

Residential Ethernet에서의 프레임 동기 방법

강성환[○]* 권용식^{**} 정민석* 김승호*

*경북대학교 컴퓨터공학과, **KT 통신망연구소

shkang[○]@mmlab.knu.ac.kr, yongsik@kt.co.kr, msjung@mmlab.knu.ac.kr, shkim@knu.ac.kr

Timing and Synchronization for Residential Ethernet

S.H Kang[○]* Y.S Kwon^{**} M.S Jung* S.H Kim*

*Dept. of Computer Engineering, Kyungpook National University

**Telecommunication Network Laboratory, KT

요약

이더넷 환경에서 시간에 민감한 데이터를 전송하기 위해 각 장치들은 시간 동기를 맞추는 필요가 있다. 동기식 이더넷에서는 정확한 시간 동기를 위해 IEEE 1588의 서브넷 개념을 이용한다. 현재 연구되고 있는 동기 방법은 LAN상의 장치 중 하나의 기준 되는 동기 정보를 제공하는 장치를 선정하고 이 장치에 의해 다른 모든 장치가 동기를 맞추는 방법을 이용한다. 본 논문은 이러한 동기 정보를 제공하는 장치를 선정하는 알고리즘을 제안한다. 제안된 방법은 기존의 동기 방법과는 달리 동기 정보를 제공하는 장치를 선정하는 동시에 통신하는 장치와 동기를 맞추게 된다. 제안된 방법으로 기존 이더넷에서 시간에 민감한 데이터 전송을 동기화하여 전송 할 수 있고 나아가 무선 통신 및 IEEE 1394와의 연계를 통한 네트워크 전반적인 백본망으로 구성 될 수 있다.

1. 서론

최근 멀티미디어 데이터가 생활에서 큰 비중을 차지하면서 통신망에서 멀티미디어 데이터 전송 기술에 많은 연구가 이루어지고 있다. 이러한 시간에 민감한 데이터들은 최소한의 지터 허용치와 엄격한 서비스 품질(QoS)을 보장해 주어야만 한다. 대부분의 근거리 통신망에서 사용되는 이더넷은 실시간 데이터를 위해 802.1p 방식을 채택하고 있다. 하지만 802.1p 방식은 전송중인 비실시간 데이터에 의해 실시간 데이터의 전송 지연이 발생하고, 대역폭을 미리 예약하지 않기 때문에 과도한 실시간 데이터를 서비스하기에는 예측 가능하지 않은 지터가 발생한다. 이러한 문제점을 해결하기 위해서 이더넷에서 시분할 다중 접속 방식을 이용한 real-time network, 가상 토큰을 사용하여 패킷의 순서와 시간을 예약하여 사용하는 virtual-token network 등의 방법이 제안되고 있다[1,2]. 또한 김슨사(Gibson)와 플럼링크사(Plumblink)에 의해 고려되고 있는 슈퍼프레임 방식은 하나의 사이클 내에 실시간 데이터와 비실시간 데이터가 함께 제공될 수 있도록 하였다. 슈퍼프레임 방식은 실시간 데이터를 위해 대역폭을 미리 예약하기 때문에, 데이터 전송을 위한 경쟁을 최소화 할 수 있어 실시간 데이터를 효과적으로 전송할 수 있다[3,4].

동기식 이더넷을 위하여 IEEE 802.1 Audio/Video Bridging Task Group 표준화 단체에서는 Plug & Play, 전송 절차, 실시간/비실시간 데이터를 함께 제공, 단단한 사이에 2ms이하의 전송 지연 허용, 거의 0에 가까운 지터(jitter)와 원더(wander), 거주지 내의 하나의 시간으로 동기화 등의 여러 목표를 상고 있다[5]. 이 중에서 이더넷을 동기화하기 위해서는 시간의 동기화가 가장 우선적으로 필

요하다. 시간을 동기화 한다는 것은 거주지 망(Residential Ethernet)내의 모든 장치들이 지터나 원더의 영향 없이 하나의 ToD (Time-of-Day) 값에 맞추어 동작하는 것을 말한다. 이렇게 동기가 맞춰지고 나면 실시간 데이터를 전달하기 위하여 자원 예약 과정을 거친다. 또한 데이터를 전달 하면서 비실시간 데이터 전송 구간에 전달되어지는 데이터가 슈퍼 프레임의 한 사이클 안에 완료되지 못해 다음 실시간 데이터 전송 구간의 시작이 지연되는 문제를 막기 위해 셰이핑 기법을 이용한다. 그림 1은 슈퍼 프레임 방식을 이용한 동기식 이더넷의 프레임 전송을 간단히 나타낸 모습이다. GCM (Grand Clock Master)은 일정 주기로 TimeSync 프레임을 전송하여 LAN상의 모든 장치들의 동기를 맞춘다. 그 후 실시간 데이터를 전송하기 전에 예약 프레임 전송하여 자원을 미리 확보한다. 그 다음 실시간 데이터 전송 구간에 실시간 데이터를 전송하고 슈퍼프레임의 나머지 구간에는 지속적으로 비실시간 데이터를 전송한다.

