
IEEE1394 기반의 홈 네트워킹에서의 TCP 성능 분석

장중욱*

Analysis of the TCP performance over IEEE1394 based Home Networkings

Jong-Wook Jang*

요약

TCP는 특정 네트워크 기술의 에러 특성이나 대역폭에 상관없이 사용자들에게 종단간의 적절한 성능을 제공한다. 그러므로 TCP의 강점은 인터넷 환경에서 유용하다는 것이다. TCP/IP 프로토콜을 사용한 고속 홈 네트워킹에 관련된 어플리케이션이 계속 소개되고 있는 추세이다. 이로 인해 인터넷과 홈 네트워킹을 통합한 환경이 요구된다. 본 논문에서는 IEEE1394를 기반으로 하는 고속 홈 네트워킹 환경에서의 TCP모델을 적용하고, IEEE1394 기반의 홈 네트워킹의 이용효율을 평가해 보았다. 시뮬레이션을 통해 IEEE1394 기반의 홈 네트워킹에서의 TCP의 성능 평가에 있어서 다음과 같은 결과를 알 수 있었다. 비동시성의 트랜잭션에서 TCP는 매우 훌륭한 성능을 발휘하였고 한편으로는, 동시성 트랜잭션의 TCP성능은 낮게 나타났다. 동시성과 비동시성의 혼합 트랜잭션에서의 TCP 성능은 작은 크기의TCP 패킷이 사용될 때 보다 나은 처리량을 보였다.

ABSTRACT

TCP typically offers reasonable end-to-end performance to users regardless of the bandwidth and error characteristics of particular network technology. The robustness of TCP has contributed to its success in the internet environment. The role of communications is already well established in the office environment. With the advent of cheap, affordable broadband communications and the increasing complexity of consumer goods, it seems natural to extend the network into homes. In-home networking means a high-speed communication among the digital appliances within a home. Introduction of application over high-speed home network using TCP/IP protocol is increasing. The integrated environment of internet and home network is demanding as well. We have validated TCP model over high speed home network environment, investigated the throughput behavior of TCP over IEEE 1394 home networks, and evaluated a potential solution for high performance of TCP over IEEE 1394 home networks. The simulation model has produced several interesting results in the performance of TCP over IEEE 1394 home network.

IEEE1394, Home Networking, TCP, Analysis

* 동의대학교 컴퓨터공학과
접수일자: 2001. 7. 27

I. 서론

TCP는 가고자 하는 곳은 어디든지 갈 수 있는 능력을 가졌기 때문에 주로 4구분 자동차로 표현된다. TCP는 일반적인 목적지향 프로토콜이기 때문에 LAN, MAN, WAN과 같은 어느 환경에서나 작동 가능하다. 더군다나, TCP는 물리계층에서의 에러 특성 및 대역폭과 같은 네트워크 기술의 특성에도 불구하고 이러한 환경에서 사용자에게 신뢰성 있는 종단 서비스를 제공한다. 이 TCP의 강점은 인터넷 환경에서 유용하다는 것인데, TCP는 파일전송, 원격접속, 전자메일, 네트워크 뉴스, WWW와 같은 상위계층 어플리케이션에서 요구되는 실시간 정보 전달을 어디에서건 제공해 준다.

데이터 통신 규약은 이더넷 LAN과 같은 네트워크 기술을 통해 이미 사무실 환경에서는 보편화되어 있고, 광대역 통신과 한층 복잡해지는 가전제품의 등장으로 태내에 네트워크 구성은 자연스러운 현상이 되었다. 또한 MPEG와 DSP기술의 등장은 가전제품의 디지털화와 고속통신을 가능하게 하고 있다. 이것은 홈 네트워킹의 등장을 촉진하였다.

최근 몇몇 투자 업체들은 음악, VCR, Video 업체들이 그들의 표준화 기술로 채택한 IEEE 1394 제품이 새로운 바람을 일으키길 기대하고 있다. 또한, 차세대 IP체제는 확장된 주소 체계를 제공하고, 그것은 태내 분산된 더 많은 네트워크 장비들을 연결하기 위한 WAN을 수용한다. 이러한 동향은 고속 홈 네트워킹을 위한 일반적인 구조 형태로 IEEE 1394 Fire Wire 표준이 채택될 것이다. IEEE 1394 표준명세서는 FireWire로 알려진 직렬 버스 구조이다. 애플 컴퓨터로부터 시작된 FireWire는 국제적으로 채택된 ISO/ICE13213 (IEEE1212) 표준명세서에 기초하고 있다[13]. 원래 컴퓨터와 주변 기기간의 내부 연결을 위한 기존의 버스(ex, SCSI)를 대체하기 위한 것이었던 FireWire 버스는 3.2Gbps의 속도로 동작하고, Hot-plugged기능을 지원한다. 융통성 있는 구조의 peer-to-peer 토폴로지 형태를 취하는 IEEE1394는 프린터와 하드 드라이브에서 디지털 오디오 및 비디오까지 실시간으로 요구를 처리하고 연결해 준다. 초고속 IEEE1394 직렬 버스는 컴퓨터와 정보가전제품을 위한

디지털 데이터 전송에 변혁을 가져올 것이다. 디지털 장비간에 고속, 저 비용과 같은 특징들을 제공함으로써 IEEE1394는 일반적인I/O 커넥션으로 자리할 것이다. 융통성 있는 구조의 peer-to-peer 토폴로지를 통해 컴퓨터 및 하드 드라이브에서 디지털 오디오 및 비디오를 연결하기 위한 디바이스로 생각해낸 것이 IEEE1394이다. 정확한 시간의 동시적 전달은 멀티미디어 인터페이스를 저 비용으로 실행 가능하게 한다.

