

실시간 통신을 위한 확장된 TCP/IP 프로토콜 개발에 관한 연구

A study on the extended TCP/IP protocol for real-time communication

尹泳燦*, 朴宰賢*
(Young-Chan Yoon* and Jae-Hyun Park*)

요 약

본 논문에서는 이더넷을 기반으로 연결되어 있는 분산 soft real-time 시스템에 적용될 수 있는, LAN/TCP라 명명된 확장된 TCP/IP 프로토콜을 제안하고 있다. LAN/TCP는 기존의 TCP/IP 환경에서 아무런 충돌 없이 호환성을 유지하면서 soft real-time 특성을 제공하기 때문에 기존의 응용 프로그램을 별 다른 수정 없이 사용할 수 있다. LAN/TCP는 주기적으로 수집, 갱신되는 제어 신호를 상대적으로 낮은 트래픽으로 전송할 수 있는 PTM(Periodic Transmission Mode) 전송 방식을 제공한다. 본 논문은 제안된 프로토콜의 시뮬레이션 및 실험 결과를 제시하고 있다.

Abstract

This paper proposes an extended TCP/IP protocol, LAN/TCP, that can be used for the distributed soft real-time systems connected through the Ethernet-based local area network. Since LAN/TCP shows soft real-time performance with keeping compatibility and interoperability with the standard TCP/IP, the existing application software can be used without any modification. LAN/TCP also provides the periodic transmission mode (PTM) with which the periodical data collection and updating the control signals can be efficiently implemented with relatively small traffic overhead. This paper includes the computer simulation and experimental results of the proposed protocol.

Keywords: Real-time Communication, TCP/IP, Ethernet

I. 서 론

현재 가장 널리 사용되는 개방형 컴퓨터 네트워크로는 TCP/IP[14][17]와 이더넷[1] 등을 들 수 있다.

TCP/IP는 대규모의 네트워크 환경에서 서로 다른 네트워크를 경유한 신뢰성 높은 데이터 전송을 위해서 개발되었으며, 전 세계적으로 네트워크 기능이 포함된 대부분의 업무용, 개인용 및 산업용 컴퓨터 시스템에서 지원되고 있는 표준 프로토콜이다. TCP/IP는 프로토콜 구조가 원거리 통신망에 적합한 것임에도 불구하고 그 효율성이 검증되지 않은 산업용 네트워크에 적용하

* 仁荷大學校 自動化工學科

(Dept. of Automation Engineering, Inha Univ.)

接受日: 1998年3月30日, 修正完了日: 1998年7月16日

고자 하는 이유는, 네트워크 기능을 가진 대부분의 시스템에서 공통적으로 지원되는 유일한 프로토콜이기 때문이다. TCP/IP와 함께 가장 널리 사용되는 이더넷 또한 같은 맥락에서 볼 수 있다. 아직까지 이더넷만큼 저렴한 비용과 탁월한 확장성을 제공하는 근거리 통신망은 개발되지 않았다. 이러한 이유들 때문에 TCP/IP와 이더넷을 사용하여 기존의 시스템과 호환성을 유지하며 실시간 특성을 높일 수 있는 프로토콜에 대해서 연구하였고 그 결과 확장된 TCP/IP 프로토콜을 개발하게 되었다.

일반적으로 실시간 시스템이라 함은 외부의 비주기적인 이벤트에 대한 응답이 예측 가능한 시간에 처리되어야만 하는 시스템을 의미한다. 실시간 시스템은 크게 soft real-time과 hard real-time으로 구분된다. hard real-time은 항공기 제어나 원자력 발전소 제어 시스템과 같이 이벤트 처리 시간이 deadline 안으로 한정되고 deadline을 넘어서는 이벤트 처리는 치명적인 문제로 다루어지는 시스템을 의미한다. 반면에 soft real-time은 deadline의 개념과는 상관없이 이벤트 처리가 우선 순위라는 기준에 의해서 이루어진다. 즉, 우선 순위가 높은 이벤트가 가장 먼저 처리되는 시스템을 의미한다. soft real-time의 대표적인 예로는 자동차 설비의 공정 제어나 멀티미디어 통신 등을 들 수 있다.

TCP/IP와 이더넷이 일반적인 비 실시간 시스템에서 사용될 때는 큰 문제가 없지만, 실시간 시스템에 사용되는 경우에는 현실적이고 잠재적인 많은 문제점들이 있다는 것이 알려져 있다. 이더넷에서 사용되는 CSMA/CD 프로토콜에서는 실시간 시스템에서 가장 문제가 되는 전송 지연 시간의 비 예측성이 존재한다. 특히, TCP/IP 프로토콜과 함께 사용될 때는 좀 더 많은 문제점을 유발하게 된다. 이러한 문제들을 해결하기 위해서 많은 노력이 있어 왔으며 이는 크게 3가지 접근 방식으로 구분할 수 있다. MAC (Medium Access Control) 계층의 변경, network 계층의 변경, transport 계층의 변경이 그것이다.

이더넷에서 사용하는 MAC 프로토콜인 CSMA/CD를 보완하려는 많은 연구들이 있었다. 대표적인 예로 BRAM (Broadcast Recognizing Access Method)[4], MBRAM(Modified BRAM)[5]등을 들 수 있다. 새롭게

제시된 프로토콜들은 근본적으로 패킷 충돌을 방지해 줌으로써 예측 가능한 전송 지연 시간을 유지할 수 있다는 장점이 있지만, 이러한 장점에도 불구하고 제안된 프로토콜들이 널리 사용되기 힘든 이유는 다음과 같다.

