

대용량 멀티미디어 전송을 위한 IEEE 1394 고속 직렬 버스의 성능 분석

(Performance Analysis of IEEE 1394 High Speed Serial Bus for Massive Multimedia Transmission)

이 희 진 [†] 민 구 봉 ^{**} 김 종 권 ^{***}

(Hee-jin Lee) (Ku-bong Min) (Chong-kwon Kim)

요 약 IEEE 1394 고속 직렬 버스(이하 IEEE 1394 혹은 버스)는 실시간 데이터 및 일반 패킷 데이터 전송을 위해 가전제품군과 PC 간의 고속 멀티미디어 통신을 저가로 제공한다. IEEE 1394의 전송 모드는 크게 비동기와 등시성 모드가 있어, 비동기 모드는 최선형 서비스를 제공하고, 등시성 전송 모드는 대역폭을 예약해 QoS를 보장한다. 본 논문은 링크 계층 및 응용 계층 수준에서 IEEE 1394의 성능을 분석하고 각 전송 모드를 비교한다. 응용 계층 수준의 성능 분석을 위해 1394 계층과 상위 계층 사이에 필요한 대표적인 인터페이스인 고정 방식과 적응적 방식을 시스템에 적용하고, 이를 각기 폴링 시스템으로 분석한다. 그리고 분석 모델의 타당성을 시뮬레이션을 통해 확인한다. 그 결과, IEEE 1394의 데이터 전송 모드를 적용적으로 선택하여 버스 접근 지연시간을 줄이고 대역폭 낭비를 막아 버스 이용률을 높일 수 있음을 보였다.

키워드 : IEEE1394 성능 분석, 대용량 멀티미디어 전송, 버스트 데이터 전송, 등시성 전송 모드

Abstract The IEEE 1394 High Speed Serial Bus is a versatile, high-performance, and low-cost method of promoting interoperability between all types of A/V and computing devices. IEEE 1394 provides two transfer modes: asynchronous mode for best effort service and isochronous mode for guaranteed service with bandwidth reservation. This paper shows the bus performance and compares the transfer modes first at the link level and then at the application level. For the application level performance, we analyze the bus systems with fixed and adaptive interfaces, applied between the upper layer and the 1394 layer, using polling systems. Also we verifies the analysis models with simulation studies. Based on our analysis, we conclude that the adaptive interface reduces the bus access time and so increases the bus utilization.

Key words : IEEE1394 performance analysis, massive multimedia transmission, burst data transmission, isochronous transfer mode

1. 서 론

IEEE 1394 고속 직렬 버스는 컴퓨터와 A/V기기를 포함한 가전제품 간의 고속 멀티미디어 통신을 지원한다[1,2]. 저렴하고 사용자에게 친화적이며 확장성이 뛰어나,

가정과 소규모 사업장에서 PC와 디지털 가전 기기를 연결하거나[3], 산업망에서 제어 시스템을 연결하기에 적합하다[4]. 이와 같이 IEEE 1394는 저가의 고속 멀티미디어 통신을 제공함에 따라 차세대 홈 네트워크 기술로 주목받고 있다[5].

IEEE 1394가 네트워크 기술로 대두되면서 IETF는 IEEE 1394와 TCP/IP 프로토콜 간 인터페이스 즉, IEEE 1394 위에 IP 데이터그램을 매핑하는 방법 및 IP 멀티캐스팅을 IEEE 1394 망 기술 위에서 구현하는 방법을 정의하고 있다[6].

기존 연구 분야로는 IP 멀티캐스팅을 보다 효율적으

[†] 비 회 원 : 서울대학교 전기·컴퓨터공학부
heejin@popeye.snu.ac.kr

^{**} 비 회 원 : LG전자 연구원
min90@lge.com

^{***} 종신회원 : 서울대학교 전기·컴퓨터공학부 교수
ckim@popeye.snu.ac.kr

논문접수 : 2001년 8월 3일
심사완료 : 2003년 3월 27일

로 지원하는 방법[7]과 같이 인터넷 프로토콜과의 결합에 대한 연구 외에 1394의 전송 특성에 대한 연구가 진행되어 왔다. Stenberg와 Birk가 IEEE 1394 버스의 대역폭 사용률을 통해 트래픽 로드가 증가하면 비동기 모드의 효율이 낮아진다는 것을 발견하였다[8]. 또한 Ruiz 등은 IEEE 1394 망 기술과 산업 망에서 쓰이던 기존의 버스 기반 망 기술과 성능을 비교하였다. 이들은 IEEE 1394 프로토콜이 전송 속도와 버스 접근 지연 시간에서 가장 뛰어난 성능을 보이면서도 비용이 가장 적게 든다는 것을 보이고 있다[4]. 또한 Lee 등은 IEEE 1394 위에서 IP 데이터그램을 전송할 때, IP 트래픽의 플로우(flow)를 감지하여 비교적 긴 흐름을 가진 데이터 트래픽은 등시성 모드로 전송하고 짧은 데이터 트래픽은 비동기 모드로 전송하는 적응적인 전송 방법을 제안하기도 하였다[9].

본 논문의 목적은 IEEE 1394의 성능을 링크 계층 수준과 응용 계층 수준에서 비교 분석하는 것이다. 우선 기본적인 버스 성능을 보이기 위해 링크 계층 수준에서 IEEE 1394의 전송 모드를 비교 분석한다. 그리고 응용 계층 수준의 성능을 분석하기 위해, 상위 계층과 1394 사이에 필요한 대표적인 인터페이스 방식인 고정 인터페이스 방식과 적응적 인터페이스[9] 방식을 채택하였으며, 이들 두 인터페이스를 채택한 버스 시스템을 각각 폴링 모델로 구성하여 버스 접근 지연 시간을 분석, 비교한다. 그리고 시뮬레이션을 통해 분석 모델의 타당성을 검증하였다.

IEEE 1394 계층을 위한 인터페이스는 데이터 패킷을 전송하고자 하는 상위 계층에게 적절한 1394의 전송 모드를 선택하여 최적의 전송 서비스를 제공하는 것으로 고정 인터페이스가 기본이 된다. 고정 인터페이스는 데이터 패킷을 최선형으로 서비스하는 비동기 전송 서비스를 상위 계층에 고정적으로 제공하는 인터페이스이다. 그리고 적응적 인터페이스는 비동기 모드를 기본으로 제공하다가 특정 조건을 만족하는 데이터 플로우에게 등시성 전송 서비스를 제공하는데, 본 논문에서는 [9]에서 제안된 여러 적응적 인터페이스 방식 중에서 데이터 길이 기반 방식을 선택하여 분석하였다.