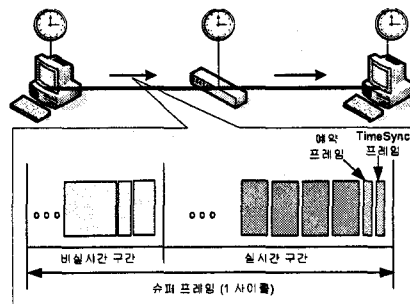


그림 1: 동기식 이더넷의 프레임 전송

이더넷의 동기화를 위해 기가비트 이더넷에서는 송신측에서 전달한 프레임의 프레임들을 통해 전송된 시간 정보를 수신측의 PMA (Physical Medium Attachment)에서 받아들여 PCS (Physical Coding Sublayer)의 동기부에서 동기를 맞추어 상위계층인 MAC 계층으로 전달한다. 이때 송신 과정에서 발생하는 지터나 원더에 대한 고려를 하지 않기 때문에 실제 수신측에서 시간 정보를 재생하여 처리할 때 동기가 맞지 않는 문제가 발생한다. 이러한 문제 때문에 마이크로초 이내의 엄격한 지터나 원더가 요구되는 공장 자동화 시스템이나 데이터 수집 시스템 등에서는 사용될 수 없다.

본 논문에서는 이더넷의 동기 문제를 해결하기 위해 IEEE 1588의 서브넷 개념인 TimeSync 프레임의 동기화 방식의 이더넷에서의 동기화 방법을 제안하고 OPNET 네트워크 시뮬레이션 도구를 이용해 실험 결과를 보인다. 동기식 이더넷에서 시간을 동기화시키기 위해 하나의 장치를 기준으로 이 장치에 의해 모든 장치들이 시간을 맞추게 된다. 이러한 장치를 GCM이라 하며, GCM으로 동작하는 장치를 선정하고 나면, GCM으로부터 시간이 맞춰지는 장치들을 CS (Clock Slave)라 한다. 장치와 장치를 연결하는 중간 장치인 스위칭 허브는 각 포트별로 CM (Clock Master) 또는 CS로 동작한다[6].

본 논문의 구성으로 제 2장에서는 실시간 데이터 전송을 위한 시간 동기화 방법들을 기술하고 제 3장에서는 동기식 이더넷에서의 GCM 선정 알고리즘을 제안하고 GCM이 선정되는 동시에 동기를 맞추는 방법을 기술한다. 제 4장에서는 GCM 선정 시뮬레이션 및 TimeSync 프레임의 전송 주기 시뮬레이션을 분석하고 마지막으로 제 5장에서 결론을 맺는다.

2. 실시간 데이터 전송을 위한 시간 동기화 방법들

2.1 기가비트 이더넷에서 동기 문제

기가비트 이더넷에서는 물리 계층의 PCS 동기부에서 각 장치의 동기를 맞춘다. PCS의 동기부는 PMA로부터 위상이 서로 다른 두 개의 62.5MHz 클럭과 두 클럭에 동기된 한 조의 10비트 데이터열을 수신한다. 즉 두 개의 위상이 다른 62.5MHz에 동기 되어 번갈아 125MHz 클럭 속도로 10비트 데이터열이 수신된다. 이후 PCS 수신부는 디코딩한 8비트 데이터열과 함께 125MHz 클럭을 상위 계층인 MAC에 함께 전송한다. 이러한 기가비트 이더넷의 클럭 동기화 방법은 수신 디바이스 내에서 수신되는 비트열에 대한 동기를 맞추지만 다른 장치와 같은 클럭을 이용하여 동기가 된다고 할 수 없다. 즉, 송신 과정에서 발생한 지터나 원더에 의해 송신 프레임 자체가 여러에 영향을 받는다면 수신측에서 클럭을 추출할 때 잘못된 신호를 추출하게 된다. 그림 2에서 송신측은 수신측 A와 B로 각각 데이터를 전송한다. 송신측에서 생성된 데이터는 수신측 A와 B에 도달할 때 중간 노드와 매체에서 발생하는 지터에 영향을 받게 된다. 그 결과 수신측 A와 B에서 전송 되어진 데이터는 같은 시각에 처리된다고 볼 수 없다.