TCP/IP 프로토콜을 사용하는 초고속 홈 네트워킹 상에서의 어플리케이션은 증가하고 있다. 마찬가지로, 인터넷과 홈 네트워킹의 통합된 환경도 요구되고 있으므로 홈네트워킹 환경에서 TCP프로토콜을 사용하는 환경은 당연히 될 것이다.

따라서 본 논문에서는 초고속 홈 네트워킹상에서 고속 응용을 지원하는 IEEE1394환경에서 인터넷 응용 작업과 TCP를 이용한 홈네트워킹 응용 프로그램이 잘 처리되도록 하기 위한 TCP모델을 설정하여 TCP의 처리동작을 연구하였고, 실제 초고속 홈 네트워킹 상에서의 TCP 성능을 시뮬레이션을 통해 평가하였다. 성능 문제는 TCP와IEEE1394간의 상이한 패킷간의 매치, 프래그멘테이션 (Fragmentation), TCP의 혼잡 제어 기법, IEEE1394 에러 기법과 같은 여러 사실들을 이용하였다. 제 2장에서는 TCP프로토콜 및 매커니즘 그리고, IEEE1394 네트워크에서 어떻게 동작하는가에 필요한 백그라운드를 제공하고, 제 3장에서는 시뮬레이션 환경에서의 TCP모델의 수행과 설계를 나타내고 있다. 그리고 제4장에서는 시뮬레이션 설정을 설명하였고, 제 5장에서는 TCP 모델의 시뮬레이션과 IEEE1394상에서의 TCP가 갖는 몇가지 문제들을 확인하고, 끝으로 결론을 다루었다.

II. TCP over IEEE 1394 Protocol

이 장에서는 IEEE1394 네트워크 상에서의 TCP전송 과정에서 발생하는 문제점과 프로토콜에 관한 몇 가지 백그라운드 정보를 제공하겠다.

2.1 TCP over IEEE 1394 Protocol Stack

TCP와 IP는 일반적인 TCP/IP 프로토콜 단위로 구성된다[1,2,3]. TCP over IEEE1394 프로토콜 스택의

예를 그림 1에서 보여주고 있다. 이 프로토콜의 스택은 end-to-end 프로토콜 형태인 TCP와 어플리케이션 간의 인터페이스이다. 몇 가지 계층의 처리 과정을 거친 후에 데이터는 IEEE 1394를 통해 전송된다. 그림 1의 최하위 계층인 물리계층, 링크계층, 트랜잭션 계층은 IEEE1394 네트워크 상에 데이터 전송을 위해 함께 사용된다.

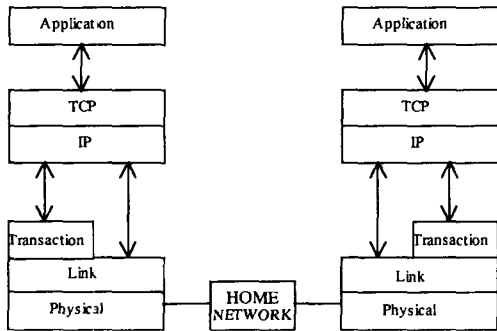


그림 1. IEEE 1394상에서 TCP 프로토콜 스택
Fig.1 TCP Protocol Stack over IEEE1394

IEEE 1394 직렬 버스가 peer-to-peer 트랜잭션을 제공하기 때문에 중재(arbitration)는 직렬 버스의 소유권을 어느 노드가 획득할 것인가를 결정하는 기능을 수행하게 되었다. 이 중재 기법은 모든 전송 자들이 버스에 접근하기 위한 공정한 알고리즘을 제공한다. 직렬 버스는 2가지의 데이터 전송 형태를 가지고 있다. 하나는 일정한 데이터 전송율을 가지고 있지 않은 비동기 전송이다. 비동기 전송은 특정 어드레스를 가진 한 노드가 목적이다. 이러한 전송은 일정한 버스 대역폭을 요구하지 않고, 따라서, 버스 사용 간격도 일정하지 않지만 시간 외 접근을 얻을 수 있어야 한다. 다른 하나는 일정한 간격을 가지고 데이터를 전송하는 동시적 전송이 있다. 이 전송은 특정 어드레스보다는 채널 번호를 설정하여 채널 번호에 응답하는 하나 혹은 여러 개의 노드에 뿌려진 동시적 데이터 스트림을 제공한다. 이러한 전송자들은 일정한 간격으로 버스 접근을 요구하고, 따라서 비동기적 전송자보다 더 높은 우선권을 가진다.

직렬 버스 데이터 트랜잭션은 패킷 전송 정보와 데이터의 배열을 발생시킨다. 각 트랜잭션은 request에 의해 송신되고, request는 responder라 불리는 목적지

장비에 수신된다.

비동기적 트랜잭션은 목적지 노드로부터 응답을 요구하고, 그 결과 추가적인 트랜잭션이 발생한다. 동시적 트랜잭션은 목적지 노드를 인지하기 위해 채널 번호를 사용한다. 추가적으로 목적 노드로부터 되돌아오는 응답은 없다. 데이터는 6가지 속도 중 한가지로 전송될 수 있다 : 100Mbps, 200 Mbps, 400 Mbps, 800 Mbps, 1.6Gbps 및 3.2Gbps

다음 절에서는 그림 1에 제시된 TCP IEEE1394 프로토콜 스택상에서의 몇 가지 모드 세부사항을 설명한다.

2.2 PHY 계층

물리계층은 1394 장치와 1394 케이블 간의 전기적, 기계적인 연결을 제공한다. 데이터 송수신을 활성화시키는 작업 외에 물리계층은 모든 장치가 버스에 공정한 접근을 가질 수 있도록 중재를 제공한다.

2.3 링크 계층

링크 계층은 두가지 형태의 데이터 패킷 전송 서비스를 제공한다 : 비동기 및 동시적 전송. 언급하기 전에 비동기적 전송은 전통적인 전송-응답 형태의 프로토콜이고, 동시적 전송은 정보의 정확한 전달을 위한 실시간 대역 확보가 보장된 프로토콜 방식이다.