분석에 따르면 IEEE 1394의 데이터 전송 모드를 적응적으로 선택한 결과, 버스 접근 지연 시간을 줄여 대역폭 낭비를 막고, 보다 안정되고 빠른 서비스와 높은 버스 이용률을 제공할 수 있음을 알 수 있다.

논문의 구성은 2장에서 IEEE 1394의 기본 특징과 전송 모드를 설명하고, 3장은 링크 계층에서 각 전송 모드의 대역폭 사용률을 비교 분석한다. 4장은 응용 계층 수

준의 버스 성능을 보이기 위해 버스 접근 지연 시간을 수학적으로 분석하고 시뮬레이션 결과와 비교하여 도시한다. 5장에서 본 논문의 결론을 내리고 끝을 맺는다.

2. IEEE 1394 고속 직렬 버스

2.1 IEEE 1394 개요

IEEE 1394는 직렬 케이블을 전송 매체로 사용하는 고속 직렬 버스로, 현재 100/200/400Mbps의 전송 속도를 제공한다. 하나의 버스는 최대 63개의 노드(node)를 지원하는데, 이들은 루트(root) 노드를 기준으로 트리 구조를 이루게 된다. 버스는 여러 노드가 공유하는 자원이므로 매체 접근 제어가 필요한데, IEEE 1394에서는 전송 준비가 된 노드는 매 사이클마다 아비트레이션(Arbitration)을 통해 루트에서 가까운 순서대로 접근 권한이 부여된다. 그리고 IEEE 1394는 자동으로 삽입/삭제된 노드를 인식하는 PnP(Plug and Play)를 지원한다. PnP는 버스 리셋(reset)이 일어날 때마다 버스의 구조를 새로 구성해 지원하며, 그 과정은 버스 초기화(bus initialization), 트리 식별(tree identification) 및 자기 식별(self-identification)을 통해 각 노드에게 고유한 노드 ID를 재 할당하여 이루어진다.

IEEE 1394는 크게 두 가지 종류의 전송 모드를 제공한다. 비동기 전송 모드(Asynchronous transfer mode)는 최선형 서비스를 제공하고 등시성 전송 모드(Isochronous transfer mode)는 보장형 서비스를 제공한다.

보장형 서비스를 제공하기 위해 등시성 전송 모드는 대역폭을 예약하고 패킷을 매 125 μ s 마다 정기적으로 전송하도록 최대한 보장한다. 그림 1은 IEEE 1394가 제공하는 125 μ s의 기본 사이클 구조를 보여 준다. 그러나 그림 2에서 보는 바와 같이 전송 중인 비동기 패킷에 의해 한 사이클의 길이가 125 μ s를 넘을 경우, 다음 사이클에서 사이클 시작이 밀린 만큼 보상하게 된다. 등시성 모드의 우선순위가 비동기 모드 보다 높기 때문에, IEEE 1394는 등시성 모드에 의한 대역폭 예약을 전체 대역폭의 80% 이하로 제한하여 비동기 모드 전송이 무기한 지연되는 것을 방지한다. 등시성 전송을 하기 위해서는 대역폭 예약과 함께 채널 번호를 할당받아야 한다. IEEE 1394의 채널은 한 버스에서 최대 64개까지 쓸 수 있는 최소 자원이다. 그리고 각 전송 모드마다 최대 전송 패킷 크기가 정해져 있는데, 등시성 모드의 최대 패킷 크기는 시간으로 나타내어 약 80 μ s이다.

비동기 전송 모드는 등시성 전송에 비해 버스 접근

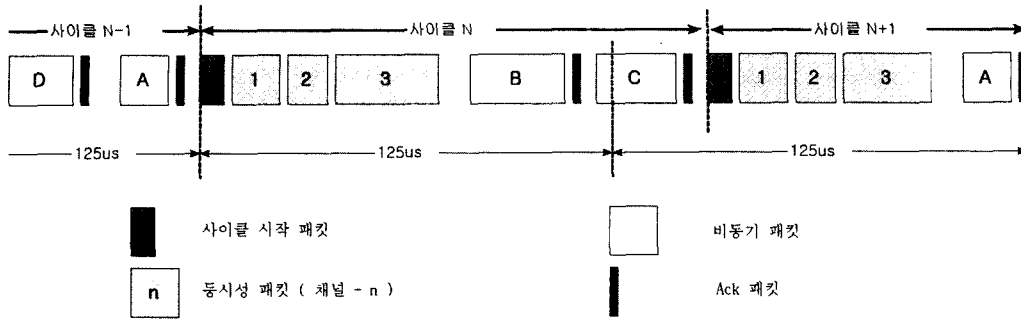


그림 1 사이클 시작 스큐

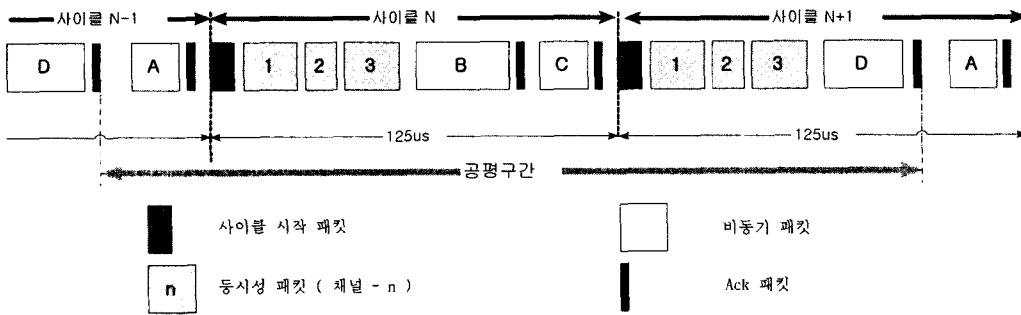


그림 2 IEEE 1394의 사이클 구조 및 공평구간

우선순위가 낮으므로, 매 사이클마다 동시성 모드의 전송이 끝난 다음부터 다음 사이클이 시작하기 전까지 비동기 전송이 허락된다. 비동기 전송 모드에서는 각 노드에게 전송 기회를 고루 제공하기 위해 공평구간(fairness interval)을 정의한다. 노드는 한 공평구간 동안 한 번의 비동기 전송을 할 수 있고, 공평구간에 참가한 모든 노드들이 전송을 마치면 다음 공평구간이 시작된다. 그림 1은 한 사이클 내에서 동시성 전송과 비동기 전송이 어떻게 이루어지는가를 공평구간과 사이클의 관계를 통해 보여 준다. 비동기 모드의 최대 전송 패킷 길이는 약 40 μ s이다.

