論文 96-21-9-10

고속망에서의 멀티캐스트를 위한 고속 수송 프로토콜(XTP)의 구현 및 성능 평가

正會員 이 경 호*, 이 완 직*, 이 선 우*, 김 철 우**,
김 정 삼***, 장 성 식****, 한 기 준*

Implementation and Performance Evaluation of the XTP(XpressTransport Protocol) for Multicasting in High-Speed Networks

Kyeong Ho Lee*, Wan Jik Lee*, Sun Woo Lee*, Cheol Woo Kim**,
Jeong Sham Kim***, Seong Sik Chang****, Ki Jun Han* *Regular Members*

요 약

본 논문에서는 고속 통신망에서의 멀티캐스트를 위한 고속 수송 프로토콜 XTP(Xpress Transport Protocol) Rev 4.0을 Windows NT 상에 구현하고 그 성능을 평가하였다. 프로토콜은 이벤트(Event) 구동 방식으로 설계되었으며, 성능향상을 위해 커널 내부에 네트워크 드라이버의 형태로 구현하였다. 구현된 프로토콜의 기능은 LAN 환경하에서 Windows NT 응용 프로그램들로 테스트되었으며, TCP와 성능비교를 수행하였다.

ABSTRACT

This paper describes implementation and performance evaluation of XTP(Xpress Transport Protocol) on the Windows NT for multicasting in high-speed communication networks. We designed the protocol by an event-driven method and implemented it in form of network driver in a kernel for performace enhancement. Various applications program are used for its functional test and comparison with the TCP protocol.

I. 서 론

JTC1/SC6, ITU-T/SG7및 IETF와 같은 국제 표준화 기구에서는 ECFF(Enhanced Communication Functions and Facilities) 작업을 통해 새로운 네트워크및 트랜스포트 계층의 프로토콜을 개발하기 위한 작업을 진행중이다. 이러한 작업과 관련된 노력으로 1990년대 초 미국 PROTOCOL ENGINE사는 고속 통신

*경북대학교 컴퓨터공학과
**한국통신 통신망 연구소 연구원
***경북전문대 전자계산과
****김천전문대 전산정보처리과
論文番號:96173-0613
接受日字:1996年 6月 13日

프로토콜로서 XTP를 설계하였고, 현재 XTP-FORUM을 중심으로 계속적인 수정, 보완이 이루어져 최근 XTP Revision4.0이 제안되어 있는 상태이다. XTP는 고속망 환경에 적합한 구조를 가지는 통신 프로토콜로서 트랜스포트 계층에서 멀티캐스트를 수행하는 대표적인 프로토콜이다[1].

본 논문에서는 XTP 프로토콜 Rev 4.0을 Windows NT상에서 구현하고 그 성능을 평가하였다. 프로토콜은 이벤트 구동 방식으로 설계하였으며, 성능향상을 위해 Windows NT 커널 내부에 네트워크 드라이버의 형태로 구현하였다. XTP는 프로토콜의 융통성과 서로 다른 통신 환경에의 적응성을 위해 프로토콜 정책과 그 메카니즘을 분리하여 메카니즘은 구현자에게 맡기고 있다. 본 논문에서는 XTP 프로토콜의 기능중 특히 멀티캐스트 메카니즘과 그룹 멤버쉽 관리에 초점을 맞추어 구현을 하였다. 본 연구에서는 프로토콜 내부에서 데이타 복사를 통하여 1:n 멀티캐스트 기능을 수행하도록 하였으며, 또한 멀티캐스트 연결중 그룹 멤버의 참여, 탈퇴등의 그룹 멤버쉽 기능을 수행할 수 있도록 하였다. 그리고 프로토콜의 성능을 평가하기 위하여 TCP 프로토콜과 성능비교를 수행하였다.

