분산제어 시스템을 위한 실시간 Ethernet 통신 응용

박 재 현
인하대학교 정보통신공학부

1. 서론

제어 시스템을 분산 시스템으로 구성하는 것은 이제는 새로운 일이 아니며, 모두들 당연시 여기는 접근 방법이다. 이와 같은 분산 제어 시스템을 구축하는데 있어서 필수적으로 고려되어야 하는 것이, 분산 제어 시스템 구성 요소를 연결하는 네트웍이다. 제어 시스템을 구성하는 각종 제어 및 계측 시스템을 연결하는데 있어서 네트웍을 도입함으로써, 네트웍 자체로 기인한 각종 불안정한 요소가 제어시스템에 포함된다. 이러한 불안정한 요소로는 대표적으로, 데이터 전달이 되지않는 데이터손실과 데이터가 전달되는데 지연시간이 발생하는 전달지연시간이 있다. 이러한 불안정한 요소는 곧바로 제어시스템의 안정성 등에 영향을 미치게 된다. 따라서 기존의 제어시스템을 분산 제어시스템으로 재구축하거나, 새로운 제어시스템을 분산환경으로 구축하는데 있어서, 가장 중요하게 고려하여 할 점은 이와 같은 불안정 요소를 제거하거나, 영향을 최소화 하는데 있다.
분산 시스템을 연결하는 네트웍의 지연시간은 현실적으로 피할 수 없다. 따라서 제어시스템을 설계할 때 이를 고려하여 설계하는 것이 보편적이며, 실제로 지연시간을 고려하는 설계 방법이 많이 연구되어 왔다. 그런데 이와 같은 네트웍에서의 지연시간은 사용하는 네트웍의 종류에 따라 다양하게 변화된다. 즉 아주 고속의 네트웍에서부터, 저속의 네트웍에 이르기까지, 지연시간의 편차도 매우 크다. 그러나 네트웍을 이용한 분산 제어시스템에서 더 큰 영향을 미치는 것은 네트웍 지연시간의 편차(jitter)이다. 네트웍 지연시간의 편차는 전체 시스템의 불확실성을 높이는 매우 중요한 요소로서, 이의 배제 혹은 축소는 매우 중요한 의미를 가진다 할 수 있다.

네트웍 지연시간의 편차를 제거하는 가장 확실한 방법 중 하나는 네트웍을 통한 데이터 전송의 편차가 없도록 특수하게 설계된 실시간 필드버스(fieldbus)를 채택하는 것이다. 실시간 필드버스는 제어 시스템에서의 중요한 데이터의 전송 지연시간을 확정적으로 보장하도록 설계된 특수한 통신망으로, 실제 제어시스템 구축에 있어서 널리 사용되고 있다. 이와 같은 필드버스에는 주지하는 바와 같이 Profibus, World-FIP, Foundation Fieldbus, CAN, LonWorks 등 주요 응용 분야에 따라 다양한 규격이 제안되어 사용되고 있다. 이러한 필드버스는 분산 제어시스템을 위하여 설계되었으므로, 네트웍 지연시간에 있어서의 불확실성을 제거하는 데는 탁월한 장점을 가지고 있음에도 불구하고, 응용 소프트웨어 개발의 어려움과, 고가의 하드웨어, 그리고 호환성의 결여 등의 내재적인 단점으로 인하여, 아주 널리 사용되는 데는 많은 어려운 점이 따른다.

반면에 정보 기술의 비약적인 발전은 컴퓨터 네트웍 기술에 있어서 많은 진보된 결과를 가져왔으며, 고속의 Ethernet, ATM과 같은 네트웍 하드웨어와, TCP/IP, RTP/RTCP, HTTP 등과 같은 다양한 종류의 프로토콜 등이 표준화되어 널리 사용된다[1,8,14,17]. 이와 같은 정보기술 분야의 네트웍은 분산제어시스템을 구성하는 산업전자 분야에 비할 수 없을 정도로 큰 시장을 형성함으로써, 고성능 하드웨어가 저렴한 가격으로 공급되고 있으며, 매우 다양한 소프트웨어가 널리 개발 이용되고 있다. 따라서 이와 같은 정보기술 분야에서 널리 사용되는 컴퓨터 네트웍기술을 산업전자에 응용하여 분산 제어시스템 구축에 사용하는 것은 필연적인 시대의 흐름으로 생각된다.

