

실시간 제어 데이터통신을 위한 IEEE-1394 용 수정 TCP/IP 의 성능분석

Performance Analysis of Modified TCP/IP for Realtime Control Data Transmission over IEEE-1394 Network

염복진*, 윤기중, 박재현
(Bok-jin Youm, Gi-Jung Yun, and Jaehyun Park)

Abstract : A real-time network in a distributed control system plays an important role for the reliable data transmission. Compared to the field-buses used in the past, TCP/IP protocol on the top of Ethernet provides a compatibility between applications as well as an economical method to develop softwares. This paper proposes a modified TCP/IP structure for IEEE-1394 network, with which asynchronous and isochronous data transmission is selectively used for the real-time data transmission in a distributed control system. This paper also shows the performance of the proposed protocol by experiments.

Keywords : real-time network, IEEE-1394, TCP/IP

I. 서론

최근 고성능 제어시스템의 주류를 이루고있는 분산 제어 시스템의 일반적인 특징은 '효과적인 실시간 데이터 전송', '데이터 전송의 중심 없이 이루어지는 1대1 통신', 그리고 '플러그 앤 플레이와 같이 쉽게 재구성 가능한 구조를 가져야 한다'는 특징이 있다. 이러한 분산제어시스템을 구성하는 네트워크로 과거에는 시스템마다 특징적인 필드 버스를 사용하였다. 필드버스는 각 표준에 한정적으로, 타 시스템과의 호환성이 없으며, 이를 구현하는데 있어 많은 시간과 비용을 요구한다. 이러한 이유로 필드 버스로 분산 제어 시스템을 구현하는데 많은 어려움이 있었다. 최근에는 필드버스의 고비용, 비호환성의 단점을 극복하기 위해 TCP/IP와 이더넷(Ethernet)을 널리 사용한다[1,2,13]. 이들은 전 세계적으로 많이 사용되고 있는 표준 프로토콜이며, 구현에 있어 경제적이다. 또한 벤더에 독립적인 표준을 지원하며, 많은 소프트웨어가 지원된다는 장점이 있는 반면에, 제어 시스템의 데이터 전송에 효과적이지 못하며, 실시간성을 보장하지 못하므로 많은 문제를 야기한다.

TCP/IP 프로토콜은 데이터의 신뢰성 있는 전송에 중점을 두기 때문에, 프로토콜의 구조가 원거리 통신에 적합하게 설계되어 있다. 이것을 제어 시스템에서 사용하는 것은 단지 표준 프로토콜이라는 장점을 활용한 것이다. TCP/IP의 비실시간성을 보완하기 위해 고안된 프로토콜은 RTP (Real-time Protocol)[4]와 RSVP (Resource ReSerVation Protocol) 가 있다[5]. RT는 UDP (User Datagram Protocol)[3]를 기반으로 구현되어 있기 때문에 어떠한 전송 제어도 제공하지 못한다는 문제점

을 가지고 있다. RSVP는 전송 흐름에 대한 QoS (Quality-of-Service)를 보장 받지만 그것으로 실시간성을 보장 받는 것은 아니다. 또한, 이더넷에 사용되는 CSMA/CD (Carrier-sense Multiple Access with Collision Detect) 프로토콜은 구조적으로 전송 지연 시간의 비 예측성이 존재하기 때문에 실시간 데이터의 전송에 많은 문제점을 야기한다. 그러므로, 이더넷상의 TCP/IP를 이용하여 실시간 데이터를 신뢰성 있게 전송하는 것은 거의 불가능에 가깝다.

일반적인 실시간시스템과 달리 분산제어 시스템에 국한하여 전송되는 데이터를 살펴보면, 다음과 같이 데이터를 분류할 수 있다.

- 실시간 데이터 (Real-time data-hard real-time): 시스템의 각종 이벤트와 경고 신호 같이 데이터 길이가 짧고, 시간 제약을 받는다.
- 주기적 데이터 (Periodic data - soft real-time): 피드백 제어를 위한 센서 데이터와 같은 주기적으로 발생되며, 일정시간 이내에 전송되어야 한다.
- 다량의 비 실시간 데이터 (Bulky non-real-time data): 프로그램 및 데이터 파일과 같은 데이터의 길이가 길고, 시간 제약을 받지 않는다.

이들 데이터 중 제어용 데이터는 주기적인 데이터가 대부분을 차지하고 있으므로, 효과적인 데이터 통신을 위하여 이들 주기적인 데이터에 초점을 맞춘 통신망의 구축이 필요하게 된다.

한편, 고속의 멀티미디어 통신을 위한 네트워크로 IEEE-1394를 생각할 수 있는데, IEEE-1394 네트워크는 400MBPS의 높은 전송대역과 주기적인 데이터 통신을 지원하는 특징을 가지고 있다. 따라서 IEEE-1394의 이러한 특징은 분산제어 시스템의 기본 네트워크로 사용되는데 적합하다고 할 수 있다. 그러나 가장 널리 사용되는 프로토콜인 TCP/IP는 IEEE-1394의 주기적인 통신을 효과적으로 지원하지 않는 구조로 되어있어 단순한 TCP/IP - IEEE1394의 조합은 실시간 분산제어에 적합한 조합이라 할 수 없다.

* 2 임지자(Corresponding Author)

논문 접수 : 2003. 9. 26., 채택확정 : 2004. 1. 8.