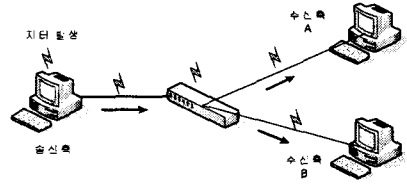


그림 2. 이더넷의 동기 문제

2.2 산업용 이더넷

최근 산업용 이더넷 시장이 급성장함으로써 새로운 관련 기술 또한 계속적으로 연구되고 있다. 산업용 이더넷은 산업의 필드버스(FieldBus) 환경에서 동작하는 이더넷을 말한다. 산업용 이더넷 기술로 PROFINET, Ethernet Powerlink, EtherCAT 등이 있다. 이들은 모두 실시간이 엄격하게 지켜져야 하는 환경을 우선적으로 고려하여 개발된 기술들이다. PROFINET은 전이중 100Mbps 이더넷 상에서 1ms 사이클로 동작하고 1us 지터를 보장하며 실시간 트래픽과 비실시간 트래픽을 동시에 전송할 수 있다. Ethernet Powerlink는 200us에서 1ms 사이의 사이클로 동작하면서 1us 이하의 지터를 보장한다. SoC (Start of Cycle) 메시지를 전송하여 실시간 데이터의 전송을 시작하고 EoC (End of Cycle) 메시지를 전송하여 비실시간 데이터의 전송을 시작한다. EtherCAT은 로컬링 구조를 사용하여 마스터 클럭이 개별 슬레이브 클럭에 대한 전달 지연 오프셋을 결정할 수 있어 1us 보다 짧은 지터를 보장한다. 또한 논리적으로 확장시킨 타임스탬프 데이터 타입으로 분산 클럭을 이용한다. 이와 같은 방법으로 IEEE 802.3 이더넷에 실시간 데이터에 대한 QoS를 보장해 주어 공장 자동화 시스템에 이용되고 있다.

2.3 IEEE 1588

IEEE 1588은 이더넷을 이용한 자동화 시스템에서 실시간 애플리케이션을 위한 정확한 시간 동기를 목적으로 한다[7]. 이더넷에서 마이크로초 이하의 동기를 보장해 주는 최초의 시도였다. 기본 동작은 네트워크상의 가장 정확한 시간으로 모든 장치들을 동기화 시키는 것이다. 이때 장치들은 동기 시간을 제공하는 마스터 혹은 동기에 맞춰지는 슬레이브가 될 수 있다. 마스터와 슬레이브간에 동기를 맞추기 위해 오프셋 및 전파지연시간을 조절한다. 그림 3은 IEEE 1588의 절차를 이용하여 시간 동기를 맞추는 예를 보여준다. 마스터는 일정 주기를 가지고 슬레이브에 Sync 메시지와 마스터의 시간 정보를 포함하고 있는 Follow Up 메시지를 멀티캐스트한다. 이 메시지를 받은 슬레이브는 자신의 시간 정보와 Follow Up 메시지의 시간 정보간의 차이를 구하여 오프셋을 구한 후 자신의 시간 정보에 오프셋을 반영하여 시간을 조정한다. 그 다음 마스터와 슬레이브간의 전송상의 지연을 알아내기 위해 슬레이브는 마스터로 Delay Request 메시지를 전송하고 이에 대한 마스터의 응답으로 딜레이 값을 구한다. 이로써 슬레이브는 딜레이 값까지 알 수 있으므로 오프셋과 딜레이 값을 반영해 자신의 시간 정보를 조정하고 마스터와 동기를 이룬다.