비동기 트랜잭션을 위해 링크 계층은 트랜잭션 계층과 물리 계층간에 인터페이스를 제공하고, 트랜잭션 계층에서 사용하는 동일한 요구/응답 모델에 기반을 둔 서비스를 제공한다.

요구자의 링크 계층은 1394 케이블을 통해 수신한 1394 패킷에 있는 트랜잭션 계층으로부터의 트랜잭션 요구를 해석한다. 그리고 나서, 패킷이 수신자에게 수신되었을 때 그것은 번역되어 그것의 트랜잭션 계층에 전달된다.

동시적 트랜잭션을 위해서 링크 계층은 동시적 소프트웨어 드라이버와 물리 계층간에 인터페이스를 제공한다. 전송하는 동안 링크 계층은 케이블을 통해 보내질 동시적 패킷을 생성한다. 링크 계층은 또한 케이블로부터 동시적 패킷을 수신하고, 그 패킷 채널 넘버를 디코드한다. 이 노드를 위한 패킷이 설정되면 그 패킷은 IP 혹은 소프트웨어 드라이버에 전달되어 진다.

2.4 트랜잭션 계층

트랜잭션 계층은 비동기적 프로토콜의 쓰기, 읽기, 잠금 명령을 제공한다. 쓰기 명령은 발신지에서 수신지로 데이터를 보내는 것이고, 읽기 명령은 발신지로 데이터를 불러오는 것이다. 잠금 명령은 수신자의 처리과정을 포함한 송수신자 간의 데이터의 a round trip routing을 생산함으로써 쓰기와 읽기 명령의 기능을 합친 것이다.

비동기적 트랜잭션 모델은 요구자와 응답자 노드간의 통신에 기반을 두고 있다. 각 트랜잭션은 요구 동작과 응답 동작으로 구성되어 있고, 요구자와 응답자 트랜잭션 계층간에는 링크와 물리계층의 동작을 포함하고 있다.

IEEE1394 어플리케이션들은 주로 IEEE1394 통신 모델의 그 중간 계층에서 발생하는 동작들에 대해서 관여하지 않는다. 다시 말하면, 그들은 트랜잭션 계층에 데이터 전송 요구를 알린다. 이 소프트웨어 계층은 전송자들을 비교하여 하나 혹은 더 많은 전송 요구들 속에서 더 필요한 전송 요구를 번역 전달한다. 언어낸 트랜잭션 요구는 트랜잭션 타입(읽기, 쓰기, 잠금)을 식별하고, 트랜잭션이 쓰기 혹은 잠금으로 구성되었다면, 트랜잭션 계층은 또한 요구동작 동안 전송되는 데이터를 전달해주는 기능을 한다.

2.5 IP 계층

IP 프로토콜은 네트워크 계층 연결이 없는 프로토콜이다[4]. 이것은 데이터그램의 신뢰성 있는 전달을 보장한다. IP는 최상의 서비스를 제공한다. 일시적으로 라우터 동작 이상과 같은 오류가 발생하였을 때 IP는 간단한 예외 교정 알고리즘을 가지고 있다. 이것은 데이터그램을 폐기하거나, 다시 보내보거나 ICMP메시지를 소스에게 되돌려준다. 상위계층에도 신뢰성이 제공되어야 한다. IP는 또한 프래그멘테이션을 제공하는데 이것은 네트워크 인터페이스를 통해 전송될 수 있도록 TCP 세그먼트를 분할하는 것이다. 이 프래그멘테이션 처리과정은 오버헤드를 발생시키므로 TCP는 모든 세그먼트가 MTU보다 더 작은 길이를 갖도록 한다.

2.6 TCP 계층

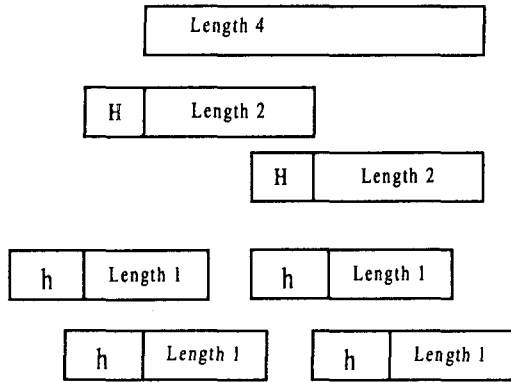
TCP는 Connection-Oriented를 제공하기 위해 설계

된 전송계층 프로토콜로써 실질적으로, 양단간에 byte 스트림 서비스를 제공한다[3]. TCP는 IP를 신뢰성이 없는 네트워크를 통해 응용개체 데이터를 전송하는데 사용한다.