염복진 : 현대중공업 기계전기연구소(comouse@resl.inha.ac.kr)

윤기중, 박재현 : 인하대학교 정보통신공학부

(gyj_in90@resl.inha.ac.kr / jhyun@inha.ac.kr)

※ 본 연구는 산업자원부지원 차세대신기술개발 사업인 "퍼스널로봇을 위한 시스템엔지니어링 기술개발"의 지원으로 수행되었음.

본 논문에서는 IEEE-1394 네트워크의 특징을 효과적으로 사용할 수 있도록 TCP/IP 프로토콜을 수정하여 TCP/IP의 손쉬운 프로그래밍 인터페이스와 IEEE-1394의 효율적인 실시간 주기적 통신기능을 동시에 지원하는 네트워크 구조를 제안한다. 제안하는 TCP/IP는 현재 널리 사용중인 표준을 확장한 것으로 기존의 표준 프로토콜스택과 호환성을 유지하며 IEEE1394의 주기적 전송기능을 최대한 이용하도록 설계되었다. 본 논문의 구성은 우선 2장에서 IEEE1394의 기본적인 특징을 소개하고, 이의 주기적 전송기능을 살리기 위한 확장된 TCP/IP 구조 제안을 3장과 4장에서 하며 5장에서는 성능 분석 결과를 기술한다.

II. 고성능 시리얼 버스 IEEE 1394

1. IEEE 1394 개요

IEEE 1394는 패러럴 버스 또는 패러럴 백플레인 버스에 사용하기 적당한 고속, 저가의 시리얼 버스이다[8,9]. 또한 IEEE 1394는 핫-플러그와 플러그 앤 플레이를 지원하며, PC와 같은 특정한 호스트 없이 피어-투-피어 방식으로 연결의 유연성을 갖고 있다. IEEE 1394는 다음과 같은 서로 완전히 다른 두 가지 데이터 전송 방식을 지원한다.

- 비동기 전송(Asynchronous transfer): 제어와 셋업을 목적으로 하는 짧은 메시지에 주로 사용되며, 버스 대역폭의 최소 20% 이상이 사용 가능 하다. 데이터 교환은 요청(request)과 응답(response)의 구조로 이루어지며, 각 패킷은 아비트레이션에 의해 전송을 시작하고, 패킷을 받은 수신자는 ACK 패킷을 바로 전송 하게 되어 시간보다 신뢰성이 중요한 곳에 사용된다.
- 동시 전송(Isochronous transfer): 고정된, 또는 보장된 대역폭으로 다량의 데이터를 전송할 때 사용되며, 버스 대역폭의 최대 80%까지 사용 가능하다. 데이터의 교환은 사이클 마스터(Cycle Master, 이하 CM)에 의해 125us(8kHz)마다 사이클 스타트(cycle start) 패킷이 전송되고, 아비트레이션에 의해 전송을 시작한다. 수신자는 패킷이 유효하고, 채널이 활성 되었을 때만 받아 들이고, 어떠한 ACK 패킷도 보내지 않는다.

IEEE 1394 네트워크 프로토콜은 트랜잭션, 링크, 물리 계층으로 구성되는데, 트랜잭션 계층은 'read', 'write' 그리고 'lock' 트랜잭션을 제공 하지만, 동시전송에 대한 트랜잭션은 제공하지 않는다.

- Read: 응답자의 특정 주소의 데이터를 요구자에게 전송한다.
- Write: 요구자의 데이터가 응답자의 특정 주소에 전송 된다.
- Lock: 요구자의 데이터가 응답자에게 전송되어, 응답자가 이를 처리한 후, 요구자에게 다시 전송한다.

링크 계층은 트랜잭션 계층에서 승인된 데이터그램의 단방향 데이터 전송 기능을 제공한다. 어드레싱, 데이터 체크, 전송을 위한 패킷 구성과 수신을 제공하며, 또한 isochronous를 위한 서비스를 제공한다.

- 아비트레이션 순서: 요구 패킷을 전송하려는 노드는 물리 계층에서 버스의 게인 컨트롤을 한다.
- 비동기 데이터 패킷 전송: 소스 노드는 데이터 prefix 신호(속도 코드를 포함한), 출발지와 목적지의 주소, 트랜잭

션 코드, 트랜잭션 라벨, 재시도 코드, 데이터와 CRC (Cyclic redundancy Check)를 전송한다.

- Isochronous 데이터 패킷 전송: 짧은 채널 확인자, 출발지와 목적지의 주소를 전송하지만 트랜잭션 라벨과 재시도 코드는 전송하지 않는다.
- Acknowledgment: 유일한 목적지 주소를 갖는 패킷은 수신자가 리턴 코드를 전송한다. Isochronous 패킷이거나 asynchronous의 방송 패킷은 어떠한 acknowledgment도 전송하지 않는다.

물리 계층은 'data'와 'strobe' 신호로 이루어져 있으며, 한번에 오직 하나의 노드만이 데이터를 전송할 수 있도록 보장하고, 시리얼 버스 데이터 스트림과 신호 레벨을 링크 계층에서 요구한 노드로 보낼 수 있도록 하기 위한, 초기화와 아비트레이션 기능을 제공한다. 버스 관리는 버스 매니저(Bus Manager, 이하 BM)와 동시 자원 매니저(Isochronous Resource Manager, 이하 IRM)가 있다. BM은 '진보된 버스 전원 관리', '스피드 맵의 유지', '버스 토폴로지의 유지' 등의 서비스를 제공하며 IRM은 'isochronous 대역폭의 할당', '채널 번호의 할당', 'CM의 선택' 등의 서비스를 제공한다. Isochronous 자원인 대역폭과 채널의 할당은 IRM에 의하여 관리되는데 버스 리셋 후 1000ms 이후에 할당이 가능하다. 대역폭의 계산은 IEEE 1394의 속도가 S1600 (16*98.304 Mbit/s)일 때를 기준으로 한다. 할당 대역폭의 계산은 전송에 필요한 오버헤드와 데이터 전송 시간을 고려하여 계산한다. 즉, 'isochronous gap' (4bit), 'arbitration' (16*32bit), 'data prefix' (16bit), 'data' 그리고, 'data end'(24bit)가 포함된 전송 시간을 계산 해야 한다.