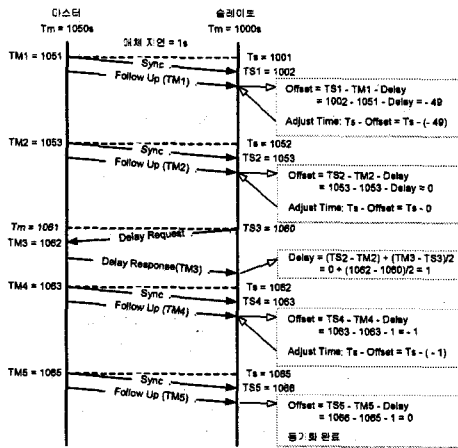


그림 3. IEEE 1588 절차에 의한 동기화

동기를 위해 송신되는 PTP 패킷은 3 계층에 수납되어 전송된다. PTP 메시지에 수납되는 타임스탬프는 PHY 계층과 MAC 사이에 위치한 MII (Media Independent Interface)에서의 프리앰블 송수신 시간이 기록된다. 그리고 마스터는 일정주기로 슬레이브에게 자신의 시간 정보를 멀티캐스트해서 LAN상의 모든 장치들에게 동일한 시간을 제공한다.

3. 프레임 전송의 동기화

동기식 이더넷에서는 IEEE 1588의 방법과 유사하게 LAN상의 장치들에게 시간 동기를 제공한다. 하지만 다음과 같은 차이점이 있다.

- 3 계층의 중계 장치인 라우터를 경유하지 않고, 2 계층의 스위칭 허브를 통해 동기 정보를 전송한다.
- 각 장치에 동기 정보를 조정하는 동시에 자신의 동기 정보를 이웃 장치에 전송한다.
- 마스터 장치에서 슬레이브 장치로 동기 정보를 전송할 때 오프셋을 반영한 전송 시간 정보도 함께 전송한다. 따라서 슬레이브에서는 따로 delay request 메시지를 마스터 장치에 보낼 필요가 없다.

동기식 이더넷에서는 LAN상에 기준 클럭을 제공하는 GCM을 우선적으로 선정할 후 각 장치에 동일한 시간을 제공함으로써 동기를 이룬다.

3.1 지터와 원더

IEEE 802.3 이더넷은 전송 중에 지터나 원더의 고려 없이 수신측에서 시간 정보를 추출하여 장치의 동기를 맞추므로 실시간성을 지켜야하는 장치들에게 정확한 동기를 보장해 줄 수 없다. 여기서 지터란 '이상적인 시간으로부터 누적되지 않은 짧은 기간 동안의 변화량'이다 [8]. 그리고 원더란 지터보다 긴 기간 동안의 변화량을 의미한다. 지터와 원더는 그림 4에서처럼 전송중의 각 장치를 거칠 때 또는 전송 선로 상에서 원하지 않는 필드 신호의 위상 변화가 발생할 때 그 크기가 증가한다.

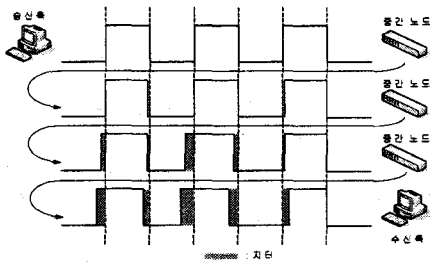


그림 4. 디지털 신호의 원치 않는 위상 변조

지터와 원더의 양을 나타낼 때는 MTIE (Maximum Time Interval Error)를 이용한다. MTIE는 일정 관측구간에서 기준 시간과 현재 관측되는 신호의 시간 차이 중 peak-to-peak 변화이다. MTIE는 식 1로부터 구해진다.

$$MTIE(n\tau_0) \cong \max_{1 \leq k \leq N-n} (\max_{k \leq i \leq k+n} x(i) - \min_{k \leq i \leq k+n} x(i))$$

, where $n = 1, 2, \dots, N-1$ (1)

MTIE는 전체 구간 N에서 시간 구간 n동안 τ_0 의 간격으로 샘플링 했을 때 가장 큰 시간 오차가 일어난 $x(i)$ 에서 가장 작은 시간 오차가 일어난 $x(i)$ 를 뺀 값 중에 가장 큰 값이 된다. ($x(i)$ 는 i번째 위상 에러 샘플 값이다.)

일반적으로 실시간 데이터는 복합 스트림 형태로 제공된다. 복합 스트림 형태의 데이터는 여러 개의 비디오/오디오 스트림을 가지게 되고 복합 경로 형태로 서로 다른 위치로 동시에 전달된다. 하나의 복합 오디오 스트림이 여러 스피커로 전달 될 경우 $\pm 10\mu s$ 내로 지터가 지켜져야 하며 같은 프로그램 안에서 두 개의 비디오가 연관되어 동시에 플레이 되거나 오디오를 수반한 비디오 스트림이 동시에 플레이 되는 경우는 각각 $\pm 80ms$ 내로 지터가 지켜져야 한다[9].