TCP에서의 신뢰성이 있는 데이터 전송은 여러 가지 기술로 실행된다. 응용 데이터는 TCP가 송신하기에 적당한 크기로 무엇을 고려하느냐에 따라 분할된다. TCP가 IP에 전달하는 정보 단위를 세그먼트라 한다. TCP가 세그먼트를 보내고 타이머가 그것을 체크하는 동안 다른 종단이 세그먼트를 수신했음을 알리기를 기다리고 있다. 정해진 시간 안에 응답이 없으면 세그먼트는 재전송된다. TCP는 그것의 헤더에 있는 checksum과 데이터를 관리하고, 전송 중에 있는 데이터의 어떤 변화를 감지하는 것이다. TCP 세그먼트가 IP 데이터그램으로써 전송되고, IP 데이터그램이 순서 없이 전송될 수 있기 때문에, TCP 세그먼트도 순서 없이 도착할 수 있다. 가능한 한, 수신측 TCP는 도착 순으로 수신된 데이터를 재배열한다. IP 데이터그램은 복사될 수도 있기 때문에 수신측 TCP는 복사된 데이터를 폐기해야 된다. 또한, TCP는 흐름 제어도 수행한다. TCP 연결의 각 종단은 제한된 버퍼 공간을 가지고 있다. 수신측 TCP는 다른 종단에서 수신측에서 가진 버퍼 양만큼의 데이터를 보내도록 한다. 이것은 속도가 늦은 호스트가 모든 버퍼를 점령하는 것을 방지한다. TCP는 혼잡 제어도 수행한다. TCP는 네트워크 상황을 평가하고, byte 단위로 윈도우 혼잡을 유지 관리함으로써 조절한다. 송신자가 가질 수 있는 미 해결된 최대 바이트 수는 송신자 윈도우와 혼잡 윈도우보다 적다. 혼잡 윈도우에 사용되는 표준 알고리즘은 Slow-Start라 불린다[5]. 세그먼트를 잃었을 때, 송신측 윈도우는 세그먼트 하나를 준비하고, 각각의 전송된 세그먼트가 성공적으로 응답함으로써 한 세그먼트가 증가된다. 혼잡 윈도우가 그것의 이전 크기에 접근하였을 때 TCP는 더 천천히 혼잡 윈도우를 공개한다. 이것은 혼잡 회피 기법으로 잘 알려져 있다.

더 빠른 재전송 알고리즘은 한 소스가 같은 세그먼트의 응답을 이중으로 수신하였을 때 세그먼트를 재전송한다. 빠른 복구 알고리즘은 재전송 뒤에 slow-start 성능을 회피한다. 다수의 enhancement 는 또한 고속으로, 더 긴 거리의 네트워크와 더 큰 윈도우를 송

신하기 위한 TCP에 의해 처리되는 것이다[6].



(H : TCP/IP Packet header, h : 1394 Packet header, length n : Packet length rate.)

그림 2. IEEE 1394상에서 TCP를 위한 응용 데이터의 분리
Fig. 2 Segmentation of Application Date for TCP over IEEE1394

2.7 응용과 소켓 인터페이스

4.2 BSD와 그 뒤 버전에는, 응용개체는 네트워크 소켓을 통해 TCP 프로토콜로 통신한다. 소켓 인터페이스는 네트워크 연결, 데이터 버퍼링, 데이터 스트림의 송수신과 같은 기능을 가지고 있다. 응용개체에서 트랜잭션 계층까지의 사용자 데이터 처리 프로토콜의 개념도를 그림 2에서 보여주고 있다. 응용개체에서의 데이터는 소켓 인터페이스를 통해 TCP 계층에 기록된다. TCP와 IP는 TCP/IP 세그먼트에서 데이터를 분할하고, 각각은 데이터의 MSS 길이를 따른다.

III. TCP IEEE 1394 시뮬레이션

TCP 모델은 IEEE 1394 네트워크상에서 양 호스트 간의 데이터 전송을 시뮬레이션하기 위해 설계 되었다. 데이터 전송은 한쪽 혹은 양쪽에서 발생할 수 있다. 최초의 송신자는 source로 참조되고 다른 호스트는 sink로 참조된다. 시뮬레이션의 전체 길이를 위한 단일 연결을 간주하는 모델이기 때문에 TCP connection handshaking은 제공되지 않는다. 시뮬레이션에서 사용하는 옵션과 파라미터는 TCP와 IEEE1394의 전

체적인 구성을 따른다.

이 모델은 참조4.4 BSD-Lite로부터의 TCP/IP 네트워킹 코드를 따른다. 이 수행은 최대의 대역폭과 오랜 지연 경로에 관해 언급하고 있는 참조 4.3 BSD에 기본적으로 기초하고 있다[4].

TCP IEEE 1394 시뮬레이션 모델을 이용하여 양방향 송수신 시스템에 TCP/IEEE1394 프로토콜 스택을 구성하여 CSIM을 이용하여 성능 분석하였다[7]. 전체 프로토콜 스택은 응용개체에서부터 아래 IEEE 1394 계층까지 모델화되어 있다. 그 모델은 네 개의 부분으로 나뉘어져 있다: 응용 계층, 소켓 계층, TCP/IP 프로토콜, IEEE 1394계층

3.1 응용 계층

응용 단계는 소켓 인터페이스를 통해 기록된 시뮬레이터된 데이터를 수행한다. 각 호스트는 시뮬레이션 동안 사용할 데이터 양을 할당한다. 어떤 호스트가 0 byte를 보낸다면 그것은 응답을 보낼 것이고 그 모델은 단방향 트래픽 소스의 기능을 갖는다.

3.2 소켓 계층

소켓 계층은 시뮬레이터된 데이터의 버퍼링과 TCP/IP 계층간의 통신을 수행한다. 소켓 모델은 커널 mbuf 데이터 구조를 시뮬레이터하는 기능을 가지고 있다[8]. 소켓 복사 규칙과 mbufs 는 몇 가지 상황에서의 ATM상에서의 TCP의 낮은 처리량의 원인을 보여준다[9,10,11]. 이 모델은 송수신자의 버퍼 사이즈 및 다른 소켓의 옵션을 명시하기 위한 파라미터들을 가지고 있다.

3.3 TCP/IP 계층

TCP는 개략적으로 모델화되었다. TCP 수행 모델은 가상으로 CSIM과 동일하고 slow start, 재전송, 빠른 복구 등과 같은 동일한 특징을 가지고 있다[12]. 또한 전이중 데이터 통신 방식을 제공한다. IP 계층은 정확하게는 모델화되어 있지 않다. 대신, 모든 TCP 모델은 IP 헤더와 프래그멘테이션을 고려한 세그먼트 크기를 조절한다.