2. TCP/IP over IEEE 1394

'IPv4 over IEEE 1394'에서는 인터넷 프로토콜 버전 4 (Internet Protocol version 4 이하IPv4) 데이터그램의 구조와 전송 방법을 정의하고 있다[6]. 또한, 1394 시리얼 버스에 맞춘 Address Resolution Protocol (이하 ARP)의 데이터 구조와 Multicast Channel Allocation Protocol (이하 MCAP)를 제안하고 있다. 또한 'Transmission of IPv6 Packets over IEEE 1394 Networks'에서는 IPv6 패킷을 전송하는데 필요한 링크 계층의 주소지정 방법을 제안한다. IP 데이터그램은 IEEE 1394의 Global Asynchronous Stream Packet (이하 GASP)을 사용하며, 이 GASP의 MTU는 전송 속도에 따라 다르다 (S400일 때 2048). 그리고, isochronous 전송에 의해 전송 되는 MTU는 GASP의 두 배 값을 갖는다.

III. 전송 방법 선택형 IP

인터넷 프로토콜은 packet-switched 컴퓨터 통신 네트워크를 위한 시스템의 연결에 사용하기 위해 설계되었다. IP는 출발지에서 목적지까지 데이터의 블록이라고 불리는 데이터그램 패킷을 전송하여 네트워크 시스템을 연결해주는 기능을 제공한다. 일반적인 호스트-호스트 프로토콜에서 볼 수 있는 신뢰성, 흐름제어, 순서화 등의 메커니즘은 없다. 따라서 본 논문에서 제안한 수정된 IP(Modified IP 이하 M-IP)에서는 이러한 IP의 기능과 IEEE 1394의 특징적인 전송 방법을 고려한다. 서로 다른 두 전송 방법을 선택하여 IP datagram의 전송을 하기 때문에 실시간 전송이 가능하며, IEEE 1394의 자원을

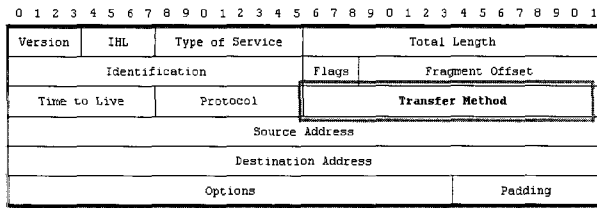


그림 1. M-IP 헤더.
Fig. 1. M-IP Header.

효율적으로 사용하게 한다.

1. 수정된 IP 헤더

M-IP 헤더는 IEEE 1394의 완전히 서로 다른 두 전송 방법을 효율적으로 사용할 수 있는 구조로 되어 있다. 그림1은 M-IP 헤더를 보여주고 있다. Transfer-method 필드에서 비동기 전송과 동기 전송 방법 중 선택한다. 실시간성을 보장해야 하는 데이터는 M-TCP(Modified TCP; 4장에서 설명됨)에서 TS 플래그를 사용하여 기한(deadline)을 표기하고, M-IP에서는 이를 시간 내에 전송하기 위한 방법을 선택한다. 실시간 데이터는 hard real-time이므로 소켓 생성 초기부터 isochronous 채널과 대역폭을 할당 받는다. M-IP에서는 isochronous 전송을 사용하기 위한 플래그와 대역폭을 헤더 정보에 가지므로 목적지에 전송 기한 내에 데이터그램을 전달하게 된다. 주기적 데이터는 soft real-time으로 전송 기한을 크게 넘지 않는 범위 내에서 asynchronous와 isochronous의 전송 방법을 조합하여 사용한다. 다량의 비실시간 데이터는 기존 IP와 같은 구조로 asynchronous 전송 방법을 사용한다.

IPv4상의 checksum 필드는 패킷 핸들링 과정의 지연을 야기하므로 비교적 안정적이 통신망인 IEEE-1394에서는 이를 사용하지 않을 수 있다. 이는 IPv6의 "Header format simplification"에서도 볼 수 있다[7]. 제거된 checksum 필드 영역은 IEEE-1394 네트워크의 동기 전송을 위해 사용된다. Transfer-method 필드는 다음과 같이 구성된다.