Ⅱ. XTP 프로토콜 구현

2.1 XTP 개요 및 관련연구

고속 망으로의 망 환경의 변화로 인해 TCP와 같은 기존의 프로토콜을 보완하고자 많은 연구가 이루어지고 있는데 이는 크게 두 가지 방향으로 이루어 지고 있다[2]. 그중 하나는 프로토콜을 VLSI, High-Speed Adapter Boards, Transputer와 같은 하드웨어로 구현함으로써 성능을 향상시키고자 하는 방향인데 현재 TCP 프로토콜도 하드웨어로 구현하려는 연구가 진행중이다. 또 다른 방향은 고속망 환경에 적합한 프로토콜 구조를 갖는 새로운 고속 전송 프로토콜을 개발하는 것이다. OSI/TP4, VMTP, XTP, NETBLT등의 프로토콜이 그러한 예이다[3]. 이중 XTP는 현재까지도 XTP Forum에 의해 계속적인 보완이 되고 있는 상태인데 현재 XTP forum은 JTC1/SC6의 ECFF작업에 적극적으로 참여하면서 현재 제안되어 있는 XTP Rev 4.0을 DAVIC및 Space Communication에서 사용될 프로토콜 후보로 제안하고 있는 단계이다[2]. XTP 외에 트랜스포트 계층에서 멀티캐스트를 지원하는 프로토콜로는 VMTP, MTP등이 있다[3].

XTP4.0에서는 하위 계층에서 멀티캐스트를 지원하는 경우 이를 이용한다. 본 구현에서는 XTP4.0의 Ethernet Encapsulation 방법을 따르며, 하위 계층이 멀티캐스트를 수행할 수 없는 경우에도 멀티캐스트를 수행할 수 있도록 프로토콜 자체적으로 멀티캐스트를 수행하도록 하였다.

2.2 프로토콜 구현구조

본 논문에서 설계한 XTP 프로토콜의 구현 구조를 그림 1에 나타내었다. XTP 프로토콜은 크게 상위 사용자의 요청(Primitive)을 받아들이거나 프로토콜 내의 데이타를 상위계층으로 전달하는 등의 역할을 하는 User Interface 모듈, 하부 네트워크 카드 드라이버와의 접속을 담당하는 Network Interface 모듈, 독립적인 스레드로 이벤트 큐의 우선순위에 따라 이벤트를 디스패치하여 Action 모듈을 호출하는 XTP Thread 모듈, 그리고 실제적인 프로토콜의 동작을 수행하는 함수들을 포함하는 ACTION 모듈로 구성된다. XTP 내부의 각 모듈에서 발생한 이벤트는 이벤트큐에 삽입되고 XTP 스레드는 정의된 우선순위에 따라 이벤트 큐에서 이벤트를 가져와 그 타입에 따라 처리해야 할 ACTION 모듈을 호출한다.

그림 1에 나타나 있는 Device Extension은 디바이

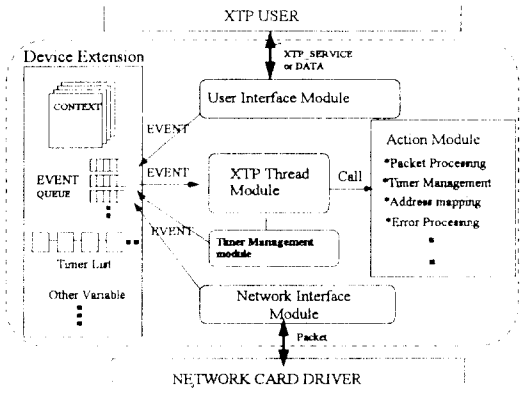

그림 1. XTP 프로토콜의 구현 구조.
Fig. 1. Implementation structure of XTP protocol.

스의 상태와 시스템에서 정의한 오브젝트, 그리고 드라이버의 작성자가 정의한 데이타 구조를 포함하는 드라이버의 전역 변수(global variable) 데이타 영역이다.

2.3 프로토콜 구현을 위한 주요 자료 구조

2.3.1 CONTEXT

본 구현에서 XTP는 그 연결이 유니캐스트연결이든 멀티캐스트 연결이든 상관없이 하나의 연결을 관리하기 위해 Context라는 자료 구조를 유지한다. 특히 여기에는 멀티캐스트 그룹의 그룹멤버에 관한 정보를 유지하는 RCV_MEMBER라는 자료구조가 포함된다.

2.3.2 서비스 프리미티브

본 논문에서는 XTP 프로토콜 서비스 프리미티브를 그림 2와 같은 XTP_SERVICE라는 구조체 형태로 정의하였다. XTP는 이 자료구조를 통해 멀티캐스트, 그룹관리, 트래픽 정보를 프로토콜과 교환한다.