IEEE 1394에서는 여러 가지 휴지 간격을 두어 패킷과 구간(사이클 및 공평구간)을 구분한다. 표 1은 휴지 간격과 다음절에서 사용한 패킷 길이를 시간 단위로 표시한 것이다. AR은 아비트레이션 구간이고 AG는 비동기 패킷과 Ack 패킷(AP)간의 간격이며, IG는 동시성 트랜잭션간의 간격으로 이들은 무시할 수 있을 정도로 작은 값을 갖는다. 반면 비동기 트랜잭션간의 간격인 SG와 공평구간 간의 간격인 ARG은 IG에 비해 월등히 큰 값을 갖기 때문에 비동기 모드로 대량의 패킷을 전송할 경우 대역폭을 낭비하게 되는 주요 원인이 된다.

표 1 IEEE 1394의 휴지 간격과 패킷의 종류
(100/200/400Mbps의 경우)

약어	이름	길이(min/max)
AR	Arbitration Period	variable
AG	Acknowledge Gap	0.04 μ s / 0.05 μ s
IG	Isochronous Gap	0.04 μ s / 0.05 μ s
SG	Subaction Gap	~ 10 μ s
ARG	Arbitration Reset Gab	~ 20 μ s
RqP	Request Packet Header	1.6/0.8/0.4 μ s
RsP	Response Packet Header	1.28/0.64/0.32 μ s
AP	Acknowledge Packet	0.08/0.04/0.02 μ s
IH	Isochronous Data Block Packet Header	0.32/0.16/0.08 μ s

RqP와 RsP는 비동기 패킷 포맷 중 요구와 응답 패킷이고 AP는 비동기 전송이 잘 되었음을 확인해 주는 Ack 패킷이다. IH는 동시성 전송에서 쓰이는 패킷 헤더의 크기이다.

2.2 비동기 모드 전송

비동기 전송 모드는 요청-응답 모델에 의해 데이터를

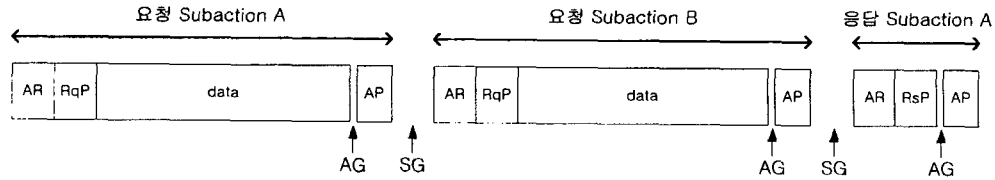


그림 3 분리 트랜잭션

전송한다. 하나의 요청-응답 쌍을 트랜잭션(Transaction)이라고 하며, 읽기, 쓰기, 잠금 트랜잭션이 정의되어 있다. 각각의 요청, 응답 패킷에 대해서는 서로 AP를 주고받는다. 또한, 이 요청과 응답 패킷을 보내는 시점에 따라 분리 트랜잭션(Split Transaction), 연결 트랜잭션(Concatenated Transaction), 통합 트랜잭션(Unified Transaction)으로 분류한다. 비동기 모드에서의 최대 패킷 길이는 약 40 μ s이며 보내려는 데이터가 이보다 클 때에는 분할하여 여러 공평구간에 걸쳐 전송한다.

분리 트랜잭션에서는 요청 패킷을 받은 노드가 즉시 응답을 하지 못하고 아비트레이션 과정을 거친 다음 응답 패킷을 전송한다. 분리 트랜잭션은 응답 노드가 요청 패킷을 처리하고 응답 패킷을 생성하는데 시간이 걸리는 경우에 사용된다. 그림 3은 응답노드가 요청 패킷 A를 받은 다음 즉시 응답을 하지 못하고 아비트레이션을 거친 다음 응답패킷 A를 전송하는 것을 보이고 있다. 요구 패킷을 받은 응답 노드는 응답을 패킷을 즉시 전송하지 못하더라도 AG 이후에 AP를 전송한다. 요구 패킷이나 응답 패킷을 받고 비동기 패킷이 잘 도착했음을 알리기 위해 AP을 전송하는 것을 서브액션(Subaction)이라고 한다. 따라서 한 트랜잭션은 요구 서브액션과 응답 서브액션의 쌍으로 구성된다.

연결 트랜잭션에서는 요청 패킷을 받은 응답 노드가 아비트레이션 과정을 거치지 않고 즉시 응답 패킷을 전송한다. 그림 4에서 요청 패킷을 받은 노드는 AP 다음에 응답 패킷을 전송하는 것을 보여주고 있다.

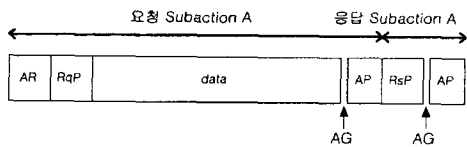


그림 4 연결 트랜잭션

통합 트랜잭션은 응답 노드로부터 대담을 받을 필요가 없는 쓰기 트랜잭션에서만 사용할 수 있다. 요청 패킷에 대한 응답을 요청 패킷의 AP에 실어 보내는 것이다. 그림 5는 통합 트랜잭션의 예를 보여주고 있다.

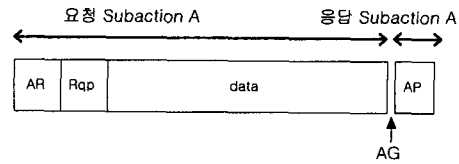


그림 5 통합 트랜잭션

2.3 동시성 모드 전송

QoS를 보장하는 동시성 모드로 전송을 하려면 전송 노드는 먼저 초기 설정이 필요하다. 우선 전송 노드는 채널을 할당받고 대역폭을 예약한 후 그 정보를 바탕으로 수신 노드와 연결을 설정해야 한다. 따라서 동시성 모드에는 채널 할당과 대역폭 예약이 성공할 때까지 각각 한번 이상의 잠금 트랜잭션과 수신 노드에게 수신 채널 정보를 알리기 위한 한 번의 쓰기 트랜잭션을 비동기 모드에서 수행하여야 하는 초기 부하가 있다. 그림 6은 초기 설정이 끝난 후 동시성 트랜잭션과 휴지기간과의 관계를 보인 것이다.