- ISO(1bit) : 데이터그램이 isochronous 전송에 의해 전송 되는 패킷임을 의미한다. ISO 플래그에 의한 isochronous 전송은 데이터그램의 대역폭과 채널을 IRM에게 할당 받아야 한다. IP 계층 상에서 isochronous 전송은 로컬 대역폭 관리자에 의해 관리 된다.
- ASP(1bit) : 데이터그램이 asynchronous 전송에 의해 전송 되는 패킷임을 의미한다. ASP 플래그에 의한 asynchronous 전송은 데이터그램의 전송 대기 큐에 저장되어 전송된다.
- Bandwidth(14bit) : isochronous 전송에 사용할 자신의 대역폭을 나타낸다. Bandwidth는 BANDWIDTH_AVAILABLE 레지스터에 해당하는 값을 소켓을 생성할 때 명시하여 사용한다. IP 계층은 실시간 데이터의 주기적 전송을 위하여 상위 네트워크 계층으로부터 대역폭을 할당 받고 연결을 맺는다. 이 대역폭 정보는 M-IP의 헤더로 네트워크로 전달된다. 주기적 데이터는 ASP로 전송을 하지만 수신자로부터 ITM 플래그가 없으면 ACK를 받으면, M-TCP의 전송 제어 메커니즘에 의해 ISM로 전송을 시작한다. 이때 M-IP의 대역폭 정보에 의해 isochronous의 자원이 할당된다. 다량의 비실시간 데이터의 전송은 ASP 만으로 전송이 이루어지기 때문에 대역폭을 명시

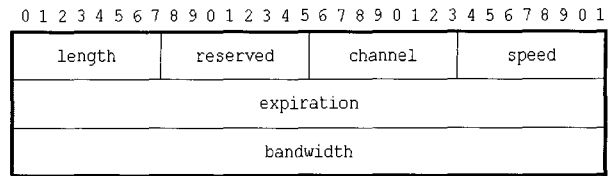


그림 2. Isochronous listen 요청 패킷.
Fig. 2. Isochronous listen request packet.

하지 않아도 된다.

2. Socket IP 옵션의 수정

M-IP의 전송 방법을 선택하기 위해 소켓 정보에 전송 방법과 대역폭을 정의해야 한다. 소켓 옵션 변수는 setsockopt() 함수로 M-IP 소켓의 설정을 한다. Iso_alloc은 실시간 데이터를 전송하는 소켓임을 나타내고, channel은 할당 받은 isochronous 채널을 나타낸다. Transfer_method는 IEEE 1394의 전송 방법을 나타낸다. 실시간 데이터는 transfer_method의 bandwidth를 자신의 대역폭으로 설정하고, 실시간 데이터는 iso를 '1'로 설정하고, 주기적 데이터는 asp를 '1'로 설정하여 전송 방법을 설정한다.

3. 로컬 대역폭 관리자

IEEE-1394에서 isochronous 전송은 IRM에 의해 대역폭과 채널을 할당 받아 전송을 하게 된다. 할당 되지 않은 패킷은 IRM에서 예외 처리를 발생 시키므로 IRM에 의한 버스 관리가 이루어 지지 않는다. 로컬 대역폭 관리자(LBM)는 내부 IP 계층에서 요청한 Isochronous 필드의 대역폭을 가지고 IRM에게 대역폭과 채널을 할당 받는다. 채널의 할당은 IRM의 CHANNEL_AVAILABLE 레지스터에서 할당 되지 않은채널을 LOCK 요청 패킷을 전송하여 이루어 진다. 채널이 할당되면 IRM의 BANDWIDTH_AVAILABLE 레지스터에 Isochronous 필드의 대역폭을 가지고 LOCK 요청 패킷을 전송한다. 할당 받은 정보를 저장하고, 송신자에게 할당 받은 채널을 listen할 것을 요청하는 패킷(그림 2)을 전송한다. 또한, 사용이 끝난 소켓의 대역폭과 채널 자원은 IRM에게 반환 한다. 먼저 송신자와의 연결을 닫고 할당된 대역폭과 채널을 IRM에게 LOCK 요청으로 동기 전송 자원을 반환한다.

IV. 실시간 제어용 TCP

TCP 계층은 IP의 상위 계층으로, 연결을 맺고 데이터를 전송하는 connection-oriented 프로토콜이다. 신뢰성 있는 데이터 전송과 byte-stream-oriented 프로토콜로 연결의 시작과 종료, 데이터 흐름 제어를 중요한 목적으로 갖는다. TCP는 신뢰성 있는 데이터의 전송을 위해서 'Basic Data Transfer', 'Reliability', 'Flow Control', 'Multiplexing', 'Connections' 과 같은 서비스를 제공한다. 데이터의 흐름 제어를 위한 'window'는 'Sliding window'방식을 사용한다[11,12]. 이 방식을 사용하기 위한 네트워크 상태를 나타내는 두 가지 중요한 변수는 cwnd (congestion window)와 ssthresh (slow-start threshold)이다[14]. Slow-start 상태에서의 연결 확립 후, cwnd는 지수적으로 증가한다. 이 상태는 cwnd가 ssthresh를 초과하면 'congestion avoidance' 상태로 전환된다. 'Congestion avoidance' 상태는 각 RTT (Round Trip Time)가 1일 때까지 증가한다. cwnd의 증가는

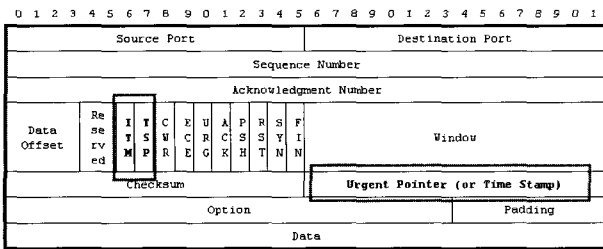


그림 3. M-TCP 헤더.

Fig. 3. M-TCP Header.