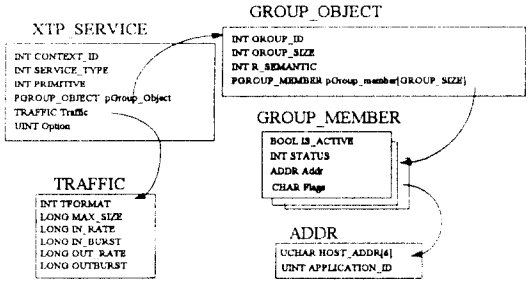

그림 2. XTP_SERVICE 구조체.
Fig. 2. XTP_SERVICE STRUCTURE.

2.3.3 RECEIVER_MEMBER

그림 3은 멀티캐스트 송신자가 멀티캐스트 수신 그룹을 관리하기 위해 Context내에 유지하는 자료구조로서 수신 그룹의 현 상태를 유지한다. 송신자는 RCV_MEMBER라는 이 구조체내의 정보를 토대로 에러, 흐름, 전송률제어를 수행하게 된다.

RCV_MEMBER 구조체 배열의 각 원소는 멀티캐스트그룹의 수신멤버 개개에 관한 정보를 포함한다. 그래서 XTP는 그룹의 수신자가 증가하더라도 RCV_MEMBER의 원소만을 추가하는 것만으로 수신자

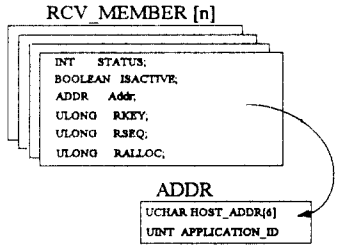

그림 3. RCV_MEMBER 구조체.
Fig. 3. RCV_MEMBER STRUCTURE.

정보관리가 가능하다.

2.4 프로토콜 주요 기능 구현

2.4.1 데이타 전송

그림 4는 멀티캐스트 데이타 전송과정을 나타내고 있다.

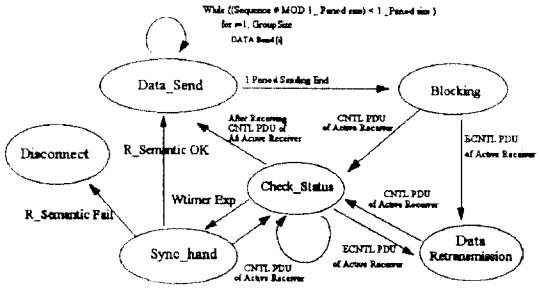

그림 4. 데이타 전송 상태 천이도.
Fig. 4. Transitoin Diagram of Data Transmission.

본 구현에서는 그림4에서 보는 바와 같이 1주기의 데이타 전송이 이루어진 후 수신 멤버의 ACK를 요구하는데, 수신 멤버가 응답하지 않을 경우 Synchronize-Handshake 통해서 수신멤버의 탈퇴여부를 조사한다. 그 결과 Reliablity_Semantic를 위반한 경우 멀티캐스트 연결은 해제된다.

2.4.2 에러제어 및 흐름제어

본 구현에서는 GO-BACK-N방법을 사용하여 데이타 재전송을 수행한다. 그리고, 흐름 제어는 Sliding Windows 방식을 사용하였다.

2.4.3 전송률제어

XTP는 기존의 트랜스포트 프로토콜과는 달리 PDU의 전송속도를 제어하는 전송률 제어 기능을 가지고 있는데, 본 구현에서는 XTP프로토콜 내부에서 RTIMER 값을 동적으로 결정할 수 있도록 했다.

2.4.4 그룹 멤버쉽 관리 기능

XTP는 멀티캐스트 기능과 함께 멀티캐스트 그룹을 관리하기 위한 기능을 지원한다. 일반적으로 그룹통신의 경우 그룹 멤버의 참여, 탈퇴, 그룹 신뢰성 보장이 이루어져야 하는데 본 구현에서는 다음과 같은 방법으로 이를 지원한다.