- 새로운 프로토콜들은 아직까지 표준화 된 것이 없기 때문에 제어 장비나 소프트웨어 개발 회사에서 지원되고 있지 않다. 새로운 프로토콜을 쉽게 사용하기 위해서는 최소한의 호환성이 보장 되어야만 한다.
- 네트워크 어댑터 또는 운영체제 종류만큼의 디바이스 드라이버가 다시 작성 되어야만 한다. 프로토콜 처리가 하드웨어로 구현된 경우에는 수정이 불가능하다.
- 네트워크에 연결된 모든 노드가 동일한 MAC 프로토콜을 사용해야 되기 때문에 새로운 프로토콜을 모든 시스템에 이식 해야만 한다. PCSMA와 같이 기존의 이더넷 네트워크 하에서 통신이 가능한 프로토콜에 대한 연구도 있었지만 대부분의 MAC 프로토콜은 프레임의 스케줄링 방식과 충돌 처리 방식이 다르기 때문에 동일한 네트워크에서 함께 사용될 수 없다.

두 번째 접근 방식인 network 계층 프로토콜로는 ST-II[10]나 RSVP[13]등을 들 수 있다. 이 두 가지 프로토콜은 실시간 멀티미디어 통신을 위해서 개발되었다. ST-II는 네트워크와 네트워크간의 단일 또는 복수 개의 대상에 대해서, 데이터 전송률과 제어 가능한 전송 지연을 필요로 하는 멀티 미디어 통신을 위해서 개발되었다. ST-II는 실시간 통신을 위한 유용한 제어 메커니즘을 가지고 있지만, 표준 TCP/IP와 호환성이 없다는 단점이 있다. 그 뿐 아니라, 이더넷 환경에서 사용되는 경우에는 네트워크에 연결된 모든 노드가 동일한 프로토콜을 지원하지 않게 되면 ST-II의 이점을 살릴 수 없다는 문제가 있다. RSVP는 QoS (Quality-of-Service) 지원을 위한 새로운 network 계층의 프로토콜이다. 호스트가 정해진 QoS를 필요로 하는 경우에 RSVP가 사용된다. 또한 RSVP는 라우터가 모든

노드에게 정해진 라우팅 경로를 따라서 connection-less 접속을 요구할때 사용되기도 한다. RSVP의 기본적인 개념은 현재 연결되어 있는 네트워크의 대역폭을 일정량만큼 할당하는데서 출발한다고 볼 수 있다. RSVP는 네트워크간에 음성 메일이나 디지털 화상 전송 같은 connection-less 서비스에 사용될 때에 좋은 특성을 보여주지만, 기본적으로 connection-less 통신을 사용하기 때문에 분산 처리 시스템과 같은 산업용 네트워크에는 적합하지 않다. 더욱이, RSVP는 ST-II와 같이 제안된 성능을 보장받기 위해서는 단일 네트워크에 접속되어 있는 모든 노드에서 동일한 대역폭 할당 프로토콜이 지원되어야 한다는 한계가 있다.

세 번째 접근 방식은 개선된 transport 계층 프로토콜을 사용하는 방안이다. RTP[9]는 transport 계층에서의 실시간 멀티 미디어 통신을 위해 개발된 대표적인 프로토콜이다. RTP는 자체적으로 전송 제어를 위한 아무런 메커니즘도 제공하지 않는다는 특징 때문에, 대부분의 경우 UDP를 기반으로 동작하게 된다. RTP 뿐만 아니라 모든 멀티미디어 프로토콜은 connection-less 프로토콜을 기반으로 구현되어 있다. 전송되는 데이터 일부가 유실되어도 큰 문제가 발생하지 않기 때문이다. 이러한 특징들 때문에 분산 처리 시스템과 같은 산업용 네트워크에서는 멀티미디어 프로토콜이 적용되기는 힘들다.

이번 연구를 통해서 개발된 확장된 TCP/IP는 transport 계층을 중심으로 변경되었으며, 이더넷을 사용하는 산업용 네트워크에서 soft real-time 특성을 제공하면서 기존의 TCP/IP 프로토콜과 완벽하게 호환성이 유지된다. 다음 장에서는 TCP/IP의 구체적인 문제점들과 확장된 프로토콜의 기본적인 개념들을 설명하고, 이어서 구현 방법, 성능 분석, 실험 결과 등에 대해서 설명한다.

II. TCP/IP의 문제점

이번 장에서는 TCP/IP가 산업용 네트워크에 적용될 경우 문제가 발생할 수 있는 요소들에 대해서 설명한다. 여기서 제시하는 문제점들은 실시간 통신에 장애가 될 수 있는 사항 들일뿐이며, TCP/IP가 일반적인

비 산업용 네트워크에 사용될 경우에 발생하는 문제들은 아니다. 또한, 다음 장에서 제시되는 문제 해결 방안들은 TCP/IP가 단일한 물리적 네트워크로 구성된 근거리 통신망을 기준으로 한 것이다.

1. stream 지향형 프로토콜

Transmission Control Protocol(이하 TCP)에서의 데이터 전송은 기본적으로 stream의 개념을 기반으로 이루어진다. 실제 네트워크로 전송되는 데이터는 블록 단위의 세그먼트로 이루어져 있지만 TCP계층에서 연속적인 octet(1 바이트)의 stream으로 변환된다. stream의 개념은 대부분의 데이터를 아스키 코드로 구성된 텍스트로 전송하는 telnet, rlogin, http, email 등의 일반적인 TCP/IP 응용에는 적당한 형태이지만 블록 단위로 구성된 바이너리 데이터를 주로 처리하는 산업용 네트워크에는 적합하지 않다고 볼 수 있다.