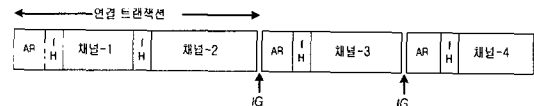


그림 6 동시성 트랜잭션

3. 단일 트래픽 소스의 대역폭 사용 분석

3.1 비동기 전송 모드의 오버헤드

본 논문에서는 임의의 길이의 데이터를 보낼 때, 보내는 데이터 이외의 버스 대역폭 사용을 오버헤드라 정의한다. 단, AR, G 등을 제외하고 직접 해당 데이터와 관련된 오버헤드만을 다루므로 실제 값보다 작다. 다음은 각각의 트랜잭션에 대해서 오버헤드를 구한 것이다.

분리 트랜잭션에서 걸리는 오버헤드는 그림 3으로부터 다음과 같이 구할 수 있다.

$$SG + AR + RqP + AG + AP + SG + AR + RsP + AG + AP = 2AR + 2AG + 2SG + 2AP + RqP + RsP \quad (1)$$

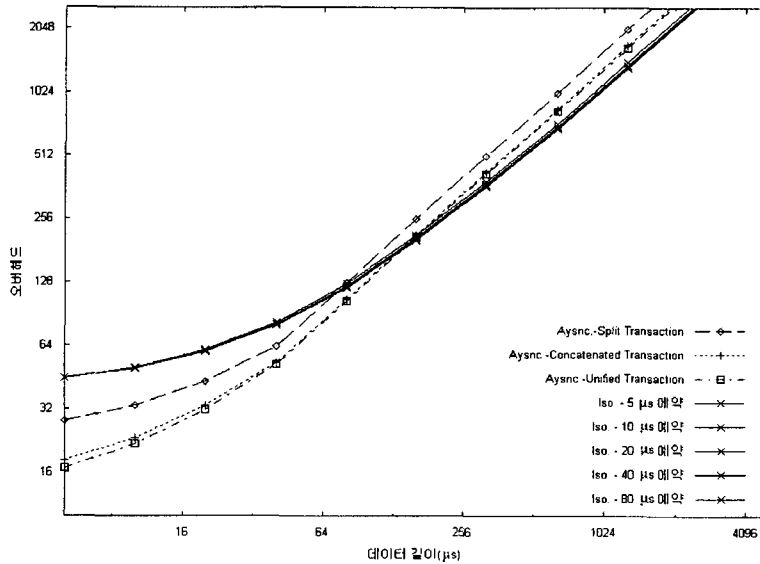


그림 7 비동기 전송과 등시성 전송의 오버헤드(100Mbps)

$x \mu s$ 의 데이터는 $\lceil \frac{x}{40\mu s} \rceil$ 개의 $40 \mu s$ 이하 길이의 패킷으로 분할된다. 이 패킷을 모두 분리 트랜잭션을 이용하여 보낼 때의 총 오버헤드, $A_s(x)$ 는 다음과 같다.

$$A_s(x) = (2AR+2AG+2SG+2AP+RqP+RsP) \times \lceil \frac{x}{40\mu s} \rceil \quad (2)$$

같은 방법으로 그림 4, 5에서 $x \mu s$ 의 데이터를 각각 연결 트랜잭션과 통합 트랜잭션 방식으로 전송할 때의 총 오버헤드, $A_c(x)$ 와 $A_u(x)$ 를 구할 수 있다.

$$A_c(x) = (AR+2AG+SG+2AP+RqP+RsP) \times \lceil \frac{x}{40\mu s} \rceil \quad (3)$$

$$A_u(x) = (AR+AG+SG+AP+RqP) \times \lceil \frac{x}{40\mu s} \rceil \quad (4)$$

3.2 등시성 전송 모드의 오버헤드

등시성 모드로 전송을 하기 위해서는 먼저 채널을 할당받고 대역폭을 예약한 후 수신 노드와 연결을 설정해야 한다. 그림 6은 등시성 전송의 기본으로 오버헤드이므로 $x \mu s$ 의 데이터를 등시성 모드로 전송하기 위해서는 식 (5)와 같은 오버헤드가 든다. 여기서 할당받은 채널은 한 사이클 당 $a \mu s$ 의 버스 대역폭을 사용할 수 있고 채널 할당과 대역폭 예약은 각기 한 번의 잠금 트랜잭션으로 성공하였다고 가정한다.

$$I(x) = \text{Write Transaction} + 2 \times \text{Lock Transaction} + (AR+IH+IG) \times \lceil \frac{x}{a} \rceil \quad (5)$$

3.3 비동기 모드 전송과 등시성 전송 비교

전송할 데이터의 길이에 따라 각각 비동기 모드와 등시성 모드로 전송할 때의 오버헤드를 비교하면 어떤 모드로 전송하는 것이 더 버스 이용률이 높을지 알 수 있다. 그림 7은 보내려는 데이터의 길이에 따른 전송 오버헤드를 나타내고 있다. 비동기 모드 전송은 분리 트랜잭션, 연결 트랜잭션, 통합 트랜잭션 순으로 오버헤드가 작음을 알 수 있다. 등시성 전송의 경우 초기 전송비용이 큰 반면 한 사이클에서 할당받는 버스 대역폭의 크기가 프로토콜 오버헤드에 크게 영향을 미치지 않는다. 따라서 보내려는 데이터의 길이가 어느 정도 이상 커지면 등시성 전송이 더 효율적인 것을 알 수 있다. 이것은 비동기 모드로 보낼 때 생기는 SG(Subaction Gap)가 다른 오버헤드에 비해 매우 크기 때문이다. ARG(Arbitration Reset Gab)을 반영하면 비동기 모드의 오버헤드는 더욱 증가한다.