그러나 정보기술에서 널리 사용되고 있는 이러한 네트웍 기술을 분산 제어시스템에 그대로 적용하는 데는 적지않은 문제점이 있다. 현재 가장 널리 사용되는 네트웍인 Ethernet 과 TCP/IP 네트웍은 기본적으로 비실시간 광역 네트웍 환경에 적합하도록 설계, 개발되었기 때문에 지연시간이 지나치게 길고, 지연시간의 편차도 매우 커서 제어시스템의 안정성에 심각한 영향을 끼치게 된다. 따라서 본 글에서는 Ethernet 네트웍과 TCP/IP 프로토콜을 분산 제어 시스템에 적용했을 때 발생할 수 있는 문제점을 소개하고, 이를 보완하는 몇 가지 방법을 소개하고자 한다.

2. 하위 계층 확장

2.1 Ethernet MAC 계층의 확장

Ethernet 네트웍을 실시간 분산 시스템에 사용하고자 하는 첫번째 노력은 Ethernet 네트웍의 MAC(Medium Access Control) 계층을 수정하는 노력이다. MAC 계층의 CSMA/CD 방식은 예측할 수 없는 지연을 발생시키며 이러한 지연은 실시간 시스템에 문제를 발생시킬 수 있다. 대표적인 예로 BRAM (Broadcast Recognizing Access Method), MBRAM(Modified BRAM)등을 들 수 있다[4,5]. 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.
- 네트웍 어댑터 또는 운영체제 종류만큼의 디바이스 드라이버가 다시 작성 되어야만 한다. 프로토콜 처리가 하드웨어로 구현된 경우에는 수정이 불가능하다.
- 네트웍에 연결된 모든 노드가 동일한 MAC 프로토콜을 사용해야 되기 때문에 새로운 프로토콜을 모든 시스템에 이식 해야만 한다. PCSMA와 같이 기존의 Ethernet 네트웍에서 통신이 가능한 프로토콜에 대한 연구도 있었지만 대부분의 MAC 프로토콜은 프래임의 스케쥴링 방식과 충돌 처리 방식이 다르기 때문에 동일한 네트웍에서 함께 사용될 수 없다

따라서 MAC 계층의 근본적인 확장이나 변경은 기존 Ethernet과의 호환성 등의 문제 등으로 현실성이 없다. 대신 최근에는 Switched Ethernet의 가격 하락에 따라 이를 이용한 방법이 현실적인 대안으로 제시되고 있다.

2.2 네트웍 계층의 확장

두 번째 접근 방식인 네트웍 계층 프로토콜로는 ST-II나 RSVP등을 들 수 있다[10,13]. 이 두 가지 프로토콜은 실시간 멀티미디어 통신을 위해서 개발되었다. ST-II는 네트웍과 네트웍간의 단일 또는 복수 개의 대상에 대해서, 데이터 전송률과 제어 가능한 전송 지연을 필요로 하는 멀티미디어 통신을 위해서 개발되었다. ST-II는 실시간 통신을 위한 유용한 제어 메커니즘을 가지고 있지만, 표준 TCP/IP와 호환성이 없다는 단점이 있다. 그 뿐 아니라, Ethernet 환경에서 사용되는 경우에는 네트웍에 연결된 모든 노드가 동일한 프로토콜을 지원하지 않게 되면 ST-II의 잇점을 살릴 수 없다는 문제가 있다. RSVP는 QoS (Quality-of-Service) 지원을 위한 새로운 network계층의 프로토콜이다. 호스트가 정해진 QoS를 필요로 하는 경우에 RSVP가 사용된다. 또한 RSVP는 라우터가 모든 노드에게 정해진 라우팅 경로를 따라서 connection-less 접속을 요구할 때 사용되기도 한다. RSVP의 기본적인 개념은 현재 연결되어 있는 네트웍의 대역폭을 일정량만큼 할당하는 데서 출발한다고 볼 수 있다. RSVP는 네트웍간에 음성메일이나 디지털 화상 전송 같은 connection-less 서비스에 사용될 때에 좋은 특성을 보여주지만, 기본적으로 connection-less 통신을 사용하기 때문에 분산처리 시스템과 같은 산업용 네트웍에는 적합하지 않다. 더욱이, RSVP는 ST-II와 같이 제안된 성능을 보장 받기 위해서는 단일 네트웍에 접속되어 있는 모든 노드에서 동일한 대역폭 할당 프로토콜이 지원돼야 한다는 한계가 있다.