1. 그룹 멤버 참여:수신을 원하는 수신자는 XTP_JOIN 프리미티브를 사용하여 XTP에게 참여 요구를 통보하고 XTP가 송신측에게 JOIN 패킷을 전송하면, 송신측 XTP는 XTP_JOIN 프리미티브를 통해 유저에게 통보한다. 그후, 유저가 참여 요구를 받아들일 경우 송신측 XTP는 현재의 RCV_MEMBER구조체 배열에 그 수신자에 관한 정보를 포함시켜 현재의 수신자 그룹에 참여시킨다.

2. 그룹 멤버 탈퇴:수신자가 그룹에서 탈퇴하는 경우는 voluntary exit, forced exit, silent exit의 3가지 형태로 이루어진다. 먼저, 송신자가 수신멤버를 강제적으로 탈퇴시키는 경우, 유저는 XTP에게 XTP_DELETE 프리미티브를 사용하여 수신자 강제 탈퇴를 통보하고, XTP는 RCV_MEMBER자료구조에서 해당 수신멤버를 삭제하고, 그 수신 멤버에게 End필드를 세트시킨 제어 패킷을 전송함으로써 그 수신 멤버에 대한 데이타 전송을 중지한다. 다음으로, 수신자의 자발적인 탈퇴의 경우, 프로토콜의 Check_Status상태에서 이를 감지하고 Synchronize-Hcandshake를 통해 수신자 탈퇴를 확인후 RCV_MEMBER 자료구조를 갱신, XTP_LEAVE 프리미티브를 통해 유저에게 통보한다. 이 때도 해당 수신멤버를 RCV_MEMBER 자료구조에서 삭제한다. 그리고, 이 순간에 ALIVE상태에 있는 수신자의 갯수가 Reliability Semantic를 위배할 경우 멀티캐스트 연결은 해제된다.

3. 그룹 신뢰성 보장:XTP 멀티캐스트 그룹의 신뢰성은 ACTIVE 수신 그룹에 대해 K-Reliablity Semantic를 보장함으로써 이루어진다. 그리고, 이러한 그룹 멤버의 참여, 탈퇴등은 그림4의 데이타 전송 상태 천이도에서 check_status상태에서 이루어진다. 즉, 한 주기 동안의 데이타가 전송된 후 수신멤버는 그룹에 참여하거나 송신자는 탈퇴한 수신 멤버에게로의 데이타 전송을 중지할 수 있다.

표 1. XTP 멀티캐스트 그룹 관리 기능
Table 1. Group Management of XTP Multicast Group

프리미티브	방 향	파라미터
XTP_JOIN	XTP유저→XTP(수신측)	송신측 주소, Traffic Parameter
	XTP→XTP유저(송신측)	수신측 주소
	XTP유저→XTP(송신측)	ACCEPT/REJECT flag, ACTIVE flag
	XTP→XTP유저(수신측)	ACCEPT/REJECT flag
	XTP유저→XTP(송신측)	수신멤버ID
XTP_DELETE	XTP→XTP유저(수신측)	
XTP_LEAVE	XTP→XTP유저(송신측)	수신 멤버ID

Ⅲ. 프로토콜 성능 분석

3.1 성능 분석 방법

본 논문에서는 먼저, 프로토콜 자체의 성능 분석을 위해 하위 네트워크 드라이버의 성능을 측정하였으며, 그리고 이벤트 별 처리속도를 측정하여 내부모듈의 오버헤드를 측정하였다. 다음으로, 두 대의 Windows NT시스템사이에서 XTP를 억세스하여 유니캐스트및 멀티캐스트로 화일을 전송하는 프로그램을 작성하여 XTP 프로토콜의 성능을 측정하였으며 또, TCP프로토콜의 성능 측정을 위해 SOCKET 프로그래밍으로 같은 화일전송 프로그램을 작성하였다. 여기에서 사용하는 TCP 프로토콜은 MicroSoft사가 제공하는 Windows NT 3.1용 TCP 네트워크 디바이스 드라이버이다.

