


 사례  
발표

# MOST<sup>TM</sup> 제어 채널의 대기 지연 시간 성능 분석

## 목 차

1. 서 론
2. MOSTM 기술 요약
3. MOSTM 제어 채널 모델링
4. 성능 평가
5. 결 론

박부식 · 정한균 · 신대교 · 임기택  
최종찬 · 윤종호  
(전자부품연구원 · 한국항공대학교)

## 1. 서 론

1980년대 후반 최초로 차량 내 장치들을 제어하기 위해서 네트워크가 구성된 이래, 차량에는 다양한 형태의 네트워크가 적용되어 그들의 가치를 돋보이게 하는 역할을 수행하였다. 1983년 Robert Bosch GmbH 에 의해 개발이 시작된 CAN(Controller-Area Network)과 1999년에 1.1 버전이 출판된 방송 형태의 직렬 통신 방식인 LIN(Local Interconnect Network) 등이 차량 내 장치들을 연결하는 대표적인 네트워크 프로토콜들이다.

유,무선 통신 기술이 발달되고 멀티미디어 기기의 보급이 늘어나면서 대용량 데이터 전송에 대한 요구가 늘어났다[1]. 하지만 차량 제어에 바탕을 둔 CAN 및 LIN 프로토콜은 오디오 및 비디오 데이터를 전송하기 위해서는 대역폭이 부족하였고 뿐만 아니라 멀티미디어 응용 계층에 대한 고려가 전무하다. 이런 문제점을 해결하기 위해서, BMW, Becker (현재는 Harman/Becker), 그리고 OASIS Silicon Systems (현재는 SMSC)는 1996년 D2B 시스템을 기반으로

MOST<sup>TM</sup> 개발을 시작하였고, 25 Mb/s의 MOST25 및 50 Mb/s의 MOST50이 상용화되었고, 2008년 초에 150Mb/s 를 지원하는 MOST 150을 발표하였다.

차량이 운행하기 위해서는 다양한 기능이 요구된다. 아래 표는 현재 널리 사용되는 차량 통신 프로토콜들의 기능 별 특성을 정리한 것으로 각 프로토콜의 적용 분야를 짐작할 수 있으며, MOST<sup>TM</sup>가 인포테인먼트(Infotainment) 및 텔레매틱스 분야에서 상대적 우월한 특성을 갖음을 확인할 수 있다[2]. 물론 현재 대부분의 유럽 자동차 생산회사들은 MOST<sup>TM</sup>를 멀티미디어 통신 매체로 채택하고 있으며, 국내 유력 자동차 생산회사 두 곳 또한 MOST<sup>TM</sup>를 장착한 차량을 2008년도에 출시하여 MOST<sup>TM</sup>가 차량의 멀티미디어 데이터 네트워크 프로토콜로 유력함을 예상할 수 있다.

본 서론에 이어, 2 장에서는 MOST<sup>TM</sup> 기술을 소개하고 3 장에서는 MOST<sup>TM</sup> 제어 채널의 대기 지연 시간 성능 해석을 위한 모델을 제안하고 4 장에서는 해석된 모델과 시뮬레이션 내용을

〈표 1〉 통신 방식 별 동작 영역 적합성 [2]

네트워크	패쇄형 제어	개방형 제어	고장 진단	인포테인먼트 & 텔레매틱스	최대 속도
CAN	★★★	★★★	★★★		1 Mb/s
LIN		★★★	★		20 Kb/s
MOST™			★	★★★	150 Mb/s
FlexRay	★★★		★		10 Mb/s
Bluetooth			★	★★	3 Mb/s

비교 분석한다. 마지막으로 5 장에서 결론을 맺겠다.

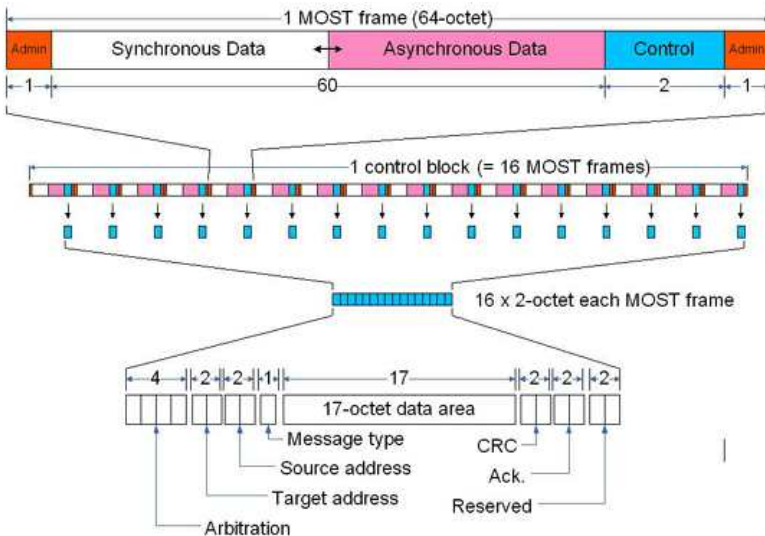
## 2. MOST™ 기술 요약

MOST™ 는 차량 내 난환경을 극복하기 위해서 광 통신 방식을 채택하였고, 비좁은 차량 내부에서 배선을 해야 하기 때문에 유연하게 굽어지는 POF(Plastic Optical Fiber) 케이블을 사용한다. 또한 차량 내 배선의 무게를 줄이고 안정적인 데이터 전송을 보장하기 위해서 원형 구조의 형태를 갖는다.