3.2 TimeSync 프레임 형식

TimeSync 프레임은 동기식 이더넷에서 동기 정보를 포함하고 있다. 그림 5의 TimeSync 프레임의 필드들 중 GCM은 TimeSync 프레임의 precedence, uniqueID를 이용하여 선정되며, 각 장치의 동기는 last FlexTime, offsetTime, deltaTime에 의해 이루어진다.

6	da	
6	sa	
2	protocolType	동기식 이더넷 프레임 구분
1	subType	TimeSync 프레임 구분
1	synCount	프레임 카운트
1	hopsCount	GCM으로 부터의 홑수
1	precedence	노드의 우선 순위
8	uniqueID	노드의 고유한 ID
10	lastFlexTime	노드의 타이머
10	offsetTime	GCM으로 부터 누적된 오프셋
6	deltaTime	전달 시간
4	diffRate	타이머의 편차 값
4	lastBaseTime	편차가 조정된 타이머
4	fcs	

그림 5. TimeSync 프레임 형식

각 장치들은 슈퍼프레임의 실시간 구간에서 TimeSync 프레임 송수신하면서 GCM을 선정하고 동기를 맞추어 전체 LAN상의 모든 장치들이 같은 ToD를 이루도록 한다.

3.3 Grand Clock Master 선정

동기식 이더넷을 위해 프레임을 동기화 할 때 우선적으로 고려되어야 하는 것이 LAN상의 장치들 중 기준 클럭을 제공하고 이 기준 클럭에 맞추어 모든 장치가 동일한 시간을 가질 수 있도록 해 주는 GCM의 선정이다. 네트워크가 초기에 세팅될 때 LAN상의 모든 장치들은 GCM 선정과정을 먼저 거친 후 선정된 GCM으로부터 각 노드의 ToD를 동기화 시킬 경우 최초 성립 시 많은 시간이 소요된다. 또한 현재 GCM 보다 우선하는 노드가 추가될 때마다 GCM 선정을 수행하여야 하므로 이는 전체적인 네트워크 부하 증가의 원인이 되는 한편 전송중인 실시간 데이터에 대한 대비책이 없다.

본 논문은 이러한 GCM 선정 방식의 문제점을 해결하기 위해 GCM 선정이 이루어지는 동시에 동기화가 이루어지는 방법을 제안한다. 제안된 GCM 선정 알고리즘을 그림 6에서 나타내었다. 네트워크가 최초로 성립될 때 LAN상의 노드들 중 가장 높은 uniqueID를 가진 노드를 초기값으로 GCM 선정 값으로 결정한다. 그리고 GCM 선정 알고리즘이 수행되는 동안 더 낮은 uniqueID를 가진 노드로부터 TimeSync 프레임이 전송되면 uniqueID를 GCM 선정 값으로 대체한다. 따라서 GCM 선정 값을 등으로 해서 LAN상의 장치들 중 유일하게 하나의 GCM만 선정될 수 있다. GCM으로 선정되기 위해서는 아래의 조건을 만족해야만 한다.

- 자신의 uniqueID가 현재의 GCM 선정 값보다 더 작거나 같은 경우
- 자신의 precedence가 전송받은 precedence보다 더 작거나 같은 경우
- 자신의 uniqueID가 전송받은 uniqueID 보다 더 작은 경우

이때 precedence는 네트워크 관리자의 편의상 특정 장치를 지속적으로 GCM으로 선정될 수 있도록 지정하는 값이다. 만약 GCM으로 선정이 되면 GCM임을 알리는 플래그를 1로 세팅하고 자신의 ToD 값을 자신의 타이머를 이용하여 결정하고, 다른 이웃 노드에게 lastFlexTime과 deltaTime을 TimeSync 프레임에 실어 전송한다. GCM으로 선정되지 못한 장치의 경우 GCM 플래그를 0으로 세팅하고 GCM으로부터 받은 타이머 값을 이용해 자신의 ToD 값을 동기화 시킨다. 새로운 장치가 추가되어 현재 LAN의 GCM 보다 더 우선한다면 새로운 장치는 GCM 선정 값에 자신의 uniqueID를 기록하고 새로운 장치와 송수신하는 노드들은 새로운 ToD 값을 가지게 된다.