'loss'와 'timeout'을 야기 시키는데, 이때 ssthresh가 반이 되며, cwnd는 1이 된다. 이러한 방법으로, rate의 증감이 결정된다. 이런 방식은 'bulk data'와 같은 양이 많을 데이터를 전송할 때 문제점을 야기 시킨다. 그것은 하나의 노드가 전체 대역폭을 독점하게 된다는 것을 의미 한다. 실시간 제어 시스템의 통신은 TCP의 신뢰성 있는 데이터 전송과 데이터의 전송 시간 조건을 만족해야 한다. M-TCP에서는 이러한 시간 조건을 만족하게 하기 위한 시간 제어 메커니즘과 이를 확인하기 위한 방법을 갖추고 있다.

1. 수정된 TCP 헤더

M-TCP는 데이터 세그먼트에 시간 정보를 포함시켜 전송하게 된다. 각 세그먼트는 송신될 때 현재 시간 값에 deadline 값을 더해진 시간 정보를 갖고, 수신자는 세그먼트를 수신했을 때의 현재 시간 값을 이와 비교하게 된다. 세그먼트의 시간 정보 값이 현재 시간보다 이전이면 데이터는 시간 조건을 만족하지 못 했고, 현재 시간 이후이면 데이터는 시간 조건을 만족한 것이 된다. 송신자에 의해 전송되는 ACK 패킷에는 시간 조건을 만족했는지에 대한 정보를 포함시킨다. 이로써 수신자는 전송 제어를 할 수 있다. 그림 3에서 M-TCP 헤더를 보여주고 있다. 세그먼트에 시간 정보를 기록하는 경우 TSP (time stamp) 플래그와 Time_Stamp 값으로 알려 준다. 송신자는 시간의 제약이 필요한 데이터를 전송할 때 TSP 플래그를 설정하고, 그에 따른 deadline 시간 값을 Time_Stamp에 설정하여 송신한다. 수신자가 ACK를 전송할 때 받은 데이터가 deadline 시간을 지났는지를 ITM (in time) 플래그로 알려 준다. TSP 전송으로 시간 조건을 만족해야 하는 데이터임을 나타내고, ITM으로 전송이 지연에 대한 정보를 얻을 수 있다.

(1) Time_stamp 필드

Time_Stamp 필드는 TCP 헤더의 Urgent_Pointer 필드를 대체하여 사용한다. Urgent_Pointer는 URG 플래그에 의해 유효한 값을 가지므로 URG 플래그를 사용하지 않는 경우 사용하지 않는 필드이다. 또한, TSP 플래그와 Time_Stamp를 사용하는 모든 데이터는 Urgent 데이터에 해당하기 때문에 모두 URG 플래그를 사용해야 하는데, 이를 TSP 플래그로 대신할 수 있다.

시간 정보가 필요하지 않는 데이터는 TSP 플래그를 클리어하고 Time_Stamp 값은 기존의 Urgent_Pointer로 하여 전송한다. Time_stamp 값은 IEEE-1394의 iso-chronous_cycle_time값을 기준으로 사용한다. CYCLE_TIME 레지스터는 second_count 7bit, cycle_count 13bit, 그리고 cycle_offset 12bit를 갖고 있다. 이 레지스터는 CM이거나 isochronous 패킷을 수신하거나

송신할 수 있는 링크 계층에 의해 1/24.576us마다 cycle_synch_event를 발생시켜 증가한다. Cycle_offset 값이 3071 (125ms)가 되면, cycle_count값이 증가한다. 이때 CM은 CS 패킷을 전송하고, 모든 노드의 CYCLE_TIME을 동기 시킨다. Second_count는 cycle_count가 7999 (1s)가 되면 증가한다. Time_stamp는 CYCLE_TIME의 상위 16bit를 사용한다. Time_stamp의 cycle_count는 0-499로 2ms의 분해능을 갖고, second_count는 0-127까지 시간을 갖게 된다. 또한 이 Time_Stamp는 CM에 의해 자동으로 동기 되기 때문에 동기화를 위한 별도의 패킷이 필요하지 않는다.

(2) Socket TCP 옵션의 수정

M-TCP의 실시간 전송을 보장하기 위해 소켓 정보에 실시간 전송 소켓임을 정의해야 한다. Real_time 변수는 실시간 데이터 전송을 나타낸다. In_time은 ACK패킷에 의해 전송되었던 데이터가 deadline 시간이 지나지 않고 재 시간에 도착했음을 나타낸다. Deadline 변수는 데이터가 생성된 후 유효 시간을 Time_Stamp 구조에 맞게 설정 한다.

2. M-TCP 데이터 흐름 제어

M-TCP는 TCP의 흐름 제어 메커니즘을 그대로 사용하지 않, 실시간 데이터를 전송하는 경우 시간 제어 메커니즘을 제공한다. 시간 제어 메커니즘은 TSP와 ITM에 의한 정보를 필요로 하게 된다. 송신측에서는 TSP 플래그와 Time_Stamp, 수신측의 ITM 플래그는 전송 지연에 대한 정보를 송신측에 전달하게 되는데, 이때 ITM 플래그가 없는 ACK 패킷은 데이터의 전송이 지연 되었음을 나타낸다.