3.4 IEEE 1394 계층

트랜잭션 계층, 링크 계층, 물리 계층이 있다. 트랜잭션 계층은 비동기 트랜잭션에 관련된 읽기, 쓰기, 잠금 기능을 위한 요구/응답 프로토콜을 제공한다. 트랜잭션 계층은 동시성 트랜잭션을 위한 서비스를 제공하지 않는다. 동시성 트랜잭션은 응용개체에 의해 직접 동작된다. 응용개체는 transaction data request 서비스를 통해 비동기 트랜잭션들을 수행한다. 트랜잭션 계층은 패킷 전송을 확인하고 응답 패킷을 제공한다. transaction data request는 링크 계층 서비스 루틴을 호출한다. 링크 계층은 주소, 수신된 비동기 혹은 동시성 패킷을 디코딩하기 위한 채널번호 및 CRC[13] 기능을 제공한다. 원래 물리 계층은 중재를 처리하지만, 시뮬레이션에서는 고려하지 않았다. 중재는 IEEE 1394 표준에 의하면, 좋은 성능평가를 위한 중요한 과정이다.

3.5 기타

모든 소켓, TCP/IP, IEEE 1394 대부분의 특징은 모델 파라미터 혹은 컴파일 옵션을 통해 제어된다는 것이다. 이들은 타이머, MSS, MTU, 전송을 위한 파일 크기, 시뮬레이션 시간, 에러율 등을 포함하고 있다.

IV. 시뮬레이션 설정

본 시뮬레이션은 다양한 구성과 네트워크 설정 하에 다음과 같은 다양한 성능분석을 위하여 TCP 모델을 구성하였다.

- (1) IEEE 1394 홈 네트워크 상에서 TCP의 이점을 모델링하였을 때의 성능을 분석
- (2) 네트워크 혼잡 속에서 IEEE 1394상의 TCP를 통한 두 가지 형태의 전송(비동기 및 동시성)형태의 성능을 분석
- (3) 기존의 TCP문제를 해결하기 위한 여러 가지 방법들을 이 환경에 적용

4.1 실험 설정

그림3은 시뮬레이션을 위해 사용된 네트워크 토폴로지이다. 이 토폴로지는 대표적인 IEEE 1394 홈 네트워크 형태를 나타낸 것이다. 하나 혹은 더 많은 TCP 자원이 point-to-point 링크를 통해 각 요소와 연결되

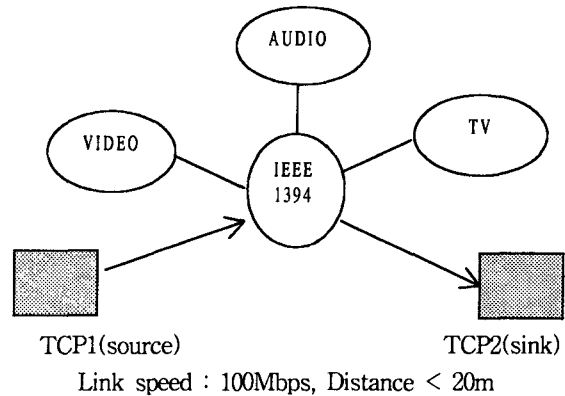


그림 3. 시뮬레이션 네트워크 토폴로지
Fig.3 Network Topology for simulation

어 있다.

TCP 모델은 데이터 전송을 시뮬레이션하는 FTP 프로토콜 사용을 위해 설계되었다. 시뮬레이션은 IEEE 1394 홈 네트워크 기반의 양쪽 호스트간에 수행된다. 네트워크 속도는 100Mbps이고, FTP 데이터 크기는 40Mbytes이고, CSIM시간은 트랜잭션 타입별로 3에서 7초이다. 시뮬레이션 시간은 경험적으로 CSIM에서 네트워크 혼잡 상황에서 성능 평가를 조사하기에 충분하다. TCP패킷 크기는 IP의 MTU(Message Transmit Unit)에 의해 결정된다. IEEE WG는IEEE 1394에서의 IP를 위한 MTU를 1500byte로 제안했다. 일반적으로 512byte의 패킷 사이즈는 IP 네트워크상에서 많이 사용되고 있다. 시뮬레이션에서 사용되는 TCP 패킷 사이즈는 1500, 512, 256byte이다. IEEE 1394 네트워크의 최대 패킷 사이즈는 링크 속도에 따라 512~ 32768byte 이다. 버스 대역의 재배치 사이클이 126s간격이기 때문이다. 시뮬레이션에서 사용되는 IEEE 1394 패킷 사이즈는 512byte이고, 이 크기는 100Mbps 속도에서 비동기 전송을 위한 최대 데이터 속도이다. 그래서, 1500 TCP 패킷은 네 개의 IEEE 1394패킷으로 프래그먼트된다. 512 TCP 패킷은 2개의 IEEE 1394 패킷으로 프래그먼트된다. 256 TCP 패킷은 한 개의 IEEE 1394 패킷이 된다. TCP 윈도우 크기는 64KB, 32KB, 16KB 및 8KB이다. 이때 네트워크 에러율은 10⁻⁶으로 설정하였다. 이것은 홈에서의 전자파 방해를 고려한 것이다. 추가로, 1~10%의 패킷 에러율은 성능 모델에서의 TCP-STACK의 분석 목적을

위한 시뮬레이션에 적용되었다.

V. 시뮬레이션 결과

에러 없는 상태에서의 비동기, 동시성, 그리고 두 가지가 혼합된 전송 형태의 IEEE 1394상에서의 TCP 처리량과 IEEE 1394가 아닌 일반적인 TCP의 처리량을 비교하였다. 일반적으로 TCP프로토콜의 성능을 분석하는데 다음과 같은 네트워크 파라미터를 사용한다. 이것은 TCP 패킷 크기, IEEE 1394 패킷 크기, 네트워크 속도, 윈도우 크기이다.