3.2 성능 분석 결과

3.2.1 네트웍 드라이버 성능

구현된 프로토콜의 성능에 많은 영향을 주는 것은 하부 네트웍 드라이버의 성능이다. 그래서 본 논문에

서는 이러한 네트웍 드라이버 인터페이스 함수의 수행 지연 시간을 측정하였다. 이 것은 네트워크 드라이버 성능을 간접적으로나마 측정함으로써 구현된 프로토콜의 전체 처리율을 네트워크 카드 성능에 대한 비율로 나타내기 위해서이다. 측정은 그림 5와 같이 이루어 졌으며 그 결과 표 2와 같은 결과를 얻을 수 있었다. 이는 XTP 프로토콜이 하나의 패킷을 전송 및 수신하기 위해 네트워크 드라이버가 제공하는 서비스를 수행하는데 소요되는 시간이다.

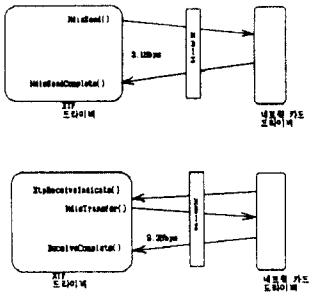

그림 5. 네트워크 드라이버의 성능
Fig. 5. Performance of Network Driver

표 2. 네트워크 드라이버의 성능
Table 2. Performance of Network Driver

구 분	지연(μs)	처리율(Mbps)
송신지연	3896	3.1
PDU Indication	41	290
PDU Transfer	3669	3.3
수신지연	3710	3.2

3.2.2 프로토콜 자체 성능 분석

본 구현의 XTP프로토콜은 이벤트 구동 방식으로 구현 되어 있기 때문에 먼저 이벤트 중심으로 프로토콜 자체의 성능을 분석하였다. 표3은 구현된 각 이벤트 처리 프로시져의 평균 수행 속도와 그 이벤트를 처리하는 오버헤드가 전체 프로토콜 동작중에서 차지하는 비율을 나타내고 있다. 이 결과는 이벤트 큐에서 이벤트를 획득하는 시점과 처리를 완료하는 시점사이의 지연시간을 측정함으로써 얻은 것이다.

이 결과에서 알 수 있듯이 데이타 송신의 경우는 데이타를 PDU에 실어 송신하는 과정, 수신의 경우는 수신된 데이타를 유저에게 전달하는 과정이 프로토콜의 성능에서 큰 부분을 차지한다는 것을 알 수 있다. 그런데 이 두 과정에서는 공통적으로 데이타의 메모리 복사가 이루어지는 데 이는 프로토콜의 동작 중 메모리 복사가 프로토콜의 성능에 지배적으로 영향을 미침을 말하는 것이다.

표 3. 이벤트 별 지연 및 오버헤드
Table 3. Delay and Overhead of Event processing

이벤트	지연시간(μs)	오버헤드(%)
FIRST Send	162	2.8
FIRST Receive Event	1317	21.7
DATA Send Event	1204	19.8
DATA Receive Event	1146	18.9
CNTL/ECNTL/TCNTL Send Event	315	5.1
CNTL/ECNTL/TCNTL Receive Event	48	0.1
DATA Delivery Event	1828	30.1
Timer Event	81	1.3

3.2.3 유니캐스트

다음 그림 6은 3Mbyte의 화일을 전송했을 경우 XTP와 TCP 프로토콜의 성능을 응용계층에서 유저 버퍼의 크기를 달리 함으로써 측정한 결과이다. 측정은 XTP 연결(혹은 TCP)을 설립한 후, 화일 전송을 위해 첫번째 WriteFile시스템 콜(TCP의 경우 Send)을 한 시점에서 마지막 WriteFile시스템 콜에 대한 응답

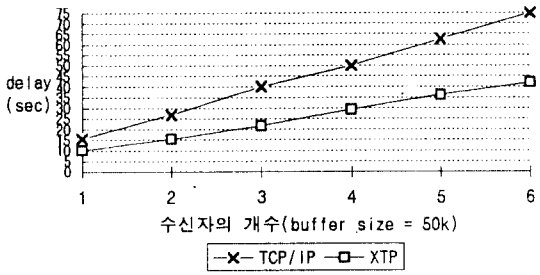

그림 6. 유저 버퍼의 크기에 따른 성능.
Fig. 6. Measurement of Performance with varing user buffer size.

을 받았을 때 까지의 지연 시간을 측정함으로써 이루어 졌다.