최근 IPv6 (Internet Protocol, Version 6)가 대두되면서 IPv6에 flow labeling에 관한 연구가 활발히 이루어지고 있다. IPv6는 flow label을 통하여 실시간 통신기능과 QoS보장을 지원한다. 그러나 이러한 기능은 여러 라우터를 지나는 WAN환경에서 적용된 기술로 일반적인 DCCS에 환경인 LAN환경에서는 여전히 실시간 통신을 위한 문제점을 가지고 있다.

3. Transport계층의 확장

앞서 지적한 바와 같이 Ethernet의 MAC 계층 혹은 네트웍 계층의 확장에 반하여 상위 계층인 Transport 계층과 응용 계층의 확장을 통한 적용 방법에 대한 연구가 있다. 이에 대한 고찰에 앞서 기존 TCP/IP 계층의 Transport계층인 TCP 프로토콜이 실시간 분산 제어 시스템에 적합하지 않는 이유를 정리하면 다음과 같다.

3.1 TCP 계층의 문제점

(1) stream 지향형 프로토콜

Transmission Control Protocol(TCP)에서의 데이터 전송은 기본적으로 stream의 개념을 기반으로 이루어진다. 실제 네트웍으로 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 네트웍에는 적합하지 않다고 볼 수 있다.

운영체제의 네트웍 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 연구들이 있었지만 모두가 stream형태로 다루어지기 이전의 처리 시간 단축을 위한 것 들이었다[6,7].

(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 전송된 데이터에 대한 응답을 받지 않았더라도 전송 버퍼에 저장된 데이터를 모두 전송하게 된다. 이러한 통신 방식은 전송 시간이 많이 걸리는 원거리 네트웍 통신의 효율을 높여 주는 역할을 한다. 특히 고속 네트웍과 저속 네트웍간의 통신에서 없어서는 안되는 중요한 기능 중의 하나이다. 그러나 근거리 통신망, 특히 Ethernet에서 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 네트웍의 대역폭을 모두 사용하게 된다. Ethernet에서는 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 네트웍 등에서는 치명적인 문제가 될 수 도 있다.

3.2 TCP 계층의 실시간적 확장

앞서 지적한TCP가 가지는 많은 문제점을 해결하고 전송의 효율을 높이기 위한 연구로는 T/TCP와 LAN/TCP가 있다. T/TCP는 TCP의 접속 시작과 종료를 위하여 사용되는 세그먼트의 수를 줄일 수 있도록 TCP를 변환하여 오버헤드를 줄이고, 또한 하나의 세그먼트 수신으로 다수 개의 상태 전이를 허용함으로써 적은 수의 세그먼트로 데이터 송수신이 가능하게 하였다[11,12,16]. 이에 반하여 LAN/TCP는 기존의 TCP가 광역 네트웍을 염두에 두고 개발되어 많은 오버헤드를 포함한 비효율적인 프로토콜이라는 점에 착안하여 이들 비효율적인 요소를 제거하고, 통신 지연의 불확실성을 비약적으로 줄이는 프로토콜로서 LAN 환경에 적합하도록 설계된 프로토콜이다[2]. 특히 산업용 분산제어시스템에서 데이터 전송의 많은 부분을 차지하는 주기적 데이터 교환을 위하여 LAN/TCP에서 제공하는 PTM은 TCP의 오버헤드를 많이 줄여서 통신지연의 비예측성을 많이 줄 일 수 있다. 이에 대하여 좀더 자세하게 살펴보면 다음과 같다.