분석 결과 보내려는 데이터의 길이가 약 $80 \sim 120 \mu s$ 이상만 되어도 등시성 모드로 전송하는 것이 대역폭 낭비가 적음을 알 수 있다. 이는 파일전송과 같은 비실시간 데이터 전송이라도 충분히 큰 데이터를 전송할 경우 등시성 모드 전송을 통해 버스 이용률을 높일 수 있음을 의미한다. 또한 제어 메시지와 같이 전송할 길이가 작은 경우 비동기 모드로 전송하면 대역폭과 지연시간 모두에서 이익을 볼 수 있다. 그림 8과 9는 각기 버스 속도가 200Mbps, 400Mbps인 경우이다. 전송라인이

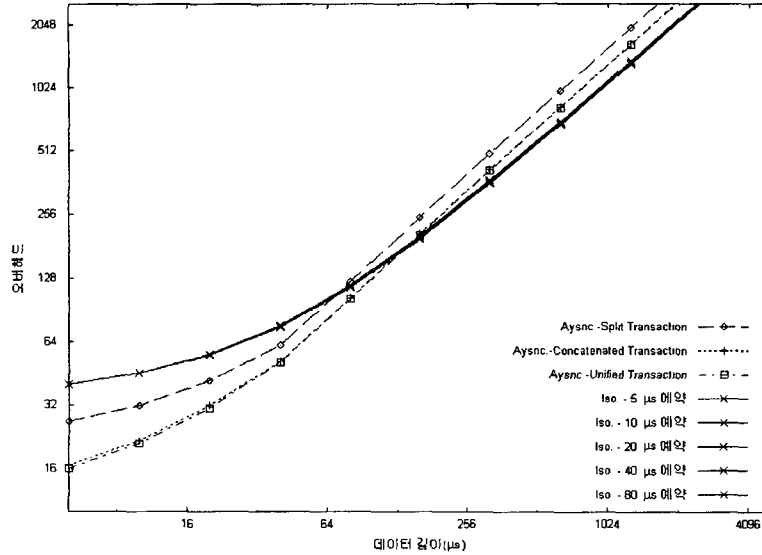


그림 8 비동기 전송과 등시성 전송의 오버헤드(200Mbps)

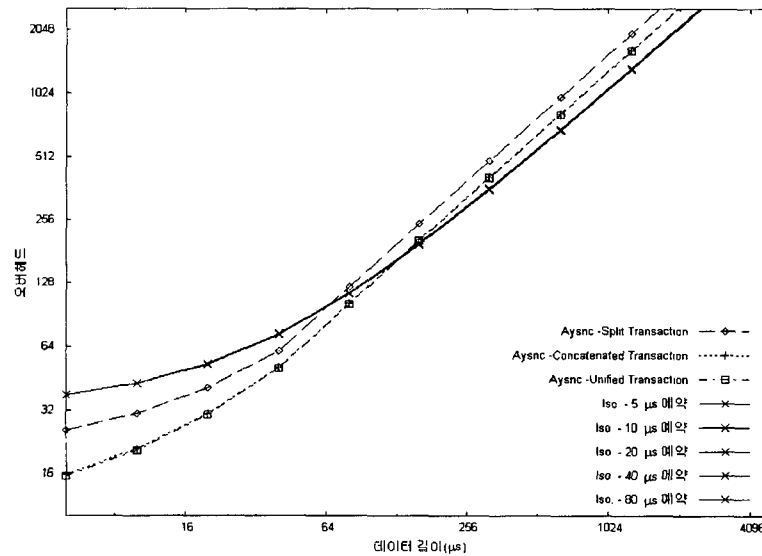


그림 9 비동기 전송과 등시성 전송의 오버헤드(400Mbps)

200Mbps, 400Mbps인 경우에도 100Mbps인 경우와 동일한 결과를 보임을 알 수 있다.

4. IEEE 1394 평균 버스 접근 지연시간 분석

본 장은 사용자가 느끼는 버스 성능을 분석하기 위해 응용 계층 수준에서 버스 성능과 각 전송 모드를 비교한다. 평가 기준으로 사용된 버스 접근 지연 시간은 하

나의 패킷을 전송하기 위해 사용자가 기다린 시간으로 응용 계층에서의 성능은 물론 전체 버스의 이용률을 반영한다.

앞에서 살펴 본 바와 같이 IEEE 1394가 여러 전송 서비스를 제공하고 있으나 기존 응용 프로그램이나 상위 계층은 IEEE 1394의 전송 모드를 고려하지 않고 만들어 졌기 때문에 상위 계층과 IEEE 1394 계층 사이에

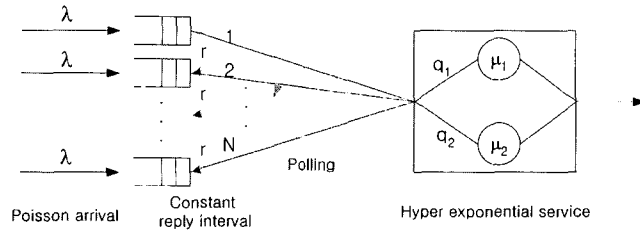


그림 10 폴링 시스템

서 적절한 1394의 전송 모드를 선택해 주는 인터페이스 계층이 필요하다[9]. 따라서 본 장에서는 대표적인 인터페이스인 고정 인터페이스와 적응 인터페이스를 선택하였다. 그리고 이들을 1394 계층과 상위 계층 사이에 적용한 두 가지 시스템을 제한적 폴링 시스템으로 구성하여 현실적이고 효과적으로 1394가 제공하는 응용 계층 수준의 성능을 분석하고 전송 모드를 비교한다.

우선 4.1절에서는 고정 인터페이스를 적용한 버스 시스템의 폴링 모델을 세우고 성능을 분석하였다. 여기에서 고정 인터페이스는 데이터 패킷을 최선형으로 서비스하는 비동기 전송 서비스를 상위 계층에 고정적으로 제공하는 방식이다. 따라서 4.1절에서는 모든 전송이 비동기 모드로만 이루어 질 경우에 대한 분석 모델을 구성하였다. 한편, 4.2절에서 분석한 적응적 인터페이스는 비동기 전송 모드를 기본 모드로 서비스를 제공하다가 적절한 조건을 만족시키면 등시성 모드로 전송을 전환하는 인터페이스 방식으로 4.1절의 폴링 시스템을 확장하여 분석할 수 있다.

특히 본 논문에서는 여러 형태의 적응적 인터페이스 중에서 데이터 길이 기반 방식을 사용하였다. 데이터 길이 기반 방식은 패킷의 크기가 S_{max} 이상인 전송이 n 회 이상 연속될 경우, 대량의 패킷 전송이 지속될 것으로 예측하여 등시성 모드로 전송 서비스를 전환하는 방식이다. 본 논문에서는 $n = 1$ 로 하고 등시성 모드의 초기 부하를 극복한 시점 이후를 분석 대상으로 하였다.