프래그멘테이션 처리가 오버헤드 처리를 필요로 하는 것은 잘 알려져 있기 때문에 TCP와 IEEE 1394 처리시간은 시뮬레이션에서 고려하지 않았다. 따라서 이 최종적으로는 TCP의 처리량을 측정하고, 네트워크 혼잡 속에서 IEEE 1394상의 TCP의 고성능을 얻는데 초점을 두었다. 동시성 트랜잭션은 몇 가지 성능 문제점을 가지고 있기 때문에 문제점을 개선하기 위한 방법을 찾기 위하여 응답(ACK)을 이용한 방법과 SACK (Selected ACK)을 이용한 방법을 적용하여 평가하였다. 이것은 인터넷에서 손실되는 패킷 수는 증가하고 있고, 같은 윈도우에서 손실되는 멀티플 패킷은 성능 저하의 원인이 되고 있는 환경에서 SACK방법이 성능 개선을 가져다 주는 한 방안이기 때문이다[17].

5.1 비동기 트랜잭션을 위한 TCP over IEEE 1394

비동기 트랜잭션은 CRC와 응답을 통해 데이터 전달을 확인하고, 전송 중에 발생한 에러는 수신후 다시 전송된다. 전송량은 가변적인 비동기 트랜잭션들은 일반적으로 버스 대역의 최소 20%를 사용한다[13].

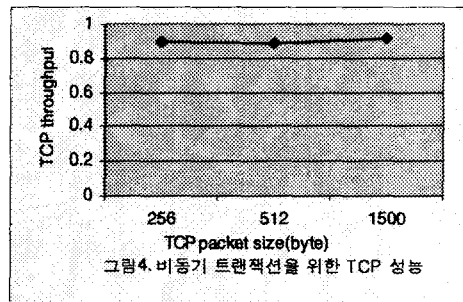
그림 4는 비동기 트랜잭션을 위한 TCP처리량의 TCP MSS의 효과를 보여주고 있다. IEEE 1394의 에러 제어 기능 때문에 세그먼트 3개를 처리하는 처리량과 성능측면에서 매우 유사하다. 처리량의 근소한 차이는 주로 헤더의 오버헤드 때문이다. TCP 혼잡 제어는 이 트랜잭션에서는 아무런 영향이 없다. 왜냐하면 IEEE1394의 제어 기능이 동작되기 때문이다. 그림 4에서 볼 때 비동기 트랜잭션을 위한 효과적인 처리량은 TCP 패킷 크기에 그다지 영향을 미치지 않는 것을

알 수 있다. 또한 TCP 윈도우 크기도 TCP 처리량에 영향을 끼치지 않는다.

5.2 동시성 트랜잭션을 위한 TCP over IEEE 1394

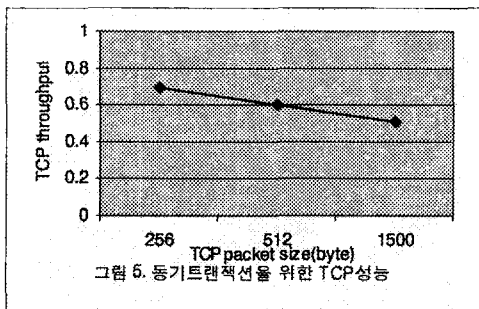
동시성 트랜잭션은 일정한 간격을 가지고 데이터를 전달하므로 수신 노드로부터 돌아오는 응답은 없고, 데이터 전송 요구가 필요하지 않다. 이것은 IEEE1394 네트워크에서 링크의 80% 이상을 할당받는다. 그림 5는 동시성 트랜잭션을 위한 TCP 처리량에서의 TCP MSS의 효과를 보여주고 있다. 낮은 처리량의 원인 중 한가지는 잘 알려져 있는 경우와 같이 패킷 프래그멘테이션이 패킷 재전송 및 대역폭 낭비를 초래하기 때문이다[14]. 이것은 동시성 트랜잭션을 위한 에러제어 기능이 IEEE 1394에는 없기 때문에 TCP 혼잡 제어 회피는 동시성을 위한 TCP의 처리량을 떨어뜨린다. 즉 TCP 혼잡 제어가 지속적으로 증가하는 윈도우 혼잡을 억제하기 때문에 더 큰 TCP 패킷 크기는 링크 전송에서 혼잡으로 인해 낭비되는 IEEE 1394패킷의 수를 증가시킨다.

TCP slow start와 혼잡 회피는 IEEE 1394를 위한 적절한 기법이 아닌 것을 알 수 있다. 그것은 링크가 혼잡을 인지했을지라도 링크는 idle이 가능하기 때문이다.



5.3 비동기와 동시성 트랜잭션의 조합

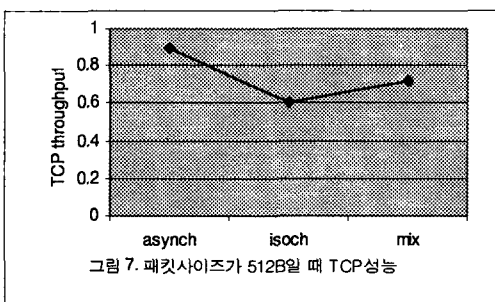
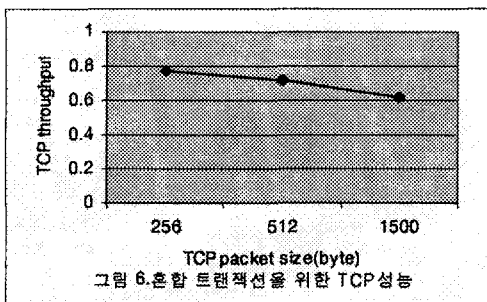
그림 6은 두 가지 트랜잭션이 섞였을 때의 MSS의 기능으로써의 TCP 처리량을 보여준다. TCP 혼잡제어의 재전송 오버헤드로 인해 더 작은 MSS가 데이터 전송에 있어서 더 나은 효과를 가진다.



5.4 비동기와 등시성 트랜잭션의 비교

그림 7은 비동기 트랜잭션이 등시성보다 더 나은 성능을 가지고 있는 것을 보여주고 있다.