(1) 전송 주기 제한

TIL(Transmission Interval Limitation) 의 기본적인 아이디어는 각 노드에서 데이터의 전송 주기를 제한함으로써 대역폭의 일정량만을 할당하는 것이다. 전송되는 데이터의 크기가 일정하지 않기 때문에 단순히 대역폭 할당이라는 표현을 쓸 수는 없지만 전송 가능한 최대 길이의 데이터를 전송한다는 가정 하에서는 대역폭 할당이라고 할 수 있다. 할당 될 수 있는 대역폭의 크기는 운영 체제의 구조, 시스템 클락, 물리 계층의 전송 속도 등에 따라 결정된다

(2) Pseudo window size

기존의 TCP/IP와 호환성을 유지하는 것은 LAN/TCP의 큰 특징 중의 하나이다. LAN/TCP는 sliding window 알고리즘을 사용하지는 않지만 호환성을 유지하기 위해서 pseudo window size를 이용해서 최소한의 sliding window만을 사용한다. 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 advertisement라는 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 Ethernet의 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 주기나 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 PTM

이라 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. 이중에서 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. Ethernet을 사용하는 근거리 통신망에서는 보통 3~5개 정도의 세그먼트에 대한 응답 세그먼트를 전송한다[14, 15,17]. 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. 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 감소는 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.

4. 응용 계층의 확장

앞서 설명한 방법들이 모두 Ethernet MAC 계층 혹은 TCP/IP 계층의 프로토콜을 확장하여야 하므로 어느 정도는 하드웨어 혹은 운영체제를 변경하여야 하였다. 따라서 이와 같은 불편함을 극복하기 위하여 응용계층에서 실시간 분산 제어시스템을 위한 프로토콜 개발에 대한 연구도 있다. 응용 계층에서 실시간 통신을 지원하기 위한 대표적인 프로토콜로는 RTP/RTCP 프로토콜을 들 수 있다[9].

RTP는 멀티캐스트 또는 유니캐스트 상에서 음성, 영상, 또는 모의 데이터와 같은 실시간 데이터를 전송하는 응용 프로그램에 적합한 트랜스포트 기능을 제공한다. 여기서 트랜스포트의 의미는 데이터 송신을 다루는 TCP, UDP와 동일한 계층이 아니라 응용 데이터의 송신을 의미한다. 인터넷에서 RTP를 제정한 이유는 모든 멀티미디어 응용 프로그램들이 음성 또는 영상 데이터를 전송할 때 각기 다른 형태로 패킷화할 필요 없이 공통적인 패킷 형태를 제정하여 이를 이용하려는데 있다. 이름은 리얼타임이지만 사용자가 데이터를 송신한 후 데드라인내에 수신이 가능하다는 것을 의미하지는 않는다. 이 경우 실시간의 스트림을 기본으로 하는 전송의 속성을 의미한다. RTP는 end-to-end 전송 서비스를 제공하며 특정 트랜스포트에 종속적이지 않다. 즉, RTP는 일반적으로 UDP의 멀티플랙싱과 체크섬 서비스를 이용하기 위해 UDP위에서 수행되지만 IPX, ATM에서도 수행될 수 있으며 ST-II와 같은 실시간 특성을 가지는 하위 프로토콜위에서도 구현될 수 있다. RTP는 자체는 연결개념이 없기 때문에 connection-oriented 또는 connectionless 하위계층에 상관없이 작동될 수 있지만 주로 데이터의 특성상 connectionless위에서 구현된다 RTP와 RTCP는 실시간 통신을 위하여 개발된 프로토콜이지만 대상이 되는 응용 분야는 인터넷상에서의 음성 및 영상과 같은 멀티미디어 분야이므로 분산제어시스템에 적용 될 경우에는 여러 가지 문제가 있을 수 있다. 이를 정리해보면 다음과 같다.

4.1 RTP의 문제점

RTP를 사용하는 프로그램들은 재전송 알고리즘을 가지고 있지 않다. RTP의 모든 혼잡제어와 흐름제어를 RTCP의 송수신자 보고에 의지하고 있다. 이러한 방법은 전송방식이 멀티케스트 또는 브로드케스트 방식을 주로 사용하는 멀티미디어 데이터의 전송에는 효율적인 방법이다. RTP는 패킷의 손실이 발생할 경우 RTCP의 보고에 따라 전송 주기를 변화시키거나 손실을 예방하기 위하여 같은 데이터의 중복 전송과 같은 방식을 사용한다. 그러나 이러한 방법은 1:1 의 주기적인 통신 방식을 사용하는 일반적인 제어시스템에서는 비효율적일 뿐 아니라 손실된 패킷에 대하여 적절한 재전송이 불가능하게 된다. 이러한 데이터의 손실은 분산제어시스템을 불안정하게 만드는 중요한 요인이 된다.