이어 4.3에서는 4.1절과 4.2절에서 보인 폴링 모델을 통해 분석된 결과와 버스를 시뮬레이션한 결과를 비교하고 IEEE 1394의 제한적 폴링 모델의 타당성을 보인다.

4.1 비동기 모드 전송만 있는 경우

4.1.1 시스템 모델링

2.1에 설명한 바와 같이 IEEE 1394 버스에서 비동기 모드로 데이터를 전송하는 경우, 각 노드는 공평구간마다 한 번 씩 아비트레이션에 의해 정해진 순서에 따라 버스에 접근할 수 있다. 따라서 IEEE 1394 버스는 비동기 전송을 하는 각 노드를 큐로 하는 폴링 시스템으

로 모델링 가능하다(그림 10). 분석을 쉽게 하기 위해서 IEEE 1394 망에 다음과 같은 제약을 가정하였다.

- 가정 1: 동일 노드 가정(Symmetric Stations)
각 노드는 모두 같은 행동을 한다.
- 가정 2: 무한 큐 가정(Infinite Queue)
각 노드가 갖는 버퍼의 크기는 무한대이다.
- 가정 3: 제한 서비스 가정(Limited Service)
서버는 각 노드를 돌며 한번에 하나의 메시지를 처리한다.

N 개의 노드로 구성된 시스템에서 각 노드의 트래픽은 도착률이 λ 인 포아송 프로세스를 따라 발생하고, 동일 노드 가정에 따라 모든 노드의 도착 프로세스는 동일하다. 이때 하나의 큐를 풀하여 서비스가 끝난 뒤 다음 큐를 풀하기 전까지 갖는 휴지 기간 r 을 두어 IEEE 1394에서 사용하는 각종 갭을 반영하도록 하였다. 소스에서 발생하는 데이터 길이는 하이퍼익스포넨셜(Hyper exponential) 분포를 따른다고 가정한다. 이는 인터넷에서 전송되는 패킷 수의 절반가량이 40~44bytes 길이의 제어 패킷들이며[10], IEEE 1394에서도 AP나 사이클 시작 패킷과 같은 짧은 패킷을 많이 사용하기 때문에 가능하다. 이들 제어 패킷들은 일반 사용자 데이터 패킷에 비해 평균 길이가 현저히 짧기 때문에 일반 사용자 데이터와 제어 데이터를 구분할 수 있는 하이퍼익스포넨셜 분포를 통해 패킷 길이의 특성을 모델에 반영하였다. 데이터 패킷과 제어 패킷의 발생 비율은 각각 q_1 , $q_2 = 1 - q_1$ 이다. 데이터 패킷의 평균 길이를 $1/\mu_1$, 제어 패킷의 평균 길이를 $1/\mu_2$ 이라고 하면 각 패킷의 처리율은 μ_1 과 μ_2 가 된다.

4.1.2 평균 버스 접근 지연시간 분석

본 논문에서 버스 접근 지연이란 임의의 노드에서 전송할 데이터가 발생하여 비동기 모드를 통해서 데이터를 보내기 시작하기까지 걸린 시간을 말한다. 그림 10에서 보인 폴링 시스템은 [11]에서 분석한 제한적 폴링 모델로 평균 접근 지연 시간을 다음과 같이 구할 수 있다.

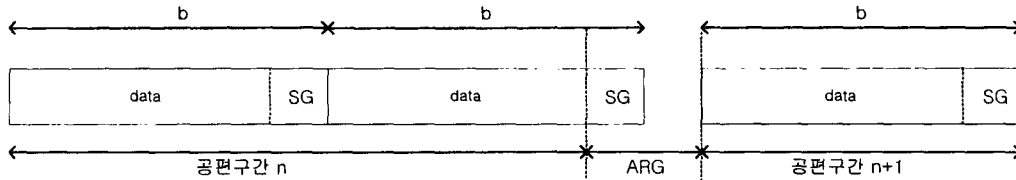


그림 11 SG과 ARG의 모델링

$$E[W] = \frac{\delta^2}{2r} + \frac{M[\lambda b^{(2)} + r(1 + \lambda b) + \lambda \delta^2]}{2[1 - M(r + b)]} \quad (6)$$

$B(s)$, b , $b^{(2)}$: 메시지 길이의 분포, 평균, 재조합

$R(s)$, r , δ^2 : 휴지기간의 분포, 평균, 분산

이때, 메시지 길이의 평균인 b 와 재조합 $b^{(2)}$ 은 전송할 데이터의 처리율이 μ_1 , μ_2 이고 발생 비율이 q_1 , q_2 인 하이퍼익스포넨셜 분포를 따르므로 다음과 같다. 이때, $B'(s)$ 은 $B(s)$ 의 LST(Laplace-Stieltjes Transform)이다.

$$B(s) = \frac{q_1 \mu_1}{\mu_1 + s} + \frac{q_2 \mu_2}{\mu_2 + s} \quad (7)$$

$$b = -B'(0) = \frac{q_1}{\mu_1} + \frac{q_2}{\mu_2} \quad (8)$$

$$b^{(2)} = B''(0) = \frac{q_1}{\mu_1^2} + \frac{q_2}{\mu_2^2} \quad (9)$$

그리고 r 은 다음 큐를 방문하는데 걸리는 시간인 휴지기간의 평균으로 SG와 ARG를 모델에 반영하기 위해 사용되었다. SG는 비동기 모드에서 패킷과 패킷을 구분하기 위한 공백이므로 각 패킷의 길이에 SG 만큼을 더하여 모델링한다. 그러나 공평 구간에서 마지막 전송한 패킷에는 공평구간을 구분하기 위한 공백인 ARG가 와야 하므로 $ARG - SG$ 만큼의 오차가 발생한다. ARG는 한 공평구간에서 한 번씩 발생하므로 동일 노드 가정에 따라 r 을 이용해 이 오차를 해결할 수 있다. 즉 그림 11과 같이 누락된 $(ARG - SG)$ 을 N 으로 나눠 r 로 사용한다.

$$r = \frac{ARG - SG}{N} \quad (10)$$

같은 방법으로 휴지기간을 이용하면 등시성 전송을 반영한 비동기 모드의 버스 접근 지연시간을 구할 수 있다. 등시성 모드를 고려하지 않을 경우 휴지기간은 상수이므로 $\delta^2 = 0$ 이다.