두 트랜잭션간의 큰 차이점은 응답이 돌아오는지의 여부이다. 비동기 트랜잭션에서 IEEE 1394는 복구할 책임이 있다. 이것은 TCP 혼잡 제어를 회피하는 것을 뜻한다.



5.5 등시성 트랜잭션에서의 성능 향상 - 중복 ACK

TCP over ATM을 언급하고 있는 논문[14]에 따르면, 재 전송되는 타이머 와 중복 ACK 시작점은 고속

지 전달 지연(low transfer delay) 환경 상에서 의미 있는 효과를 가지고 있다. 그것들은 고속과 낮은 에러율 때문에 IEEE 1394 홈 네트워크와 매우 유사한 환경을 가지고 있다. 그래서 중복 ACK 시작점을 3에서 1로 줄이는 것은 재전송 타이머상의 신뢰 회피에 의한 성능을 향상시킨다.

중복 ACK의 목적은 다른 쪽 끝이 순서가 틀린 세그먼트를 받았는가를 알게 한다. 만일 3개 혹은 그 이상의 ACK들이 연속적으로 받아진다면, 그것은 세그먼트를 잃어버렸다는 분명한 증거이다[15]. IEEE 1394 네트워크는 어긋난 상태에서 패킷이 떨어질 수 있는 packet switched network와 다르기 때문에, TCP over IEEE 1394 연결은 잃어버린 패킷을 책임지기 위한 3중복 ACK들을 기다리지 않는다. 게다가, IEEE 1394 물리 계층은 한번에 단지 하나의 노드가 버스를 통해 데이터를 전송할 수 있도록 중재 한다 [13]. 그림 8이 보여주는 것과 같이, 새로운 중복 ACK 시작점(Dup-ACK1)은 밀집된 네트워크 안에서 낭비된 패킷들의 숫자를 감소시켰고, 1500 bytes 크기의 TCP 패킷에서 56%의 TCP 성능을 증가시킴을 알 수 있다.

5.6 등시성 Transaction 속의 성능 향상 SACK

현재 TCP 실행에서, 누적되는 ACK 기법은 잘 알려주는 바와 같이 에러검출과 복구를 위해 사용되었다. 특히, 다수의 패킷들이 한 윈도우에 떨어질 때 그것은 current TCP에 심각한 성능 문제를 가지고 있다 [16]. 다수의 패킷들이 떨어지기 때문에 그것은 열악한 성능으로 이끄는 slow-start의 원인이 된다. 선별된 ACK기법은 불필요한 재전송을 회피하는 것과 다수의 손실로부터 회복하기 위해서 TCP를 신속히 허락한다. RFC[17]에서 제안된 것과 같이, 데이터는 3중복 ACK 다음부터 재 전송되어 진다. sack block의 최대치는 4로 제한되어 있고, 만일, Time stamp기능이 사용될 경우 3으로 나타날 수 있다. 이 시뮬레이션에서, 윈도우의 크기는 64K 이고 bursty 에러에 있어서 패킷의 수는 3 또는 4로 선택하였다. Bursty 또는 분리된 에러는 서로 섞여지고, 그것들은 임의의 숫자 생성기에 의해 생성된다. Bursty 에러 속의 패킷 숫자

에 대한 관점에서 1%에서 6%의 몇몇 에러율은 시뮬레이션에 적용하였다.

무엇보다도, 만일 비트 에러율이 없다면, TCP기능을 가지는 SACK와 TCP가 없는 SACK는 처리량에 따른 관점에서 보면 차이점이 없다. TCP-SACK은 재전송되어진 패킷들을 다시 잃어버릴 때 slow-start를 피할 수 없으므로, 에러율이 정말로 크다면 SACK은 성능에 아무런 영향을 주지 못한다[17,18]. 에러율이 1%일 때, TCP-SACK은 30%정도의 패킷 크기에 의존한 처리량을 향상됨을 알 수 있다.

그림 9는 0.2초 동안의 TCP가 있는 SACK와 TCP가 없는 SACK사이의 처리량 비교를 나타낸다. 눈에 띄는 처리량 향상은 TCP-SACK에 나타났고, 그것은 불필요한 재전송과 slow-start를 배제하기 때문이다. 그림 10은 TCP-reno(1%와 6% error율)와 TCP-SACK(1%와 6% error율)사이의 처리량 비교를 보여준다. 6% 에러율의 TCP-SACK 동안, 높은 에러율이 아무런 처리량 향상을 보여주지 못하는 것에 반하여 1% error율의 TCP-SACK은 좋은 성능 향상을 보여준다.