M-TCP의 새로운 제어 메커니즘에 의해 세 가지의 TCP 상태가 추가 되었다. 그림 4에 M-TCP의 상태 전이와 같이 BM_ALLOC_1과 BM_ALLOC_2의 상태는 isochronous 자원을 할당 받는 상태이고, BM_DEALLOC의 상태는 자원을 반환하는 상태 이다. 실시간 데이터의 전송은 먼저 BM_ALLOC_1 상태에서 대역폭을 할당 받는다. 그리고, 연결을 맺음으로써 실시간 전송을 보장 받는다. 주기적 데이터의 전송은 ESTABLISHED 상태에서 지연 전송이 발생하면 BM_ALLOC_2 상태로 전이되어 대역폭을 할당 받은 후 다시 ESTABLISHED 상태로 전이하게 된다.

(1) 실시간 데이터 흐름제어

실시간 데이터는 소켓을 생성한 후 소켓 옵션으로 real_time과 deadline 설정으로 이루어진다. TCP 상태는 BM_ALLOC_1으로 전이된다. 이때 하위 계층에서는 IRM으로부터 isochronous 자원의 할당을 받게 된다. 자원 할당이 이루어진 다음 할당된 채널을 connection을 맺을 때 전송하게 된다 (그림 5-a). 수신자는 이 채널 정보로 isochronous 채널을 listen하게 된다. 할당된 채널이 준비가 되면 송신자는 TSP 플래그와 SYN 플래그로 서로의 연결을 맺게 된다. 이후의 데이터 전송은 isochronous 전송으로 실시간성을 보장 받는다.

(2) 주기적 데이터 흐름제어

주기적 데이터는 소켓을 생성한 후 소켓 옵션으로 real_time과 deadline 을 설정한다. 소켓 설정을 후 TCP와 같이 연결을 맺는다(그림 5-b). 이때까지 모든 패킷은 asynchronous로 전송된다. 데이터 전송을 시작한 뒤 ITM 플래그가 없는 ACK를 받으면, 먼저 전송할 패킷이 queuing 되

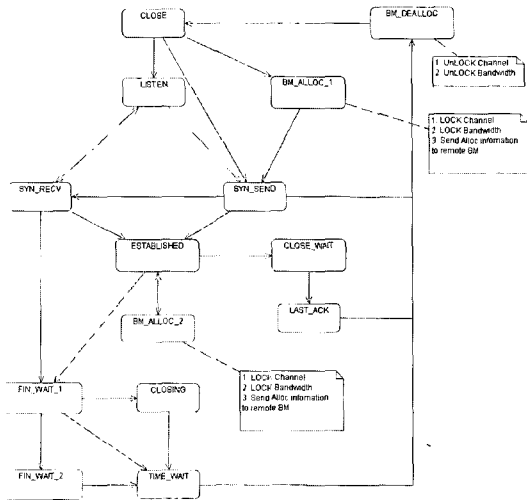


그림 4. M-TCP 상태 전이도.
Fig 4. M-TCP State Transition Diagram.

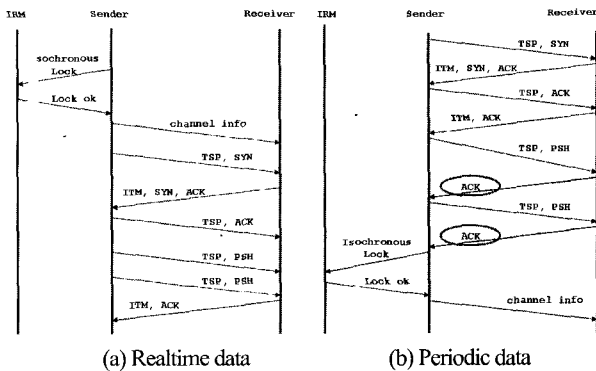


그림 5. 데이터 흐름 제어.
Fig 5. data flow control.

지 않게 하여 보다 빠른 전송을 시도 한다. 네트워크 트래픽이 증가하여 다시 ITM 플래그가 없는 ACK를 받으면 IRM에 sochronous 자원의 할당을 요청한다. 자원 할당이 이뤄진 다음은 실시간 데이터와 같이 isochronous 전송으로 데이터를 전송하게 되므로 실시간성을 보장 받는다.

V. 성능 분석

본 논문에서 제안하는 프로토콜의 성능분석을 위하여 M-TCP/IP를 구현한 리눅스 커널, Pentium-II 이상의 PC, IEEE 139-a의 S400을 만족하는 TI TSB12V26을 사용하는 PCI 카드, 그리고 이더넷 100Mbit 카드의 장비 조건에서 시험하였다. 성능 분석은 네트워크 프로토콜 독립적인 성능 평가 프로그램인 Netpipe[18]와 네트워크 프로토콜 아널라이저 프로그램인 thereal[19]을 사용하였다. 두 프로그램 모두 M-TCP/IP를 지원하지 않으므로 M-TCP/IP를 지원하도록 약간의 소스를 수정하여 사용하였다.

1. 비주기적 데이터 전송

그림 6(a)는 이더넷과 IEEE 1394의 MTU에 따른 성능 분석 결과이다. 이더넷은 MTU가 1500이고, IEEE 1394는 S400일

때 2040 (2048-헤더크기)이지만, 각 1500, 2028, 2040과 2044로 전송할 때 성능비교 그래프이며, 6(b)는 두 장비에서 4개의 다량의 비실시간 데이터의 전송 (asynchronous 전송)을 사용할 때 성능을 비교한 결과이다. 그림 6(c)는 세 장비에서 4개의 다량의 비실시간 데이터의 전송과 1개의 실시간 데이터의 전송(isochronous 전송)의 성능을 비교한 결과이다. 이들 그래프를 보면 IEEE-1394 기반의 데이터전송이 이더넷에 비하여 월등히 안정된 전송특성을 가지고 있음을 알 수 있으며, 비실시간 데이터와 주기적 전송이 혼합된 경우에도 주기적 전송이 정확하게 이루어지고 있음을 알 수 있다.