4.2 비동기 모드와 등시성 모드 전송이 함께 있는 경우

비동기 모드에서 대량의 데이터를 전송하게 되면 2장에서 보인 바와 같이 비동기 모드가 갖는 부하로 인해 비동기 모드에서의 응답 시간이 길어지고 전체 버스 이용률과 대역폭이 줄어들게 된다. 따라서 등시성 모드가

갖는 초기 부하를 극복할 수 있을 만큼 충분히 큰 벌크 데이터를 선별하기 위해, 패킷 크기가 최대 전송 패킷 크기와 같을 때 등시성 모드로 전송 모드를 전환하는 적응적 인터페이스를 적용할 수 있다.

인터넷 트래픽의 패킷 분포 중 대부분을 차지하는 것은 40bytes 대의 제어 패킷과 더불어 552/576bytes와 1500bytes의 사용자 데이터를 담은 큰 데이터 패킷들이다[10]. 552/576bytes 와 1500bytes는 각각 TCP 구현상 MTU 탐색을 안 하는 경우와 하는 경우에 따라 크기가 정해진 것으로, 대부분 TCP 응용 프로그램이 보낸 벌크 전송의 일부이다. IEEE 1394에서 이들 인터넷 최선형 서비스는 비동기 모드로 지원되므로, (버스 내의 각 노드에서 TCP가 MTU 탐색을 한다고 가정하면) 대역폭이 400Mbps인 버스의 비동기 모드에서 길이가 약 $29 \mu s (\frac{1}{\mu_1} = \frac{1500 \text{ bytes}}{400 \text{ Mbps}})$ 인 패킷들을 벌크 전송의 일부로 간주해 등시성 전송을 한다.

동일 노드 가정에 따라 등시성 전송을 위해 예약된 대역폭은 N 등분하여 r 에 반영한다.

$$r = \frac{FI_{iso} + ARG - SG + C}{N} \quad (11)$$

이 때, FI_{iso} 는 한 공평구간 안에 등시성 모드로 전송되는 데이터의 길이를 시간 단위로 나타낸 값으로 식 (12)와 같고, C 는 등시성 사이클로 인해 지연되는 오차를 반영하는 값으로 첨부-1과 같다.

$$FI_{iso} = \lceil \frac{FI_{asy}}{125 - iso} \rceil \times iso \quad (12)$$

여기에서 $b_2 = \frac{1}{\mu_2}$ 이고 비동기 모드로 보내지는 패킷의 평균 길이이다. 식 (12)에 쓰인 iso 는 한 cycle(125 μs)에서 등시성 전송을 위해 예약된 대역폭을 시간 단위로 나타낸 것이고, FI_{asy} 는 한 공평구간에서 등시성 전송 구간을 뺀 순수 비동기 전송에 드는 총 시간이다. iso 와 FI_{asy} 의 상세 식은 첨부-1과 같다.

FI_{iso} 는 상수가 아니므로 r 의 분산, δ^2 ,을 첨부-2와 같이 구할 수 있다.

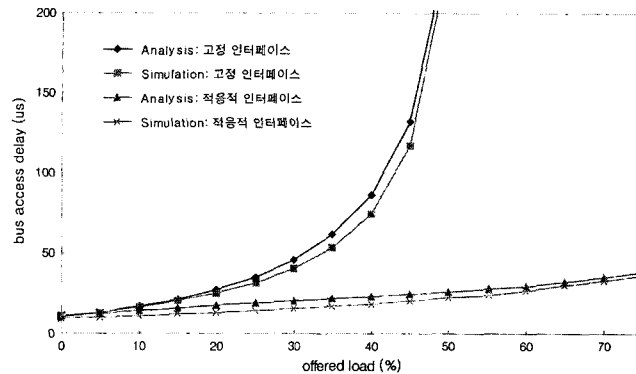


그림 12 버스 접근 지연 시간 비교

식 (12)와 첨부-2 따라 구한 r 과 δ^2 을 식 (6)에 적용하여 등시성 모드를 고려한 버스 접근 지연 시간을 구할 수 있다.

4.3 분석 결과

버스 접근 지연 시간을 식 (6)으로부터 구하면 버스 로드 에 따라 그림 12와 같은 결과를 얻을 수 있다. 이때, 조정 가능한 파라미터인 버스 속도는 400Mbps ($4 \times 98.304 \times 1024^2 \text{ bits/sec}$), $N = 10$, $q_1 = 0.5$, $q_2 = 0.5$ 이며, 등시성 연결 설정이 끝난 데이터 흐름을 대상으로 하였다. 모델의 타당성을 검증하기 위해 같은 환경에서 시뮬레이션을 수행하여, 데이터 전송을 위해 비동기 모드를 기본으로 등시성 모드를 선택적으로 채용한 적응적 인터페이스 모델과 비동기 모드 전송만을 사용한 고정 인터페이스, 두 경우 모두 폴링 시스템 분석 결과와 비슷한 시뮬레이션 결과를 얻었다. 그림 12가 보듯이 모든 데이터를 비동기 모드로 전송하는 것보다 최대 패킷 크기로 전송되는 데이터는 등시성 모드로 전송하고 나머지를 비동기 모드로 전송하는 적응적인 전송할 경우 버스 접근 시간이 짧아진다. 특히 로드가 증가할수록 성능 차가 심화되는데, 이는 비동기 모드에서 정의하는 공평구간이 등시성 모드의 사이클 크기인 $125 \mu\text{s}$ 에 비해 크고, 공평구간이 커지는 속도가 빨라지기 때문이다. 등시성 전송에서도 3번의 비동기 트랜잭션을 통해서 전송이 시작되므로 버스 접근 시간에서 성능 향상을 얻을 수 있다.

5. 결론

본 논문은 IEEE 1394가 제공하는 전송 모드의 오버헤드를 분석하고 각 모드의 효율을 비교하였다. 특히 폴링 시스템 분석과 시뮬레이션을 통해, 데이터를 전송하

기 위해 비동기 모드만을 사용할 때와, 비동기 모드와 등시성 모드를 적응적으로 선택할 경우를 비교하였다. 그 결과, 적응적 전송 정책을 쓰면 네트워크 로드가 증가함에 따라 버스 접근 시간이 줄어 대역폭 낭비를 막고 전체 버스 이용률이 높아지는 것을 알 수 있었다. 여기에서 비동기 모드로만 전송할 때 버스 접근 지연 시간이 커지는 이유는, 비동기 전송 모드가 사용하는 SG과 ARG의 크기가 등시성 모드에서 사용하는 IG에 비해 월등히 크기 때문이다. 즉, 비동기 모드의 과도한 사용을 줄이면 각종 갭(gap)으로 인해 낭비되는 대역폭을 줄일 수 있다. 따라서 적응적 전송 정책을 사용하면 버스트 전송으로 인해 버스의 성능이 떨어지는 것을 막고 응용 계층 이용자에게 더욱 빠르고 안정된 전송을 보장할 수 있다.