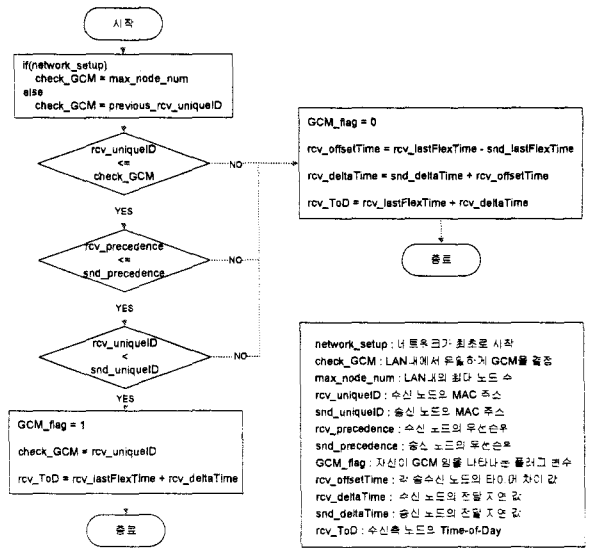


그림 6. 제안된 GCM 선정 알고리즘 및 동기 절차

3.4 장치의 시간 동기화 절차

GCM이 선정되고 나면 LAN상의 각 장치들은 GCM으로부터 동기 정보를 전송받아 자신의 ToD 값을 구한다. GCM으로부터 최초로 동기 정보를 받아 ToD를 조정하고 나면 단계적으로(cascade) 이웃 장치에게 동기 정보를 제공한다. 따라서 LAN상의 모든 장치들은 아래 식 2, 3, 4를 이용하여 모두 같은 ToD 값을 가질 수 있다. 각 장치의 ToD 값을 구하기 위해 자신의 오프셋을 식 2를 이용해 구한다. 식 2를 통해 얻어진 오프셋을 이용하여 자신의 전송 시간을 구한 후 식 3을 통해 자신의 ToD 값을 구해 GCM으로부터 동기를 이룬다.

$$rcv_offsetTime = rcv_lastFlexTime - snd_lastFlexTime \quad (2)$$

$$rcv_deltaTime = snd_deltaTime + rcv_offsetTime \quad (3)$$

$$rcv_ToD = rcv_lastFlexTime + rcv_deltaTime \quad (4)$$

각 식을 이용하여 각 장치의 ToD를 구하는 과정이 그림 7이다. GCM으로부터 동기 정보를 받아 스위칭 허브를 거쳐 CS까지 동기화를 이루는 과정을 나타낸다. 그림 7의 예에서 GCM은 동기 정보를 이웃 장치에게 전송해 주다가 자신보다 더 우선하는 GCM이 추가되면 GCM역시 오프셋과 전송 시간을 이용하여 새로운 GCM으로 동기를 맞추어 한다. 또한 이웃 노드에 동기 정보를 전송할 때 자신의 ToD 값을 변경한 후 ToD 값을 전송하는 방법을 사용하지 않고, 'bypass' 방식을 이용하여 자신의 ToD 값을 변경하는 동시에 동기 정보를 이웃 장치에 전송한다[10].

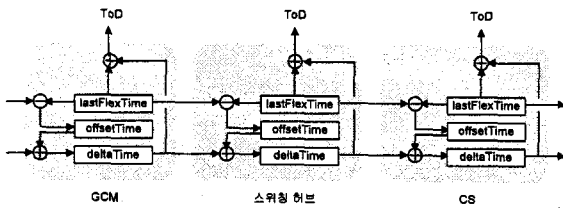


그림 7. ToD 동기화 과정

4. 시뮬레이션 결과 및 고찰

본 논문에서는 동기식 이더넷에서의 프레임 동기 방법 중 GCM 선정 알고리즘을 제안하였다. GCM 선정 알고리즘의 성능을 실험하기 위해 OPNET에서 시나리오를 구성하고 시뮬레이션 하여 실험 결과를 보인다. 그림 8은 프레임 동기를 위한 OPNET 시나리오 구성 형태이다. 2개의 스위칭 허브와 8개의 노드로 구성되어 있으며 스위칭 허브는 각 포트별로 CM 및 CS로 동작하면서 TimeSync 프레임을 목적지로 스위칭하는 역할을 한다. 또한 각 노드는 TimeSync 프레임을 송수신 하면서 GCM, CM 및 CS로 동작한다.