운영체제의 네트워크 모듈에서 수신된 TCP 데이터를 스트림의 형태로 처리하는 방식은 다음과 같다. 먼저 세그먼트라 불리는 블록 단위의 데이터에 포함된 SEQ 번호와 ACK 번호의 확인 과정을 거친 후 데이터의 일부 또는 모두가 수신 노드에서 필요로 하는 것인지를 확인 한 다음 링 형태의 윈도우 버퍼를 사용하여 필요한 데이터를 복사하게 된다. 윈도우 버퍼에 복사된 데이터는 응용 프로그램의 요청에 따라 필요한 양만큼 응용 프로그램 영역의 버퍼로 복사된다. 이러한 과정은 오버 헤드를 증가시킬 뿐 만 아니라 여러 개의 노드와 접속된 상태에서는 응답 시간의 비 예측성을 크게 증가 시키게 된다. TCP 세그먼트 처리 도중에 발생하는 오버 헤드를 줄이기 위한 연구[6][7]들이 있었지만 모두가 stream형태로 다루어지기 이전의 처리 시간 단축을 위한 것들이었다.

2. Sliding window

신뢰성 있고 효율적인 데이터 전송을 목적으로 하는 통신 프로토콜은 sliding window 방식을 채택하는 경우가 많은데 TCP 역시 이러한 전송 방식을 사용한다. Sliding window 알고리즘 하에서는 handshaking으로 발생하는 전송 지연을 최소화하기 위해서, 최근에 전

송된 데이터에 대한 응답을 받지 않았더라도 전송 버퍼에 저장된 데이터를 모두 전송하게 된다. 이러한 통신 방식은 전송 시간이 많이 걸리는 원거리 네트워크 통신의 효율을 높여 주는 역할을 한다. 특히 고속 네트워크와 저속 네트워크간의 통신에서 없어서는 안되는 중요한 기능 중의 하나이다. 그러나 근거리 통신망, 특히 이더넷에서 sliding window 알고리즘은 경우에 따라 급격한 패킷 충돌을 발생시킨다. 대량의 데이터를 전송할 때 sliding window 알고리즘을 적용하게 되면 MAC계층의 전송 버퍼에는 항상 데이터가 저장되어 있기 때문에 노드 하나가 네트워크의 대역폭을 모두 사용하게 된다. 이더넷에서는 트래픽 증가에 따라 패킷 충돌이 급격히 늘어나기 때문에 과도한 전송 지연을 유발하게 된다. 이러한 상황이 오래 지속되지는 않기 때문에 일반적인 용도로 사용되는 경우에는 문제가 되지 않겠지만 soft real-time 특성을 필요로 하는 산업용 네트워크 등에서는 치명적인 문제가 될 수도 있다.

3. TCP finite state machine

TCP 규격은 체계적이고 안정적인 접속 상태 관리를 위해서 finite state machine[14][17]을 정의하고 있다. TCP finite state machine에서는 3way-handshaking이라는 과정을 지나야 데이터 전송이 가능한 접속 상태가 이루어진다. 각 상태는 SYN 및 ACK의 수신 이후에 다음 상태로의 전이가 이루어진다. 이러한 3단계의 상태 전이 과정은 접속을 항상 유지하지 않고 트랜잭션의 형태로 데이터 전송이 이루어지는 통신 방식에서는 실제 데이터 전송을 위한 세그먼트보다 접속 시작과 종료를 위해서 송 수신되는 세그먼트가 더 많게 된다. 이러한 문제를 해결하기 위해서 T/TCP[11][12][16]라는 새로운 연결 방식이 제시되었다. 하나의 세그먼트로 전송이 가능한 데이터의 송 수신을 위해서 기존의 방식으로는 9개의 세그먼트가 필요했지만 T/TCP에서는 단 3개의 세그먼트만이 필요하다. T/TCP는 기존의 TCP와는 달리 하나의 세그먼트 수신으로 다수 개의 상태 전이를 허용함으로써 적은 개수의 세그먼트로도 데이터 송 수신이 가능하게

하였다. T/TCP의 개념은 산업용 네트워크에서 유용하게 사용될 수 있다. 한번의 데이터 전송을 위해서 송 수신되는 세그먼트의 개수가 적어지게 되면 응답 시간이 줄어들 뿐만 아니라 응답 시간의 비 예측성 또한 감소하게 된다.

TCP finite state machine은 접속 종료시 능동적으로 접속 종료를 시도한 노드에서는 TIME_WAIT라고 정의한 상태로 전이된다. TIME_WAIT상태는 네트워크 데이터 통신에서 신뢰성 높은 접속 종료를 위해서 반드시 필요한 사항이지만 MSL(Maximum Segment Lifetime)에 대한 개념이 필요 없는 근거리 통신망에서는 TIME_WAIT 상태는 그다지 의미가 없을 뿐만 아니라 리셋 등으로 TCP 포트 번호가 초기화 된 시스템이 TIME_WAIT 시간 동안 상대방 시스템에 접속을 실패하게 되는 상황이 발생할 수도 있다. 그 뿐 아니라 데이터 베이스와 같이 트랜잭션 처리가 많은 시스템의 경우 TIME_WAIT상태로 남아 있는 접속이 많아짐에 따라 시스템 리소스를 지나치게 낭비하는 문제가 생긴다. T/TCP에서는 이러한 문제에 대한 해결 방안을 제시하고 있으면 이 또한 산업용 네트워크에서 유용하게 사용될 수 있다.