유저 버퍼의 크기를 50Kbyte으로 했을 때 TCP프로토콜은 약 1.5Mbps의 성능을 보이는 반면 XTP 프로토콜은 약 2.2Mbps의 성능을 보임을 알 수 있다. 그리고 유저버퍼의 크기를 측정할 수 없는 FTP를 사용하여 같은 크기의 화일을 전송할 경우에는 1.382 Mbps의 성능을 얻을 수 있었다.

위의 그래프에서 유저 버퍼의 크기가 작을 경우는 프로토콜의 처리능력에 비해 상대적으로 적은 양의 데이타를 전송할 경우로 데이타 전송외 다른 프로토콜의 동작과 상위 계층과의 동작에 너 많은 오버헤드가 발생하므로 성능은 떨어진다. 그러나 유저버의 크기를 늘려 프로토콜에게 요구하는 데이타의 양이 증가하고 유저와의 인터액션이 줄수록 프로토콜의 성능은 향상된다. 그러나 프로토콜은 하부네트워크 드라이버의 성능에 제한을 받기 때문에 2.5Mbps이상은 증가하지 않는다. 이 결과는 3.2.1절에서 측정된 하부네트워크 드라이버 성능이 3.1~3.2Mbps을 고려하면 약 71%정도의 성능을 보여 줌을 알 수 있다.

3.2.4 멀티캐스트

XTP는 트랜스포트 계층에서 멀티캐스트 동작이 이루어지기 때문에 성능면에서 우수하다. 본 구현에서 TCP는 멀티캐스트를 지원하지 않기 때문에 여러개의 유니캐스트 연결을 열어서 멀티캐스트를 시뮬레이션하였다. 다음 그림 7은 XTP와 TCP의 멀티캐스트의 성능을 수신자의 갯수를 증가시킴으로써 측정한 결과이다. 측정방법은 앞에서와 같은 방법을 사용하였으며 도표에서 세로축은 3Mbyte의 화일 전송시 소요되는 시간을 나타낸다.

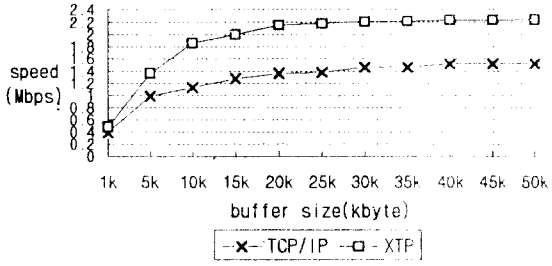

그림 7. 수신자 수의 증가에 따른 Delay증가.

Fig. 7. Measurement of dealy with varing the number of receiver.

그림 7에서 보면 XTP가 TCP보다 멀티캐스트시 수신자의 갯수가 증가할 수록 성능면에서 더 좋은 결과를 보여 줌을 나타낸다. TCP는 수신자의 갯수가 증가함에 따라 거의 유니캐스트시 지연의 배수만큼 지연이 증가한다. 이는 수신자의 갯수가 하나 증가함에 따라 연결 자체가 추가되기 때문이다. 그러나, XTP는 수송계층에서 멀티캐스트를 하기 때문에 지연의 증가는 있으나 그 증가율은 평균 유니캐스트 지연의 68%에 머문다. 이러한 증가는 앞서 언급한 제어 패킷의 처리와 메모리 복사 증가외 수신멤버들에게로의 동기적인 데이타 전송등이 그 요인이다. 프로토콜의 성능향상을 위해서는 이 부분에 대한 개선이 필요한 것으로 고려된다. 그러나 이러한 증가율은 TCP의 경우보다 작은 것으로 수송계층에서 멀티캐스트를 함으로써 얻는 이점중의 하나가 된다.

Ⅳ. 결 론

본 논문에서는 XTP 프로토콜을 Windows NT 커널내에 네트워크 디바이스 드라이버 형태로 구현하고 Ethernet LAN 환경에서 그 성능을 평가하였다. 특히 프로토콜 내부에서 멀티캐스트를 수행하기 위한 메카니즘과 멀티캐스트 그룹관리를 위한 그룹 멤버쉽 관리 기능에 중점을 두어 이를 위한 프로토콜내의 자료구조와 주요 프로토콜 기능을 설계하고 구현하였다. 본 논문에서 구현된 XTP는 멀티캐스트 성능평가를 위해 TCP 프로토콜과 성능비교를 수행하였으며 그 결과 보다 우수한 성능을 보임을 알 수 있었다.

