬 수 있다. 반면 해석 프로그램 방식에서는 응용 프로세스의 변화에 의해 해석 프로그램의 연산 방식이 바뀔 경우 관리자가 NC-PC에서 직접 프로그램을 수정하고 이를 다시 컴파일하여 동작시켜야 한다. Fig. 11에는 프로그램 기동서비스의 Start 서비스 메시지의 평균 지연시간이 해석 프로그램과 실행 프로그램에 대하여 측정된 결과가 나타나 있다. 결과에서 보듯이 두 방식에 의한 메시지 지연시간의 차이가 크지 않은 것으로 나타났다.

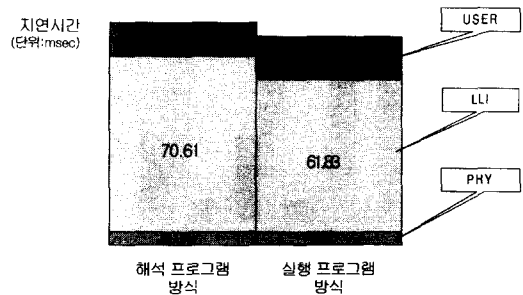


Fig. 11 Average delay of Program Invocation message(Start service)

5. 결론

본 연구에서는 Profibus 프로토콜을 생산자동화 시스템에서 실시간 통신을 지원하기 위한 네트워크 시스템으로 도입하는 경우에, Profibus의 응용 계층인 FMS를 이용하여 첨단 생산자동화 시스템을 구축하는 방법을 제시하였다. 본 연구를 통하여 개발된 첨단 자동화시스템의 실험 모델은 로봇, 컨베이어

어 벨트, NC머신, PLC, 센서 및 오퍼레이터 스테이션으로 구성되며, 본 연구에서 제시한 첨단 생산자동화 시스템 구축 방안은 실제 산업 현장의 자동화 시스템에 Profibus를 도입하는데 바로 적용될 수 있을 것이다.

본 연구에서는 또한 개발된 실험 모델을 이용하여 FMS 서비스를 통해 전송되는 메시지의 전송 지연시간을 계층별로 수집하여 이를 비교, 분석하였다. 먼저, Profibus의 FDL 계층에서 네트워크의 성능에 가장 민감한 영향을 미치는 네트워크 파라미터인 TTRT의 변화가 응용 계층의 FMS 서비스를 통하여 전송되는 변수 데이터의 전송 지연 시간에 미치는 영향을 계층별로 실험적으로 분석하였다. 실험 결과에 의하면 물리 계층 데이터 지연 시간은 TTRT 변화에 대하여 영향을 받지 않으나, FDL 계층을 포함하는 LLI 계층에서는 데이터 지연 시간이 TTRT 변화에 대하여 민감하게 영향을 받는 것으로 나타났다. 또한, FMS 계층의 지연 시간도 TTRT 변화에 대하여 영향을 받는 것으로 나타났다.

본 연구에서는 또한 FMS의 도메인 서비스를 통하여 파일 데이터를 전송하는 경우에 메시지의 전송 지연시간을 계층별로 수집하여 이를 비교, 분석하였다. 도메인이 두 개 이상의 세그먼트로 분리되는 경우에 첫 번째 세그먼트의 지연시간이 두 번째 이후부터 전송되는 세그먼트들의 평균지연시간에 비하여 상대적으로 큰 값을 가지는 것으로 나타났다. 또한, 도메인의 크기가 증가함에 따라 데이터 지연 시간이 가장 민감하게 영향을 받는 계층은 LLI 계층이며, 세그먼트 크기가 클수록 전체 도메인을 전송하는데 소요되는 지연 시간이 감소하는 것으로 나타났다. 따라서 FMS의 도메인 서비스에서 세그먼트의 길이를 적절히 선정하여야 할 필요가 있다.