III. 확장된 TCP/IP

이번 장에서는 산업용 네트워크 환경에서 TCP/IP를 사용할 때 문제가 될 수 있는 사항들을 개선하기 위하여 개발된, LAN/TCP라 명명된 확장된 TCP/IP프로토콜에 대해서 설명한다. LAN/TCP는 앞장에서 설명했던 문제점들 중 TCP계층에서 실시간 특성을 대폭 개선할 수 있는 항목을 중심으로 개발된 프로토콜로써 이더넷과 TCP/IP를 기반으로 구성된 네트워크에서 soft real-time 특성을 제공하며 기존의 TCP/IP와 완벽하게 호환된다. 이러한 목적 달성을 위해서 4가지 개념이 도입되었다. TIL (Transmission Interval Limitation), PTM (Periodic Transmission Mode), pseudo window size, datagram 지향형 프로토콜 등이 그 것이다. 이번 장에서는 이러한 새로운 개념들에 대해서 구체적으로 설명하고 다음 장에서는 그 구현 방법에 대해서 설명한다. T/TCP와 관련된 사항은 이미 많은 연구들을 통해

서 성능 평가가 이루어진 상태이므로 그에 대한 언급은 하지 않는다.

1. 전송 주기 제한

TIL(Transmission Interval Limitation)의 기본적인 아이디어는 각 노드에서 데이터의 전송 주기를 제한함으로써 대역폭의 일정량을 할당하는 것이다. 전송되는 데이터의 크기가 일정하지 않기 때문에 단순히 대역폭 할당이라는 표현을 쓸 수는 없지만 전송 가능한 최대 길이의 데이터를 전송한다는 가정하에서는 대역폭 할당이라고 할 수 있다. 할당 될 수 있는 대역폭의 크기는 운영 체제의 구조, 시스템 클락, 물리 계층의 전송 속도 등에 따라 결정된다. LAN/TCP는 sliding window 알고리즘을 사용하지는 않지만 호환성을 유지하기 위해서 pseudo window size를 이용해서 최소한의 sliding window만을 사용하게 된다. pseudo window size에 대해서는 다음절에서 설명한다.

2. Pseudo window size

기존의 TCP/IP와 호환성을 유지하는 것은 LAN/TCP의 큰 특징 중의 하나이다. LAN/TCP는 sliding window 알고리즘을 사용하는 기존의 노드와 통신할 때 대역폭 할당을 실현할 수 있는 방법을 제공한다. 표준 TCP/IP는 sliding window 알고리즘 구현을 위해서 window size advertisement[14][17]라는 기능을 가지고 있다. TCP 헤더의 window size는 상대방 노드에 있는 TCP 세션의 사용 가능한 버퍼의 크기를 의미한다. 각 노드는 최근에 수신된 세그먼트의 window size를 기준으로 전송 가능한 최대한의 데이터를 세그먼트로 나누어 전송하게 되는데 LAN/TCP에서는 필요한 경우 window size의 크기를 실제보다 작은 크기로 상대방 노드에게 알려주게 된다. 이것이 바로 pseudo window size이다. pseudo window size를 사용함으로써 기존의 TCP/IP 시스템이 지나치게 많은 양의 window를 전송하는 것을 최대한 억제하게 된다. LAN-TCP에서는 pseudo window size를 이더넷의 MTU에 맞게 설정하고 있다.

3. Periodic transmission mode

주기적인 데이터 전송은 실시간 모니터링이나 제어 등에서 가장 기본이 되는 전송 방식이다. 표준 TCP/IP의 경우 모든 주기적인 데이터에 대한 응답 세그먼트를 전송하게 된다 (전송 주거나 운영체제에서 사용되는 delayed ACK 타이머에 따라 조금씩 다를 수 있다). 근거리 통신망에서는 이러한 형태의 데이터들에 대해서 다음 번 데이터 수신 시간을 예측 할 수 있다. 이를 위해서 LAN-TCP에서는 PTM이라 불리는 새로운 형태의 전송 방식을 지원한다.

TCP는 세그먼트의 종류를 구분하기 위해서 URG, ACK, PSH, RST, SYN, FIN 등 6개의 비트를 사용하고 있다. 각 비트는 TCP state machine의 상태를 전환시키는데 사용되고 있다. 이 중에서 ACK비트는 PTM에서 매우 중요한 의미를 갖는다. TCP에서 세그먼트에 대한 응답 과정은 한쪽 노드에서 보낸 세그먼트의 sequence number에 대한 응답 세그먼트를 보내는 방식으로 이루어진다. sliding window 알고리즘을 사용하게 되면 여러 개의 세그먼트에 대해서 하나의 응답 세그먼트만을 전송하게 되고 세그먼트의 개수는 운영체제에서 사용하는 delayed ACK 타이머에 의해서 결정된다. 이더넷을 사용하는 근거리 통신망에서는 보통 3~5개 정도의 세그먼트에 대한 응답 세그먼트를 전송한다(구체적인 동작 원리는 [14][17]참조). PTM은 delayed ACK 타이머를 사용하지는 않지만, ACK세그먼트의 잦은 전송을 막기 위해서 delayed ACK 타이머의 주기를 매우 길게 설정하는 것과 같다고 볼 수 있다. PTM 모드에서 ACK 세그먼트는 window size advertisement와 반응답(negative acknowledgement)의 용도로 사용된다. PTM 모드에서 데이터 전송 도중에 상대방 노드의 window size가 MSS(Maximum Segment Size)보다 작아지면 PSH 플래그를 전송하게 되고 상대방 노드는 ACK와 함께 현재 사용 가능한 버퍼의 크기를 window size로 전송하게 된다(그림 1 참조). 예를 들어 상대방 노드의 전송 버퍼의 크기가 8192바이트이고 데이터의 길이가 50바이트인 경우에 약 150개 정도의 세그먼트 전송에 대해서 하나의 응답 세그먼트가 발생하게 된다. 크기가 작은 응답 세그먼트의 감

소는 단순한 대역폭 사용량의 감소보다 큰 의미를 갖는다. 작은 크기의 세그먼트일수록 패킷 충돌의 가능성이 높기 때문이다. 이는 곧 패킷 충돌의 감소와 그에 따른 응답 시간의 비 예측성 감소를 의미한다.