일반적인 디지털 제어시스템에서 피드백 기반의 Close-loop 제어를 할 경우 제어기와 센서 사이에 주기적인 양방향 데이터 교환이 발생하게 된다. 이러한 데이터 교환은 시스템의 샘플링 주기에 맞추어 이루어 진다. 이러한 데이터 교환에서 RTP는 재전송 알고리즘을 가지고 있지 않기 때문에 패킷의 손상이나 손실이 발생할 경우 이것을 복구 할 수 없다. 이러한 문제는 음성이나 영상 같은 멀티미디어 데이터 전송에서는 별 문제가 되지 않지만 제어 응용 프로그램에서는 이러한 패킷 손상이나 손실이 심각한 문제를 야기 시킬 수 있다.

일반적으로 네트웍 전송속도가 제어시스템의 샘플링 시간보다 충분히 빠르기 때문에 이러한 문제를

해결하기 위하여 시스템 샘플링 주기보다 수 배 빠르게 RTP의 전송 주기를 설정하여 일부 패킷의 손실에도 강인한 시스템을 설계할 수 있다. 이러할 경우 일부 패킷이 유실되더라도 중복 전송되기 때문에 패킷이 수신측에서 패킷을 받지 못할 가능성이 줄어든다. 그러나 이러한 방법은 동일 네트웍 내에 많은 노드가 존재할 경우 많은 패킷 사이의 더 많은 충돌을 야기시킨다. 또한 노드가 증가하면 증가할수록 더 많은 패킷 충돌을 야기시키며 이러할 상황은 네트웍의 이용률을 감소시키고 예측하기 어려운 통신 지연을 가져온다. 이러한 통신 지연은 데드라인을 놓치는 데이터를 발생시킴으로써 시스템을 불안정하게 만드는 요인이 된다.

4.2 RTCP의 문제점

RTP의 혼잡제어는 주기적인 RTCP의 송수신자 보고에 의하여 결정된다. 이러한 주기적인 흐름제어 방식은 예측 불가능한 임의 대량 노드들이 동적으로 세션에 참석하거나 탈퇴하는 멀티미디어 데이터의 전송에는 비교적 효율적인 흐름 제어를 가능하게 도와준다. 그러나, 일반적인 분산제어시스템의 노드들은 인터넷사의 멀티미디어 응용 프로그램과는 달리 동작중에 예측할 수 없는 대량의 노드들이 대량으로 세션에 참석하거나 탈퇴하는 경우가 없다. 이러한 분산제어시스템에서는 주기적 전송 방식의 흐름제어 방식은 오버헤드가 된다. 또한 전송주기는 전체 트래픽의 5% 이내로 정의 되어있다. 그러므로 많은 노드를 가지는 세션에서는 RTCP의 전송주기는 매우 길어져서 적절한 흐름제어를 못하는 경우가 발생한다.

4.3 RTP/RTCP 프로토콜의 확장

앞서 밝힌 것과 같은 이유로 인하여 RTP/RTCP 프로토콜은 그대로 분산 제어 시스템에 이용될 수 없으므로 이를 토대로 확장된 RTP/RTCP 프로토콜이 제안되었다[3].

(1) Event-triggered RTCP

우선 주기적인 RTCP 통신으로 인한 지나친 RTCP 패킷을 제한하기 위하여 Event-triggered RTCP(이하 E-RTCP)를 이용한다. E-RTCP의 주기능은 연결정보의 전달과 RTP와 E-RTP의 주기를 전달한다. 제어시스템의 Sampling주기는 시스템의 특성에 따라 다르기 때문에 RTP 패킷의 전송이 시작되기 전에 전송측과 수신측에 모두 전송되어야 한다. 또한 제어기와 동작기, 센서 사이에 연결관계를 위한 데이터도 E-RTCP 패킷을 통하여 전송한다. 이러한 정보를 전송하기 위하여 E-RTCP는 기존의 RTCP의 APP 패킷을 사용한다. 제안된 E-RTCP 메시지는 다음과 같은 조건일 경우에만 전송된다.