참 고 문 헌

1. XTP Forum, "Xpress Transport Protocol 4.0 Specification," March 1, 1995.
2. M. Zitterbart, "High-speed transport components," IEEE Network Mag., vol. 5, pp. 54-63, Jan. 1991.
3. Willibald A. Doesringer, Doug Dykeman, Matthias Kaiserswerth, etc. "A Survey of Light-Weight Transport Protocols for High-Speed Networks," IEEE Transaction on Communication. Vol. 38,

No. 11, November 1990.

4. A. N. Nettravali, W. D. Roomf, K. Sabnani "Design and Implementation of a High Speed Transport Protocol," IEEE Trans, On Communication, Vol. 38, No. 11, Nov 1990.
5. David C. Feldmeier, "A Framework of Architectual Concepts for High-Speed Communication Systems," IEEE Journal On Selected Areas In Communications, VOL. 11, NO. 4, MAY 1993.
6. T. F La Porta, Architecture, feature, and Implementation of high speed transport protocols, IEEE Network Mag., May 1991.
7. Changpeng Fan,Thomas Luckenbach, Xiangwen Xu Perfomance Comparison and Analysis of XTP and TCP/IP over the BERKOM Broadband ISDN Network, IEEE, 1993.

이 경 호(Kyeong Ho Lee) 정회원
1995년 2월:경북대학교 컴퓨터공학과 졸업(학사)
1995년 3월~현재:경북대학교 컴퓨터공학과 석사과정중
※주관심분야:고속 통신 프로토콜, ATM, PCS, 멀티미디어 통신등

이 완 직(Wan Jik Lee) 정회원
1992년 2월:경북대학교 통계학과 졸업(학사)
1994년 2월:경북대학교 컴퓨터공학과 졸업(석사)
1994년 3월~현재:경북대학교 컴퓨터공학과 박사과정 재학중
※주관심분야:고속 통신 프로토콜, 그룹웨어 지원 프로토콜, ATM등

이 선 우(Sun Woo Lee) 정회원
1995년 2월:경북대학교 컴퓨터공학과 졸업(학사)
1995년 3월~현재:경북대학교 컴퓨터공학과 석사과정중
※주관심분야:고속 수송 프로토콜, ATM, 멀티미디어 통신등

김 철 우(Cheol Woo Kim) 정회원
1994년 2월:영남대학교 전산공학과(학사)
1996년 2월:경북대학교 컴퓨터공학과(석사)
1996년 3월~현재:한국통신 통신망 연구소 연구원
※주관심분야:통신망 분석, ATM, 통신 프로토콜등

김 정 삼(Jeong Sham Kim) 정회원
1987년 2월:경북대학교 전자공학과 졸업(학사)
1990년 2월:경북대학교 전자계산기공학과(석사)
1990년 3월~1995년 4월:국방과학연구소 연구원
1995년 4월~현재:경북전문대 전자계산과 전임강사
※주관심분야:PCS, LAN프로토콜, 멀티미디어등

장 성 식(Seong Sik Chang)정회원
1987년 2월:경북대학교 전자공학과 졸업(학사)
1989년 2월:경북대학교 전자공학과 졸업(석사)
1994년 3월~현재:김천전문대 전산정보처리과 전임강사
※주관심분야:멀티미디어 데이터베이스, ATM, 통신 프로토콜등

한 기 준(Ki Jun Han) 정회원
1979년:서울대학교 전기공학과 졸업(학사)
1981년:한국과학기술원 전기공학과 졸업(석사)
1985년:Univerity of Arizona 전기 및 전산공학과 졸업(석사)
1987년:Univerity of Arizona 전기 및 전산공학과 졸업(박사)
1981년~1984년:국방과학연구소 연구원
1988년~현재:경북대학교 컴퓨터공학과 부교수
※주관심분야:전산망 프로토콜, B-ISDN, MAN/LAN, 분산처리등