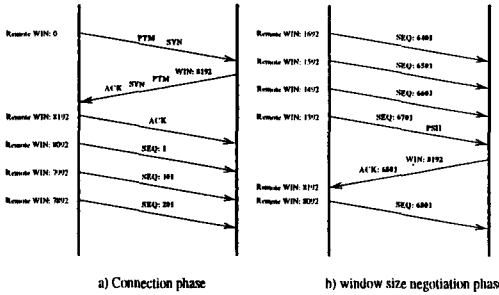


그림 1 PTM 접속에서의 세그먼트 전송 방식
Fig 1. Time line for PTM connection

4. Datagram 지향형 프로토콜

앞장에서 설명한 stream 지향형 프로토콜의 문제를 해결하기 위하여 LAN/TCP에서는 datagram 형태의 TCP 접속을 허용하고 있다. TCP의 접속 형태를 stream과 datagram으로 구분하는 것은 BSD socket의 기본 개념에 근거한 것이다. LAN/TCP 구현도 BSD socket에 기초를 두고 있기 때문에 이 논문에서는 TCP 접속을 stream과 datagram으로 표현하고 있다. BSD socket의 socket() 시스템 콜에서는 datagram 형태로 데이터를 전송하기 위해서는 UDP 접속을 사용해야만 하는데 LAN/TCP를 구현하기 위해서 개발한 실시간 운영체제에서는 datagram 형태의 TCP접속이 가능하도록 하였다. datagram 형태로 접속이 된 상태에서는 stream 처리와 관련된 루틴이 없어지고 네트워크 버퍼에서 응용 프로그램 영역으로 직접 복사가 이루어지기 때문에 네트워크 커널의 오버 헤드로 발생하는 응답 시간의 비 예측성이 크게 감소하게 된다.

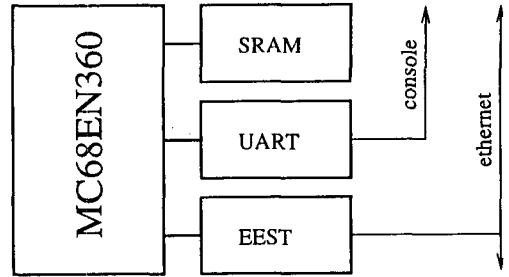


그림 2 실험용 하드웨어 구성도
Fig 2. Block diagram of test-bed

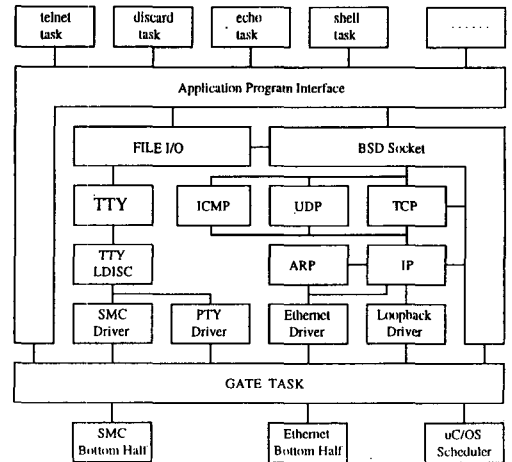


그림 3. LAN/TCP 구현을 위해 개발된 운영체제
Fig 3. Realtime operating system developed for LAN/TCP implementation

IV. 구현

이번 절에서는 LAN-TCP를 구현하기 위해서 고려해야 되는 사항들과 실제 구현된 시스템에 대해서 설명한다.

프로토콜 구현을 위해서 실험용 시스템을 개발하였다. 실험용 시스템은 MC68EN360, 512K RAM, UTP interface 그리고 직렬 디버깅 포트로 구성되었으며 μ C/OS[8]를 기반으로 BSD socket interface를 포함한 TCP/IP 커널과 유닉스 형태의 디바이스 드라이버를

장착한 실시간 운영체제를 개발하여 사용하였다. 개발된 시스템에서는 TCP state machine을 비롯한 대부분의 기능이 구현[15][18] 되었지만 slow start 알고리즘, RTT 최적화 알고리즘, nagle 알고리즘[14][17]과 같이 원거리 네트워크 통신의 효율을 높여 주는 알고리즘들은 구현되지 않았다. 실험용 시스템과 개발된 실시간 운영체제의 대략적인 구조를 그림 2와 그림 3에 도시하였다.

LAN/TCP 개발시 실제 구현과 관련하여 고려해야 하는 사항들은 다음과 같다.

1. 접속 관리

LAN/TCP에서의 접속 시작과 종결 과정은 PTM 접속을 제외하고는 기존의 표준 TCP에서와 동일하다. 현재 개발된 LAN-TCP커널에서는 실험적으로 TCP헤더에서 사용되고 있지 않은 2개의 비트를 할당해서 사용하고 있다. 하나는 TIL 비트로 전송 주기 제한을 위한 것이고 또 하나는 PTM 비트로 PTM 접속을 위한 것이다. 두 비트의 위치는 그림 4에서 보는바와 같다.

헤더를 변경하는 것은 바람직하지 않은 방법이지만 TCP 옵션을 추가해서 사용하게 되면 송수신되는 패킷을 모니터링 하면서 LAN/TCP 패킷과 TCP패킷을 구분할 수가 없기 때문에 개발 기간 중에는 헤더를 변경해서 사용하였다.