- 시스템이 초기화 될 때
- 임의 노드가 세션에 참석 또는 탈퇴 할 때
- 임의 노드가 자신의 전송 주기의 변경을 요청할 때
- 제한된 패킷 손실량 또는 네트웍 지연이 발생했을 때

E-RTP/RTCP는 송수신자 보고에 의하여 전송량을 조정하지 않기 때문에 E-RTCP를 각 노드에게 전송하는 노드를 필요로 하게 된다. 이 노드는 일반적으로 전체 시스템을 모니터링하는 노드 또는 주 제어기에 포함한다. 이러한 노드는 시스템 초기화 할 때 연결된 노드에 초기 값을 전송하며 네트웍에 사용량이 총 네트웍의 사용량을 초과 할 경우 적절할 노드에 전송주기를 조절하는 역할을 한다.

(2) Periodic Transmission

주기적 데이터 전송은 분산제어시스템의 기본적인 데이터 교환 방법 중 하나이다. E-RTP/RTCP는 이러한 주기적인 데이터 교환을 위하여 PTM/RTP (Periodic Transmission Mode over PTM) 라 불리는 신뢰성 있는 주기적인 데이터 전송 방식을 제안한다.

CSMA/CD 방식의 MAC 계층을 가지는 네트웍 구조에서 다중 접속을 허용하면서 명시적인 대역폭 할당은 불가능 하지만, 전송되는 메시지의 크기와 주기를 제한함으로써 총 전송량의 합을 제어 할 수 있다. PTM/RTP는 데이터를 전송하는 주기의 제어 뿐만 아니라 패킷 손실에 의한 재전송까지 특정 범위 내로 조정함으로써 항상 정해진 총 대역폭이내로 조정이 가능하다.

5. 결론

본 기고에서는 현재 정보기술(IT)에서 널리 사용되고 있는 Ethernet 네트웍 및 TCP/IP 프로토콜을 분산 제어시스템에 적용할 때 고려해야 할 여러 사항들 중 실시간 통신에 관련된 사항들을 중심으로 검토하였고 이들 문제점들을 해결하는 여러 연구 결과들을 간단하게 소개하였다.

분산 제어 시스템을 위해서는 Fieldbus라는 기존의 실시간 네트웍이 있음에도 불구하고, 저비용, 고효율성을 위하여 개방형 네트웍에 대한 시장의 요구는 분명한 흐름으로 보이며, 이와 같은 목적으로 Ethernet 및 TCP/IP는 분명한 대안으로 보인다. 다만 Ethernet의 MAC이 가지는 CSMA/CD 특성상 지연시간의 불확실성은 존재하나 이를 보완하려는 상위 프로토콜 계층의 노력으로 상당부분 효과를 보고있다. 따라서 이는 고도의 안정성이 요구되는 hard real-time 시스템에서 사용되기는 어려우나 대부분의 soft real-time 시스템에서는 무리없이 사용될 수 있다고 보여진다. 특히 기존의 TCP/IP와의 호환성을 유지하며 실시간성을 보이는 프로토콜들이 제안됨으로써 다양한 응용 프로그램을 그대로 이용할 수 있는 강점이 있다. 이와 같은 방향의 연구는 계속하여 진행될 것으로 보이며, 특히 switched Ethernet의 확산에 따라 더욱 더 높은 가능성이 보인다 할 수 있다.

참고문헌

[1] R. Metcalf and D. Boggs, "Ethernet: Distributed packet switching for local computer networks", Communication of the ACM, vol.19, 395-404, 1976.

[2] J. Park and J. Yoon, "An extended TCP/IP protocol for real-time local area networks," *IFAC Control Engineering Practice*, vol. 6, no. 1, pp. 111-118, Jan. 1998.

[3] J. Park and J. Park, "RTP/RTCP based real-time protocol over Ethernet for distributed control system," Preprint of the 16th IFAC Workshop on DCCS, pp.122-127, Nov. 2000, Sydney, Australia.

[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] D. Comer and D. L. Stevens, Internetworking with TCP/IP volume II: Design, Implementation, and Internals, Prentice Hall, Inc, Englewood Cliffs, New Jersey, 1994

[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.

박 재 현

1986 년 서울대학교 제어계측과 졸업. 동대학원 석사(1988), 동대학 박사(1994). 1995 년-현재 인하대학교 정보통신공학부 부교수. 관심분야는 실시간 컴퓨터 시스템 구조, 컴퓨터 통신망, 실시간 자동제어 시스템, 개방형 제어시스템, 내장형 시스템.