표 1. 이더넷 데이터 전송 실험 결과.

Table 1. Experiment results over Ethernet.

	평균전송속도	TCP 패킷 수	TCP 바이트 수
Data 1	22.710	2,602	2,616
Data 2	27.409	3,140	3,157
Data 3	45.269	5,198	5,221
전체	95.479	10,948	10,997

표 2. M-TCP/IP IEEE 1394 데이터 전송.

Table 2. Experiment results over M-TCP/IP and IEEE-1394.

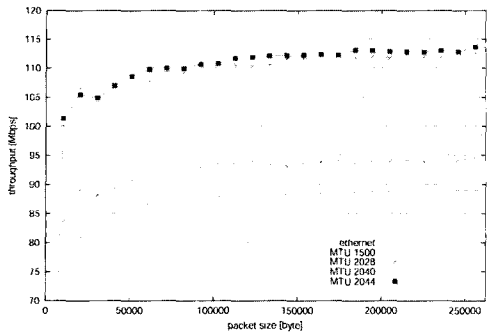
	평균전송속도	TCP 패킷 수	TCP 바이트 수
Data 1	71.204	6,470	8,731
Data 2	72.008	7,547	10,183
Data 3	82.490	8,586	11,659
Data 4	83.193	13,583	18,442
Realtime	54.063	4,591	6,606
전체	362.958	40,777	55,621

2. 실시간 주기적 데이터 전송

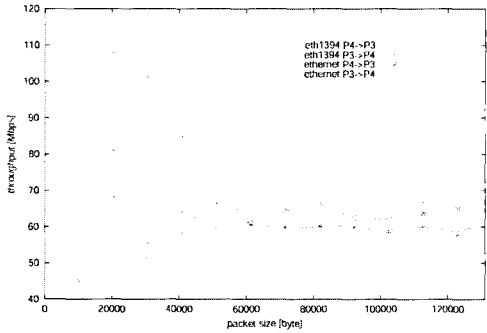
표 1은 100Mbps TCP/IP 이더넷 환경에서의 데이터 전송이다. Data1, Data2, Data3는 데이터로 200ms 주기를 가진 데이터 2048kbyte를 전송하여 얻은 결과이다. 각각 평균 전송속도, 초당 전송된 TCP 패킷 수, 초당 TCP 데이터 바이트 수이다. 전송 시간의 측정은 클라이언트가 데이터를 송신할 때 시간을 측정하고, 서버가 전체 데이터를 받았을 때, 데이터 내에 포함되어 있는 시간을 전송하여 이를 받은 클라이언트가 현재 시간과 비교함으로써 측정하였다.

TCP/IP 이더넷의 경우, 데이터 전송 초기의 혼잡 제어에 의해 전송 지연이 크게 변화함을 볼 수 있는데 이것은 이더넷에서 발생하는 데이터 충돌에 의해 발생되며, 전송이 계속 되면 어느 정도 안정되지만 이더넷의 특징상 크게 개선되지 않는다. 그림 7(a)에서와 같이 전송 지연 시간이 전송 지연 시간이 폭 넓게 위치하게 된다.

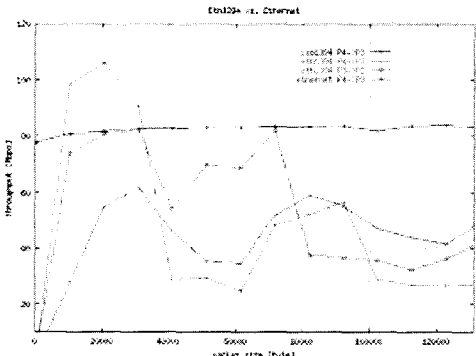
표 2는 M-TCP/IP IEEE 1394 환경에서의 데이터이다. Data1, Data2, Data3, Data4는 전송 주기 100ms를 갖고, 2048 Kbytes를 전송하며, 실시간 데이터를 100ms의 주기를 갖고, 512 Kbytes를 전송한다. 반면에 본 논문에서 제안한 M-TCP/IP IEEE



(a) Performance of Ethernet and 1394



(b) 1394/M-TCP for bulky data



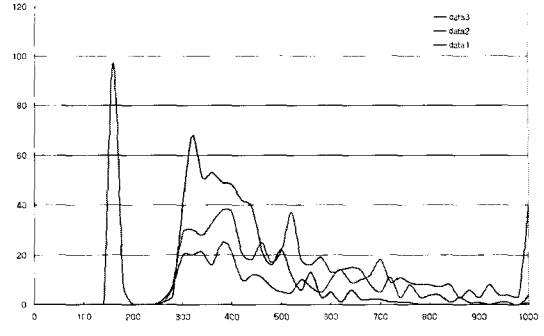
(c) Isochronous transmission

그림 6. 이더넷과 Eth1394의 성능 비교.
Fig. 6. Ethernet and Eth1394 comparison.