참고 문헌

- [1] D. Anderson, *FireWire System Architecture*, pp.11-16, Addison Wesley, Reading Mass., 1999.
- [2] "IEEE P1394.1 - High Performance Serial Bus Bridges Working Group," <http://grouper.ieee.org/groups/1394/1/index.html>
- [3] C. Severance, "FireWire Finally Comes Home," *IEEE Computer*, Vol 31 11, Nov. 1998, pp.117-118.
- [4] L. Ruiz, Ph. Dallemagne, "Using Firewire as Industrial Network," *SCCC*, pp.201-208, 1999.
- [5] T. Haibara, M. Tsubokawa, T. Sugawa, Y. Kuruusu, "An Evolutional Scenario for Home Network and Basic Trial of IEEE 1394 Home Network," *International Conference on Consumer Electronics*, pp.38-39, 1999.
- [6] P. Johansson, "IPv4 over IEEE 1394," *RFC2734*, December 1999.
- [7] G. Moon, H. Lim, C. Kim, "A Simple Channel Allocation Method for IP Multicast over IEEE-

1394," Electronics Letters Vol 364 17, Feb. 2000, pp.384 385.

- [8] D. Steinberg, Y. Birk, "An empirical analysis of the IEEE 1394 serial bus protocol," IEEE Micro, Vol 20 1, Jan. Feb. 2000, pp.58 65.
- [9] H. Lee, Y. Lee, S. Kang, G. Park, S. Lee, "An Adaptive Interface to Accomodate Internet and Data Traffic for IEEE 1394," International Conference on Consumer Electronics, pp.286 287, 1999.
- [10] K. Thompson, G. J. Miller, and R. Wilder, "Wide Area Internet Traffic Patterns and Characteristics," IEEE Network, November/December 1997, pp.10 23.
- [11] H. Takagi, *Analysis of Polling Systems*, pp.118 122, The MIT Press, 1986.

[첨부-1]

4.2절의 식 (11)에서 C는 새 등시성 사이클이 시작할 때, 지난 사이클의 마지막 SG과 RG이 중단되면서 생기는 지연 오차를 반영하는 값이다. 이는 보다 정확한 r을 구하기 위해 필요한데, 버스 접근 시간이 현저히 줄어들어 r의 차지하는 비중이 커졌기 때문이다.

$$C = \frac{SG}{b_2 \times \left\lfloor \frac{125 - iso}{b_2} \right\rfloor} \times \frac{SG}{2} \times (K - 1) + \frac{RG}{FI_{asy}} \times \frac{RG}{2} \quad (a)$$

$$K = \left\lfloor \frac{FI_{asy}}{125 - iso} \right\rfloor \quad (b)$$

또한 FI_{iso} 는 한 공평구간 안에 등시성 모드로 전송되는 데이터의 길이를 시간 단위로 나타낸 것으로 다음과 같다.

$$FI_{iso} = K \times iso \quad (c)$$

(a)와 (c)에서 iso는 한 cycle(125 μs)에서 등시성 전송을 위해 예약된 대역폭을 시간 단위로 나타낸 것이고, FI_{asy} 를 비동기 전송만을 고려한 공평구간의 길이이다.

$$iso = \frac{q_1 \times \lambda \times 125}{\mu_1} \quad (d)$$

$$FI_{asy} = \sum_{r=1}^N \left\{ N C_r \times p_{empty}^N \times (p_{empty} - 1)^r \times (r \times b + (ARG - SG)) \right\} + p_{empty}^N \times \frac{ARG}{2} \quad (e)$$

p_{empty} 는 폴링 시스템에서 서비스를 하기 위해 방문한 큐가 비어 있을 확률이다[11].

$$p_{empty} = \frac{1 - N \times \lambda \times (r + b)}{1 - N \times \lambda \times b} \quad (f)$$

이때,

$$\lambda = \lambda \times q_2,$$

$$r = \frac{ARG - SG}{N},$$

$$b = \frac{1}{\mu_2} + SG \text{이다.}$$

[첨부-2]

$\delta^2 = E[X^2] - E[X]^2$ 이므로

$$E[X] = \frac{FI_{iso}}{N} + \frac{ARG - SG + C}{N} \quad (a)$$

일 때, 다음과 같이 $E[X^2]$ 을 구하면 분산을 구할 수 있다.

$$E[X^2] = \frac{FI_{iso}^2}{N^2} + 2 \left\{ \frac{FI_{iso}}{N} \times Const \right\} + Const^2 \quad (b)$$

단, $Const = \frac{ARG - SG + C}{N}$ 이고,

iso가 상수이므로 (첨부 1), FI_{iso} 는 다음과 같다.

$$FI_{iso} = \left\lfloor \frac{FI_{asy}}{(125 - iso)^2} \right\rfloor \times iso^2 \quad (c)$$

그리고 FI_{asy} 는 다음과 같이 구할 수 있다.

$$FI_{asy} = \sum_{r=1}^N \left\{ N C_r \times p_{empty}^N \times (p_{empty} - 1)^r \times (r \times b + (ARG - SG))^2 \right\} + p_{empty}^N \times \left(\frac{ARG}{2} \right)^2 \quad (d)$$

여기에서 p_{empty} 는 방문한 큐가 비어 있을 확률이므로 첨부-1의 (f)와 같다.



이 희 진

1997년 광운대학교 전자계산학과(학사)
1997년~2000년 삼성전자 중앙연구소 연구원.
2002년 서울대학교 전기·컴퓨터공학부(석사).
2002년~현재 서울대학교 전기·컴퓨터공학부 박사과정



민 구 봉

1999년 서울대학교 전산학과(학사)
2001년 서울대학교 전기·컴퓨터공학부(석사).
2001년~현재 LG전자 주임연구원



김 종 권

1981년 서울대학교 공과대학 산업공학과(학사).
1982년 미국 조지아 공대(석사)
1987년 미국 일리노이 대학(박사).
1987년~1991년 미국 벨 통신 연구소 연구원.
1991년~현재 서울대학교 전기·컴퓨터공학부 교수