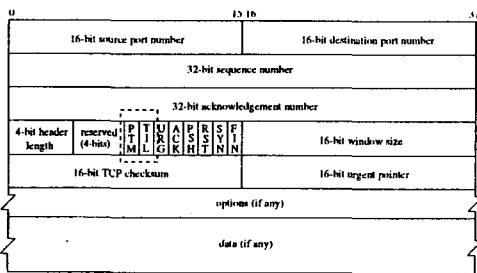


그림 4 LAN/TCP 세그먼트의 헤더 구조

Fig 4. Header for LAN/TCP segment

2. 추가적인 TCP 타이머

TCP계층에서는 전송 제어를 위해서 많은 타이머를 필요로 한다. 대표적인 타이머로는 retransmission 타이

머, delayed ACK 타이머, keep alive 타이머, persist 타이머[14][17] 등이 있다. LAN/TCP에서는 이러한 타이머들 이외에 새로운 2개의 타이머를 추가하였다.

in-band 타이머와 PTM 타이머가 그것이다. in-band 타이머는 전송 주기 제한에 사용되고, PTM 타이머는 PTM 접속 상태에서 반응답(negative ack)용 ACK 세그먼트를 전송할 때 사용된다.

3. Periodic transmission mode

PTM 접속 중에 PTM 타이머는 다음별 세그먼트가 도착하지 않았다는 것을 알려준다. 데이터 송수신 중에 노이즈에 의해서 세그먼트 하나가 유실된 경우 PTM 타이머 이벤트가 발생하게 되고, 이에 따라 ACK 세그먼트가 전송된다. ACK 세그먼트를 수신한 상대방 노드에서는 전송된 ACK 세그먼트의 sequence number를 검사한 후 적당한 위치에서부터 전송을 다시 시작하게 된다. LAN-TCP는 표준 TCP와 마찬가지로 전송된 세그먼트를 버퍼에서 제거하지 않고 window size advertisement를 위한 ACK 세그먼트가 수신되면 전송된 세그먼트를 버퍼에서 제거하게 된다. window size advertisement를 위한 ACK 세그먼트는 PSH플래그가 전송된 경우에만 그에 대한 응답으로 상대방 노드로부터 전송된다.

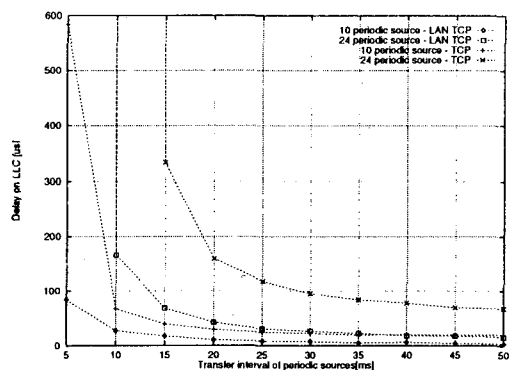


그림 5. periodic 소스의 부하에 의한 LLC에서의 전송 지연

Fig 5. Delay on LLC under loads generated by periodic sources

V. 성능 분석

1. 시뮬레이션 모델

시뮬레이션 모델은 periodic 서버, burst 서버, burst 클라이언트, periodic 클라이언트 등 4종류의 노드로 구성되어 있다. 여기서 periodic 서버란 주기적으로 전송되는 데이터를 받아서 처리하는 노드이고 periodic 클라이언트는 서버로 전송될 데이터를 주기적으로 발생시키는 노드이다. 또, burst 서버는 한 순간에 대량의 데이터를 받아서 처리하는 노드이고 burst 클라이언트는 대량의 데이터를 발생시키는 노드이다. LAN/TCP와 표준 TCP/IP 프로토콜을 periodic과 burst 서버에 대해서 모두 시뮬레이션 하였다. periodic 서버와 periodic 클라이언트간의 데이터 전송의 경우 표준 TCP/IP 프로토콜을 사용하는 노드는 전송된 데이터 세그먼트에 대한 응답 세그먼트를 매번 전송하는 것으로 가정하였고 LAN-TCP를 사용하는 노드는 응답 세그먼트를 발생하지 않는 것으로 가정하였다. burst 서버와 burst 클라이언트간의 데이터 전송시에는, 표준 TCP/IP 노드는 최대한 빠른 시간에 데이터를 전송하게 하였고 LAN-TCP 노드는 전송 주기가 미리 정해진 간격을 넘지 않게 하였다. 실제 상황에 근접한 데이터를 얻기 위해서, burst client에서 발생하는 세그먼트는 확률 분포 함수에 의해서 만들어지지 않고 여러 개의 세그먼트에 대해서 하나의 응답 세그먼트를 보내는 실제 전송 방식을 사용하였다.

2. 시뮬레이션 결과

그림 5, 6, 7에서는 같은 네트워크 환경에서 표준 TCP와 LAN-TCP를 사용했을 때 발생하는 전송 지연, 평균 네트워크 부하, 패킷 충돌 발생 횟수를 각각 도시하였다. 그림 5는 PTM 모드의 효율을 표준 TCP를 사용했을 때와 비교해서 보여주고 있다. 그림 6은 주기적 데이터 전송 도중에 대량의 데이터 전송으로 발생하는 순간적인 전송 지연 시간을 LAN-TCP와 표준 TCP를 비교해서 보여주고 있다.