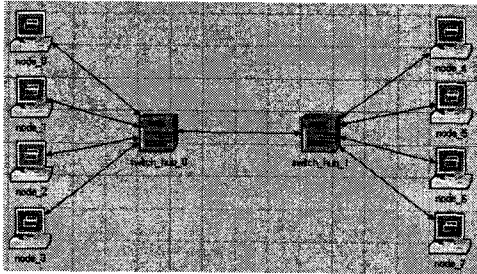


그림 8. 동기 실험을 위한 OPNET 시나리오 구성

그림 9는 스위칭 허브의 프로세스 모델이다. init 상태에서 스위칭 허브를 초기화하고 route_pk 상태에서 패킷을 수신 받아 목적지로 스위칭하는 역할을 한다. 이와 동시에 스위칭 허브의 ToD 값도 결정한다.

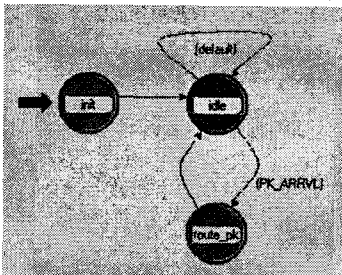


그림 9. 스위칭 허브의 프로세스 모델

그림 10은 각 스위칭 허브에 연결되어 있는 노드의 프로세스 모델이다. start 상태에서 노드마다 통신에 참여하는 시간을 다르게 주어 해당 시간이 되면 통신을 시작하도록 한다. init 상태에서 노드를 초기화 하고 xmt 상태에서 자신 노드에서 발생한 패킷을 목적지로 전송한다. rcv 상태는 패킷을 수신하였을 때 동작하는데 수신된 패킷에서 송신측의 시간 정보를 추출하여 GCM을 선정하고, 각 노드의 ToD를 결정하여 동기를 맞추는 동작을 한다.

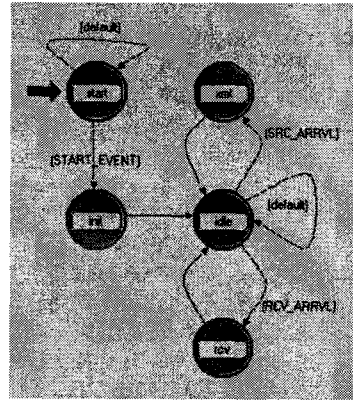


그림 10. 노드의 프로세스 모델

4.1 GCM 선정 시뮬레이션

GCM 선정 시뮬레이션을 통해 동기식 이더넷에서 이미 GCM으로 동작하는 장치가 다른 GCM으로 변경되었을 경우 어떻게 동작하는지를 보인다. GCM 선정 시뮬레이션을 위해 아래와 같이 가정한다.

- precedence 값 : 모두 동일한 값
- lastFlexTime 값 : 초 단위로 1씩 증가
- 각 노드의 타이머의 편차 : 0 (Rate 조정)
- 전체 대역폭 : 1Gbps
- TimeSync 프레임 전송 주기 : 10ms
- TimeSync 프레임 딜레이 : 0s

그림 11은 200초 동안 각 노드의 ToD 값을 구한 결과이다. 8개의 노드에서 새로운 GCM의 선정으로 인해 새롭게 동기가 맞춰지는 모습을 볼 수 있다. 노드 1이 GCM으로 선정되어 동기를 맞춰나가다 100초 이후 노드 0이 참여하게 된다. 이때 노드 0은 노드 1보다 우선순위가 높아 모든 노드들은 노드 0을 GCM으로 하여 새로운 동기를 맞추게 된다. 시뮬레이션 결과 현재 GCM으로 동작하는 노드보다 우선순위가 더 높은 노드가 참여하면 해당 노드를 기준으로 새로운 동기를 맞추는 것을 볼 수 있다.

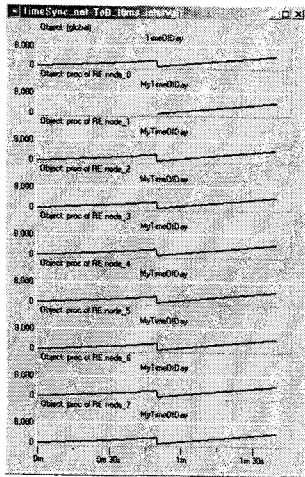


그림 11. 새로운 GCM의 시작으로 ToD의 변화 모습

4.2 TimeSync 프레임의 전송 주기

TimeSync 프레임을 앞 절에서 10ms 주기로 전송하였다. 전송 주기를 짧게 할 경우 좀 더 정확한 동기를 제공할 수 있지만 실제 구현 시 계산의 복잡도가 증가함에 따라 비용이 증가하고 반면에 전송 주기를 길게 할 경우 GCM을 선정하고 ToD를 동기화하는 동안 지터가 증가한다. 그림 12는 100초 동안 전송 주기를 1, 10, 20 및 100ms로 변경하여 실행한 시뮬레이션 결과이다.