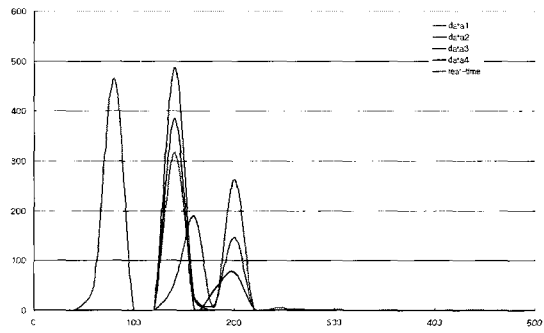
1394의 경우, 그림 7(b)를 볼 때 이더넷과는 달리 전송 지연 시간이 넓게 퍼지지 않고 집중됨을 볼 수 있다. 실시간 데이터(Deadline 100ms)의 전송의 경우 최소 전송 지연 시간은 79.252 ms, 최대 전송 지연 시간은 83.509ms, 평균 전송 지연 시간은 81.034ms이며, 표준 편차는 0.808이다.

VI. 결론

본 논문은 분산 제어 시스템에서의 실시간성을 보장하는 제어 통신 방법을 IEEE 1394에서의 M-TCP/IP로 제시하였다. 제안된 방법을 분산 제어 시스템에 적용할 경우 IEEE 1394의 특징적인 전송 방법으로 실시간 전송을 보장 할 수 있다. 또한, 이더넷과 달리 IEEE 1394는 높은 대역폭을 제공하므로, 제어 시스템을 위해 양적으로 충분한 데이터의 전송이 가능해진다. 아울러 M-TCP/IP는 TCP/IP를 기반으로 하기 때문에



(a) Ethernet



(b) M-TCP/IP IEEE 1394

그림 7. 지연시간 분포도.
Fig. 7. Distribution of latency over IEEE 1394.

기존의 호환성을 유지하므로, TCP/IP를 사용하는 모든 제어 시스템에서 최소의 수정으로 실시간 데이터의 전송을 보장 받을 수 있게 되었다.

본 논문의 M-TCP/IP는 IEEE 1394의 서로 다른 두 가지 전송 방법을 적용하여 데이터의 우선순위에 따른 전송 서비스를 제공한다. 또한, 전송 제어에서는 실시간 전송 방법을 적용하여 데이터의 시간에 따른 신뢰성을 높였다. 이는 분산 시스템에서 제어 통신을 위한 확실한 데이터 전송을 보장한다.

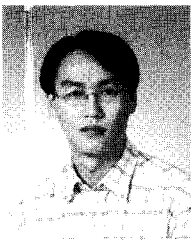
참고문헌

- [1] Information Sciences Institute, "Transmission control protocol", RFC 793, 1981.
- [2] Information Sciences Institute, "Internet protocol", RFC 791, 1981.
- [3] J. Postel, "User datagram protocol", RFC 768, 1980.
- [4] A.V.T.W. Group, "RTP: a transport protocol for real-time applications", RFC 1889, 1996.
- [5] R. Braden, Ed, "Resource reservation protocol", RFC 2205, 1997.
- [6] P. Johansson, "IPv4 over IEEE 1394", RFC 2734, 1999.
- [7] S. Deering and R. Hinden, "Internet protocol, version 6 (IPv6) specification", RFC 1883, 1995.
- [8] IEEE Standard 1394-1995, "IEEE standard for a high performance serial bus".
- [9] IEEE Standard 1394a-2000, "IEEE standard for a high performance serial bus-amendment".
- [10] IEEE Standard 1212, "Information technology - microprocessor

systems - control and status registers (CSR) architecture for microcomputer buses”.

[1] W. R. Stevens, “TCP/IP illustrated volume 1”, 1994.
 [1] G. R. Wright and W. R. Stevens, “TCP/IP illustrated volume 2”, 1995.
 [1.] J. h. Park, Y. C. Yoon, “An extended TCP/IP protocol for real-time local area networks”, IFAC Control Engineering Practice, vol. 6, no. 1, pp. 111-118, 1998.
 [1-] V. Jacobson, “Congestion avoidance and control”. ACM SIGCOMM '88, August 1988.

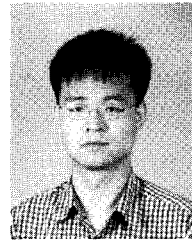
[15] Linux kernel source 2.4.19 - <http://www.kernel.org/>, May 2001.
 [16] IEEE 1394 for Linux - <http://www.linux1394.org/>, Oct 2002.
 [17] 1394 Trade Association - <http://www.1394ta.org/>, Oct 2002.
 [18] Network Protocol Independent Performance Evaluator 2.4 - <http://www.scl.ameslab.gov/netpipe/>, Dec 1999.
 [19] Network Protocol Analyzer 0.9.3 - <http://www.ethereal.com/>, Mar 2001.



염복진

1975년 12월 11일생, 2001년 인하대학교 자원공학과 졸업 (공학사). 2003년 인하대학교 대학원 자동화공학과 졸업 (공학석사). 2003년~현재 현대중공업 기계전기연구소 연구원. 관심분야는 IEEE-1394, 내장형 시스템, 실시간 통신, 임베

디드 리눅스.



윤기중

1975년 1월 3일생. 2002년 인하대학교 자동화공학과 졸업 (공학사). 현재 인하대학교 정보통신대학원 재학. 관심 분야는 IEEE-1394, 내장형 시스템, 임베디드 리눅스, 실시간 OS.



박재현

1963년 10월 8일생. 1986년 서울대 제어계측공학과(공학사). 1988년 동 대학원(공학석사). 1994년 동 대학원(공학박사). 1995년~현재 인하대 정보통신공학부, 부교수. 관심분야는 내장형시스템, 실시간 시스템, 컴퓨터 네트워크, 이산현상 시

스].