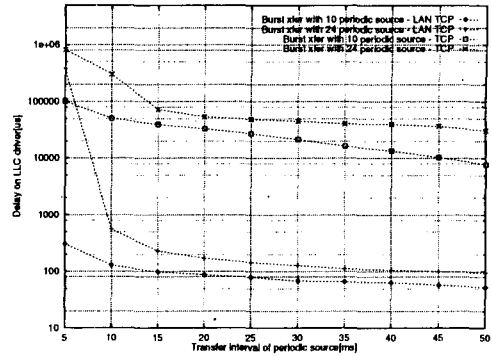


그림 6. 순간적인 대량의 데이터 전송에 의한 전송 지연

Fig 6. Transmission delay by transient burst transmission

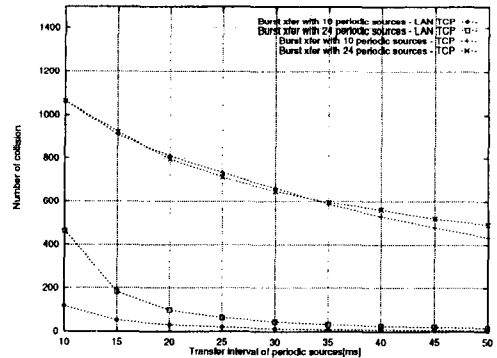


그림 7. 데이터 대량 전송에 의해 발생한 패킷 충돌의 개수

Fig 7. Number of collisions under burst transmission

그림 8에서 알 수 있듯이 대역폭의 30%정도만 사용하고 있는 상황에서 표준 TCP를 사용하여 대량의 데이터가 전송되면 네트워크 부하가 90%정도까지 급격히 증가하게 되며 그 결과 그림 6에서와 같은 과도한 전송 지연이 발생하게 된다. 과도한 전송 지연이 발생하는 이유는 그림 7의 패킷 충돌 횟수 변화를 통해서 쉽게 알 수 있다. 위쪽 2개의 선은 기존의 TCP/IP 프로토콜 하에서 6개와 24개의 노드에서 주기적인 데이터를 전송하는 도중에 대량의 데이터 전송 결과 발생

한 패킷 충돌의 횟수이고 아래쪽 2개의 선은 같은 조건의 LAN/TCP 프로토콜 하에서 시뮬레이션 결과이다. LAN/TCP에서는 패킷 충돌의 횟수가 매우 작다는 것을 알 수 있다. 이처럼 패킷 충돌의 횟수가 작기 때문에 전송 지연이 크게 줄어들게 된다. 일반적으로 이와 같은 상황이 장시간 지속되지 않지만 실시간 시스템에서는 짧은 시간 동안이라도 과도한 네트워크 부하 증가로 인한 문제를 유발 할 수 있다.

3. 개발된 시스템의 실험 결과

정확한 성능 분석을 위해서는 최소한 20여개 정도의 노드가 필요하지만 실제 개발된 시스템은 그보다 훨씬 적기 때문에 PTM 접속의 효율은 정확히 측정되지

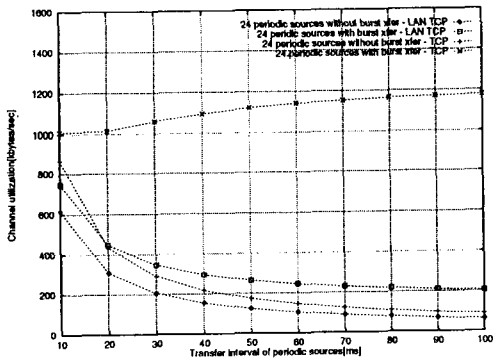


그림 8. 채널 점유율의 변화

Fig 8. A variation of channel utilization

않았다. 실험 환경은 몇 개의 LAN/TCP 노드와 표준 TCP/IP를 사용하는 UNIX 워크스테이션으로 구성되었다. 전송 주기 제한과 PTM 접속의 동작을 검증하기 위해서 Lawrence Berkeley Laboratory에서 개발된 tpdump 라는 프로그램을 사용하였고 패킷 충돌 횟수는 MAC 드라이버에서 제공하는 수치를 이용해서 계산하였다.

표 1은 MAC 계층에서 발생한 패킷 충돌 (collision) 의 횟수를 표시한 것이다. 실험 1은 두개의 노드에서 주기적인 데이터를 전송하면서 다른 두개의 표준

TCP/IP 노드 사이에서 대량의 데이터(800Kbytes)가 전송 되었을 때 발생한 패킷 충돌의 횟수이다. 실험 2의 수치는 두개의 노드에서 주기적인 데이터를 전송하면서 LAN/TCP노드와 표준 TCP노드 사이에서 pseudo window size를 이용하여 대량의 데이터(800Kbytes)가 전송 되었을 때 발생한 패킷 충돌의 횟수이다. 주기적으로 데이터를 발생시키는 노드가 2개뿐인데도 실험 1과 2에서의 패킷 충돌 횟수에 큰 차이가 나는 것을 알 수 있다.

실험 3은 대량의 데이터를 전송하는 노드가 2개에서 4개로 증가한 경우로서 실험 2의 결과와 큰 차이가 없다는 것을 알 수 있다. 실험 3을 통해서 LAN/TCP에서는 대량의 데이터 전송이 이루어지는 노드의 수가 증가해도 패킷 충돌의 증가가 크지 않다

표 1. 실험으로 얻어진 패킷 충돌 횟수

Table 1. Experimental result of number of collision.

실험 방법	구분	패킷 충돌 횟수
TIL 미사용 (실험 1)	PTM 미사용	220
	PTM 사용	201
TIL 사용 1 (실험 2)	PTM 미사용	17.4
	PTM 사용	11.2
TIL 사용 2 (실험 3)	PTM 미사용	18.4
	PTM 사용	15.5

는 사실을 보여주고 있다. 모든 실험에서 공통적으로 주기적인 데이터를 기존의 방식대로 전송 했을 때와 PTM 접속을 사용했을 때를 비교하고 있는데, 주기적 데이터를 발생시키는 노드가 2개밖에 없기 때문에 그 차이는 크지 않지만 노드의 수가 많아질 수록 그 차이는 급격히 증가할 것으로 예상된다.