프로그램 기동 서비스를 통한 NC프로그램의 원격 기동 실험에서는 해석 프로그램 방식과 실행 프로그램 방식에 의한 구현 방법을 제시하였으며, 두 방식간에 서비스 지연시간의 차가 크지 않음을 실험적으로 알 수 있었다. 해석 프로그램 방식은 네트워크 자원의 효율적인 관리가 가능하나 응용 분야에 대한 유연성을 제공하지 못하는 반면, 실행 프로그램 방식은 여러 응용 프로세스에 적용될 수 있는 유연성을 제공한다. 그러므로 프로그램 기동 서비스를 통한 프로그램의 원격 제어는 실제 적용

시스템의 특성을 고려하여 적합한 방식으로 구현되어야 한다.

본 연구에서는 필드버스를 기반으로 하는 자동화 시스템의 실험 모델을 개발하여 생산 자동화 시스템에서 필드버스를 도입하는데 필요한 기반 기술을 확보하고, 이를 실제 산업 현장의 생산 자동화 시스템에 적용하기 위한 방안을 제시하였다. 실제 산업 현장에서 장비들이 추가되고, 프로세스가 복잡해지는 경우에도 OD 관리 서비스를 통하여 새로운 객체를 등록하고, 연결 관리 서비스를 통하여 새로운 통신 관계를 설정하는 작업만이 추가될 뿐, 본 논문에서 제시한 시스템 구축 방법이 그대로 적용될 수 있다. 본 연구에서는 Profibus의 FMS 서비스를 이용하여 자동화 시스템을 구축하는데 있어서 네트워킹으로 인하여 유발되는 데이터 지연시간의 특성을 이해할 수 있었으며, 본 연구를 통하여 도출된 각종 실험 결과는 산업 현장에서 실제로 Profibus를 이용한 자동화 시스템을 설계 및 구축하는데 있어서 매우 유용하게 활용될 수 있을 것이다.

후기

본 연구는 한국과학재단 핵심전문 연구비(961-0924-138-2)지원으로 수행되었음.

참고문헌

1. Piementel, J., *Communication Networks for Manufacturing*, Prentice Hall, 1990.
2. MAP 3.0 Specification 1993 Release, World Federation of MAP/TOP Users Groups, 1993.
3. 홍승호, "MAP: 공장자동화를 위한 네트워크의 표준", 대한기계학회지, 제35권, 제5호, pp. 427-441, 1995.
4. *Basic Reference Model for Open System Interconnection*, ISO 7498, 1984
5. McMillan, A. and Gardner, C., *Mini-MAP '93*, Open I.T. Corp. 1994
6. Jordan, J., *Serial Networked Field Instrumentation*, John Willey & Sons, 1995.
7. 홍승호, "필드버스 기술 동향", 제어·자동화·시스템공학회지, 제4권, 제6호, pp.13-18, 1998.
8. InTech: Field Buses Special Issue, ISA Publication,

November 1996.

9. 김기암, 홍승호, “자동화 시스템에서 Profibus 네트워크 인터페이스 구현 및 성능 평가” 제어·자동화·시스템공학논문지, 제4권, 제1호, pp. 113-122, 1998.
10. DIN 19 245 Profibus Standard Part 1: 1991, Profibus Trade Organization
11. DIN 19 245 Profibus Standard Part 2: 1991, Profibus Trade Organization
12. Bender, K., PROFIBUS-The Fieldbus for Industrial Automation, Carl Hanser Verlag & Prentice Hall, 1993
13. Dickson, G. and Lloyd, A., Open Systems Interconnection, Prentice Hall, 1992.
14. IUC User's Manual, PEP Modular Computers, 1993.
15. PROFIBUS Application Program Interface User Manual, Version 5.00, Softing GmbH, 1996.
16. Hardware Description PROFI-IF-PCAT, Softing GmbH, September 1993.
17. 최형섭, 장석호, 이석원, 남부희 “FMS을 위한 필드버스 시스템 구현” '97한국자동제어학술회의논문집, pp. 399-402, 1997.