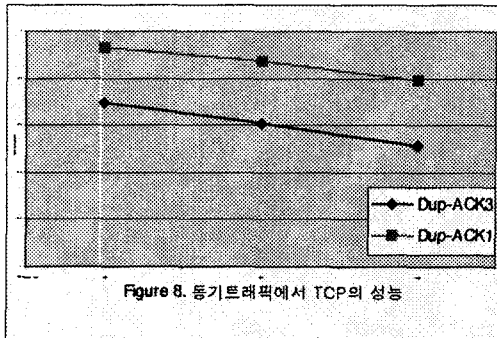


Figure 8. 동시트래픽에서 TCP의 성능

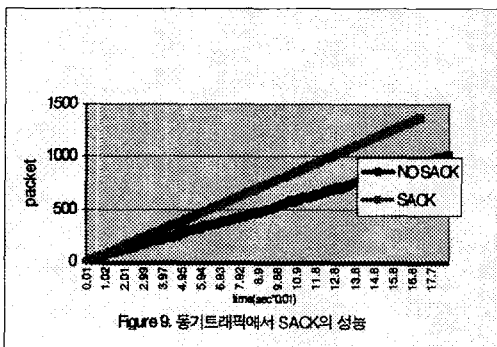


Figure 9. 동시트래픽에서 SACK의 성능

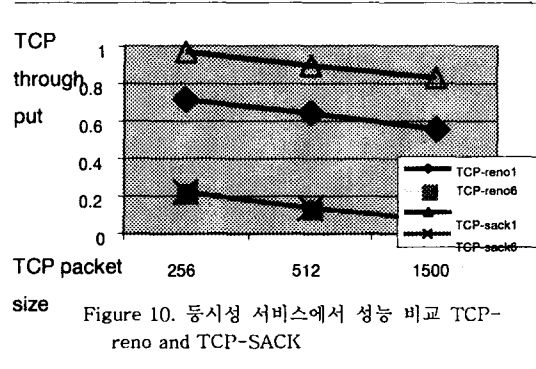


Figure 10. 동시성 서비스에서 성능 비교 TCP-reno and TCP-SACK

VI. 결론

SACK기능을 가지는TCP모델이 고속 홈 네트워크 상에서 고성능으로 수행한다는 것을 확인했고, TCP의 처리량 반응을 조사했으며, 실제 고속 홈 네트워크를 통한 TCP의 높은 성능을 위한 가능성 있는 대안을 제시하였다. 그리고 IEEE 1394 홈 네트워크들을 통한 TCP 프로토콜 적응성을 검사한 결과 다음과 같은 사실을 알 수 있었다.

첫째로, 비동시성의 트랜잭션에서 제안된 모델은 TCP 패킷 크기에 독립적으로 높은 성능을 나타내었다. 한편 동시성 트랜잭션은 낮은 성능을 보였다. 왜냐하면 근본적으로 IEEE 1394와 TCP의 에러 복구 특성 때문이다.

둘째로, 동시성과 비동시성의 혼합 트랜잭션의 성능은 보다 작은 TCP 패킷 크기가 사용될 때에 좋은 처리량을 볼 수 있다. 즉 IEEE1394 데이터 유닛 크기에 접근할수록 성능이 우수하였다.

셋째로, 일정 패킷 크기에서는 비동기 트랜잭션이 최고의 성능을 보임에도 불구하고 동시성 트랜잭션의 성능은 그렇지 못함을 알 수 있었다.

그리고 감소된 중복 ACK 시작점과 TCP-SACK방법이 IEEE 1394 네트워크상의 TCP 성능 향상을 위한 해결책이 될 수 있다는 것은 알 수 있었다. 그러나 3% 이상의 에러율을 가지는 환경이라면 해결방안이 될 수 없음을 알 수 있었다.

참고문헌

- [1] D. Comer, *Internetworking with TCP/IP, Volume 1: Principle, Protocol, and Architecture*, Second edition, Prentice-Hall, Englewood Cliffs, NJ, 1991.
- [2] W. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley, Reading, Massachusetts, 1993.
- [3] G. Wright and W. Stevens, *TCP/IP Illustrated, Volume 2: The Implementation*, Addison-Wesley, Reading, Massachusetts, 1995.
- [4] A. Tanenbaum, *Computer Network*, Second Edition, Prentice-Hall, Englewood Cliffs, NJ, 1988.
- [5] V. Jacobson, "Congestion avoidance and Control", *Proceedings of 1988 ACM SIGCOMM Conference*, Stanford, CA, pp.314-329, August 1988.
- [6] V. Jacobson, R. Braden, and D. Borman, "TCP Extension for High performance", RFC 1323, IETF, 37 pages, May 1992.
- [7] H. Schwetman, *CSIM User's Guide*, MCTCC, Jun 1992.
- [8] S. Leffer, M. McKusick, M Karels, and J. Quarterman, *The Design and Implementation of the 4.3 BSD UNIX Operating System*, Addison-Wesley, Reading, Massachusetts, 1989.
- [9] D. Comer and J.Lin, "TCP buffering and Performance over an ATM network", *Technical Report CSD TR 94-026*, Purdue University, West Lafayette, Indiana, 15 pages, March, 1994.
- [10] J. Crowcroft, I. Wakeman, Z. Wang, and D. Sirovica, "Is Layering Harmful?", *IEEE Network*, Vol.6, No.1, pp.20-24, January 1992.
- [11] K. Moleekdev and P. Gunningberg, "How a Large ATM MTU Causes Deadlocks in TCP Data Transfers", *IEEE/ACM Transactions on Networking*, Vol.3, No.4, pp.409-422, August 1995.
- [12] L. Zhang, S. Shenker, and D. Clark, "Observations on the Dynamics of a Congestion Control Algorithm: The Effects of Two-way Traffic", *Proceedings of the 1991 ACM SIGCOMM Conference*, Zurich, Switzerland, pp.133-147, September 1991.
- [13] D. Anderson, *FireWire System Architecture*, Second Edition, Addison Wesley, 1999
- [14] A. Romanow, S.Floyd, "Dynamics of TCP traffic over ATM networks", *IEEE JSAC*, V.13, N.4, pp.633-641, May 1995.
- [15] W. Richard Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithm", RFC2001, IETF, January 1997.
- [16] K. Fall, S. Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP", *Computer Communication Review*, V.26, N.3, pp.5-21, July 1996.
- [17] M. Mathis, J. Mahdavi, PSC, S. Floyd, LBNL, A. Romanow, SUN Microsystems, "TCP Selective Acknowledgment Options", RFC2018, IETF, October 1996.
- [18] R. Bruyeron, B. Hemon, L. Zhang, "Experimentations with TCP Selective Acknowledgment", *CCR* V.28, N.2, April 1998.

장종욱(Jong-Wook Jang)



1987.2~1995.1 ETRI
 1995.2 부산대학교 컴퓨터공학과 박사 취득
 1995.3-현재 동의대학교 컴퓨터공학과 부교수
 1999.1-2000.2 UMKC 포스트닥

관심분야: ATM-POM, 이동 MAC 프로토콜