VI. 결론

LAN/TCP를 개발하게 된 이유는 산업계에서 널리 사용되고 있는 TCP/IP와 이더넷 환경을 분산 처리 시스템과 같은 soft real-time 통신이 필요한 곳에 적용할 수 있는 가능성을 검증해 보기 위해서이다. LAN/TCP

에서는 전송 주기를 제한하는 방법을 통해서 이더넷을 기반으로 하는 TCP/IP네트워크에서 발생하는 트래픽을 최소화시켰고, 단일 노드가 대역폭 모두를 사용하는 극단적인 상황을 제거 하였으며, 기존의 TCP/IP와의 호환성을 유지하는데 성공하였다. 또한 모니터링과 제어를 위해서 주기적으로 발생하는 데이터를 효과적으로 처리하기 위해서 PTM 이라는 새로운 개념의 접속 방식을 제시하였고 실험을 통해서 이의 유용성을 확인하였다. 비록 노드의 숫자가 부족해서 충분한 실험이 이루어지지는 않았지만 새로운 프로토콜이 표준 TCP/IP와 비교해서 실시간 특성이 좋다는 것이 검증되었다.

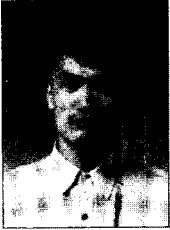
이번 연구는 TCP/IP 프로토콜을 soft real-time 네트워크에 적용하려는 목적으로 시작된 것이기 때문에 hard real-time 네트워크에서의 사용하기에는 많은 문제가 뒤따른다. 특히 CSMA/CD의 근본적인 문제 때문에 hard real-time 네트워크에 적용하는 것은 불가능하다고 봐도 될 것이다. 그러나 MAC 계층에서 이더넷이 아닌 token 네트워크등을 사용하는 방법을 고려한다면 hard real-time 네트워크에 적용하는 것도 연구할 만한 가치가 있다고 생각된다. 향후 이러한 방안에 대한 연구가 계속될 것이다.

참 고 문 헌

[1] R. Metcalf and D. Boggs, "Ethernet: Distributed packet switching for local computer networks", Communication of the ACM, vol.19, 395-404, 1976.
 [2] D. Boggs, J. Mogul, and C. Kent, "Measured capacity of an ethernet: Myths and reality", Proceeding of ACM SIGCOMM '88, pp.222-234, 1988.
 [3] J. F. Shoch and J. A. Hupp, Measured Performance of an Ethernet Local Network, Xerox Corporation, California, 1980.
 [4] I. Chlamtac, W. R. Franta, and K. D. Levin, "BRAM: The Broadcasting Recognizing Access Method", IEEE Transactions on Communication

vol.27, pp.1183-1189, 1979.
 [5] R. P. Signorile, "MBRAM - A priority protocol for PC based local area networks", IEEE network vol.2, pp.55-59,1988.
 [6] D. D. Clark, V. Jacobson, J. Romkey, and H. Salwen, An analysis of TCP processing overhead", IEEE Communication Magazine, pp.23-29, 1989.
 [7] V. Jacobson, "4BSD TCP Header prediction", Computer Communication Review, vol.20, 1990.
 [8] J.J.Labrosse, μ C/OS, the Real-Time Kernel, R & D publications, Inc., Kensas, U.S.A., 1992.
 [9] A.V.T.W. Group, "RTP: A transport protocol for real-time applications", RFC1889, 1996.
 [10] C. W. Group, "Experimental internet stream protocol, version 2(ST-II)", RFC1190, 1990.
 [11] R. Braden, "Extending TCP for transactions - concepts", RFC1379, 1992.
 [12] R. Braden, "T/TCP - TCP extension for transactions functional specification", RFC1644, 1994.
 [13] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reservation protocol(RSVP)", RFC2205, 1997.
 [14] W. R. Stevens, TCP/IP Illustrated, Volume 1, Addison-Wesley Publishing Company, Massachusetts, 1994.
 [15] G. R. Wright and W. R. Stevens, TCP/IP Illustrated, Volume 2: The Implementation, Addison-Wesley Publishing Company, Massachusetts, 1995.
 [16] W. R. Stevens, TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols, Addison-Wesley Publishing Company, Massachusetts, 1996.
 [17] D. Comer, Internetworking with TCP/IP Volume I: Principles, Protocols, and Architecture, Prentice Hall, Inc, Englewood Cliffs, New Jersey, 1995.
 [18] D. Comer and D. L. Stevens, Internetworking with TCP/IP volume II: Design, Implementation, and Internals, Prentice Hall, Inc, Englewood Cliffs, New Jersey, 1994.

 저 자 소 개



尹 泳 燦 (會員申請中)

1996년 2월 인하대학교 자동화공학과 졸업(학사). 1996년 8월 ~ 현재 동대학원 석사과정. 관심분야는 실시간 시스템, 실시간 운영체제, TCP/IP 통신 등임



朴 宰 賢 (正會員)

1986년 2월 서울대학교 제어계측공학과 졸업(학사). 1988년 2월 동대학원 석사과정 졸업. 1994년 2월 동대학원 박사과정 졸업. 1994년 ~ 1995년 University of Michigan 연구원. 1995년 ~ 현재 인하대학교 자동화공학과 조교수. 관심분야는 실시간 시스템, 산업용 통신망, 공장 자동화를 위한 컴퓨터 응용 등임