시뮬레이션 결과 20ms 이상의 전송 주기로 TimeSync 프레임을 전송할 경우 50초 지점에서 새로운 GCM을 선정하는 과정에서 지터가 발생하고 각 노드의 ToD를 동기화 하는 과정에서도 지터가 증가하는 모습을 볼 수 있다.

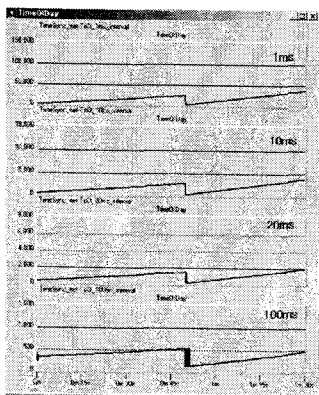


그림 12. TimeSync 프레임의 전송 주기에 따른 지터의 증가 모습

동기식 이더넷에서 GCM 선정 시뮬레이션을 통해 GCM으로 선정된 장치를 기준 시간으로 하여 모든 장치들이 같은 ToD로 동작하는 것을 보였고 GCM이 변경될

경우 새로운 ToD로 동기가 맞춰지는 것을 보였다. 또한 GCM이 TimeSync 프레임을 전송하는 주기를 길게 할 경우 동기를 맞추는데 지터가 증가하였고 동기가 맞춰진 후에도 지터가 증가하는 결과를 볼 수 있었다.

5. 결론

본 논문에서는 동기식 이더넷에서의 프레임을 동기화하는 방법 중 GCM 선정 알고리즘을 제안하였다. 피어 투 피어 방식으로 GCM을 선정하는 동시에 각 장치를 하나의 ToD 값으로 동기를 맞추어 나간다. 이로써 기존의 GCM 선정 프로세스 후 동기를 맞추는 방법보다 한 단계의 프로세스 과정을 줄일 수 있어 동기를 맞추는 시간을 단축시킬 수 있다. 또한 시뮬레이션 결과 제안된 GCM 선정 알고리즘을 통해 새로운 GCM이 통신에 참여할 경우 LAN상의 전체 장치들이 새로운 ToD 값으로 동기가 되는 것을 보였고 TimeSync 프레임의 전송 주기에 따른 지터의 영향을 보였다.

기존 이더넷에 정확한 시간 동기를 제공하므로 IPTV나 다지점 동영상 서비스와 같은 실시간에 민감한 데이터를 전송하는 서비스의 전송에 적용시킬 수 있다. 또한 무선 통신과 IEEE 1394와의 연계를 통해 네트워크 전반적인 백본망으로 발전시킬 수 있다.

참고 문헌

1. J. Kiszka, B. Wagner, Y. Zhang, J. Broenink, "RTnet - A Flexible Hard Real-Time Networking Framework," ETFA 2005, Vol. 1, pp.449-456, 2005
2. F. B. Carrerio, R. Moraes, J. A. Fonseca, F. Vasques, "Real-Time Communication in Unconstrained Shared Ethernet Networks: The Virtual Token-Passing Approach," ETFA 2005, Vol. 1, pp. 425-432, 2005
3. J. Gildred, N. Yeakel, "Synchronous Ethernet Specification Draft v0.39, Pioneer and Gibson Labs, 2003
4. M. J. Teener, "Residential Ethernet - a status report," Plumblink, 2005
5. IEEE 802.1 Audio/Video Bridging Task Group, <http://www.ieee802.org/1/pages/avbridges.html>
6. D. James, "Residential Ethernet (RE) a working paper Draft 0.143," IEEE AVB TG, 2006
7. IEEE Std™ 1588-2002, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," IEEE, 2002
8. ITU-T Recommendation G.701, "General Aspects of Digital Transmission Systems," ITU-T, 1993
9. G. M. Garner, F. F. Feng, E. HS Ryu, K. den Hollander, "Timing and Synchronization for Audio/ Video Applications in a Converged Residential Ethernet Network," CCNC 2006 3rd IEEE, pp. 883-887, 2006
10. J. Jasperneite, K. Shehab, K. Weber, "Enhancements to the Time Synchronization Standard IEEE-1588 for a System of Cascaded Bridges," 2004 IEEE International Workshop, pp. 239-244, 2004