기본적으로 통신을 위해서 네트워크 마스터에 해당하는 장치가 클럭 정보가 포함된 데이터 신호를 전송하면 원형의 네트워크에 연결된 모든 장치들은 클럭을 추출하여 동기를 맞춰 통신을

준비한다. 마스터 장치는 아래 (그림 1)과 같은 형태의 수퍼 프레임을 44.1 KHz (혹은 48 KHz) 주기로 전송하고 모든 장치들은 이와 같은 규격대로 데이터를 송, 수신해야 한다. 크게 3 가지 종류의 데이터를 전송하기 위한 영역이 제공되고 비동기 영역과 제어 영역은 기본적으로 경쟁 방식을 통해 접근하고, 동기 영역은 예약 방식을 통해 경쟁 없이 접근할 수 있다. 그러므로, QoS 가 요구되는 오디오 및 비디오 데이터는 동기 영역에 실리고, QoS 가 불필요한 제어 메시지 혹은 인터넷 데이터 등은 제어 영역 혹은 비동기 영역을 이용하여 전송된다[2][3].

MOST™ 제어 채널은 채널 대역폭의 과도한 점유를 방지하기 위해서 64 바이트의 수퍼프레임마다 2 바이트씩 쪼개어져 전송된다. 16개의 2 바이트 메시지 조각이 모두 수신되면 하나의 32 바이트 제어 메시지가 완성된다. 제어 메시지 조각 중 첫 번째에 해당하는 메시지를 인식하기 위해서 첫 번째 수퍼프레임의 프리앰블은 특별한 비트 패턴으로 구성된다. 모든 제어 메시지는 32 바이트의 고정 길이를 갖으며, (그림 1) 처럼 Arbitration, 송수신 주소, 메시지 종류, 메시지



(그림 1) MOST™ 수퍼프레임 및 제어 메시지 구조

데이터, CRC, 수신확인 영역으로 구성된다. 제어 메시지를 전송하고자 하는 장치는 Arbitration 영역을 보고 메시지 영역이 비어 있는지 파악한 후, 비어 있을 경우 자신의 메시지를 실어 보낼 수 있다.

제어 채널을 통해서 크게 2 가지 서비스가 제공된다. 일반 응용 프로그램에서 사용되는 CMS (Control Message Service)라 불리는 메시지 전송 서비스와 타이밍 마스트에 의해서 수행되는 네트워크 자원 관리를 위한 시스템 메시지 전송 서비스가 함께 제공된다. CMS는 볼륨 값과 같은 장치 속성 변경, DVD 재생과 같은 기능 블록 시작과 결과 요청과 같은 일반적인 메시지를 전달하는 서비스이다. 또한, 동기 채널을 사용하기 위해서는 채널 예약 요청 메시지를 제어 채널을 통해 네트워크 마스터에게 전달해야 한다. 제어 채널은 MOST<sup>TM</sup> 네트워크가 동작하기 위한 기반이 되는 채널로서 중요한 부분임에 틀림없다.

### 3. MOST<sup>TM</sup> 제어 채널 모델링

MOST<sup>TM</sup>네트워크는 기본적으로 원형 형태를 갖고, 64 바이트 수퍼프레임 중 2 바이트 씩 나누어서 전송된다. 그리고, 제어 메시지의 시작을 알리는 데이터는 매 수퍼프레임의 제어 영역에 표시된다. 제어 채널을 모델링하기 위해서 우리는 MOST<sup>TM</sup> 제어 채널을 64 바이트의 슬롯 크기를 갖는 슬롯 기반의 원형 네트워크와 같다고 가정한다. 예를 들어, 44.1 KHz의 수퍼프레임 전송률을 갖는 경우라면, 768 Kb/s의 링크 대역폭을 갖는 64 바이트의 슬롯으로 구분된 단방향 링크가 서로 원형으로 연결된 네트워크라고 가정할 수 있다.

MOST<sup>TM</sup> 동기 채널의 예약을 위해서 제어 채널이 사용되지만, 한번 예약이 성공할 경우에는 무경쟁, 무충돌, 고정 지연의 속성으로 동기식 채널을 사용할 수 있으므로 동기식 채널은 제어 채널에 영향을 미치지 않는다고 가정할 수 있다.

반면에, 제어 채널을 통해 제공되는 CMS는 제어 채널 뿐만 아니라 비동기 채널을 통해서도 제공될 수 있다. 상황에 따라 비동기 채널을 통해서 더 많은 대역폭을 CMS를 위해 사용하도록 시도할 수 있으나, 본 논문에서는 비동기 채널은 임의의 길이를 갖는 인터넷 데이터를 위해 사용되는 영역으로 제어 채널과 서로 독립적인 채널임을 가정한다.

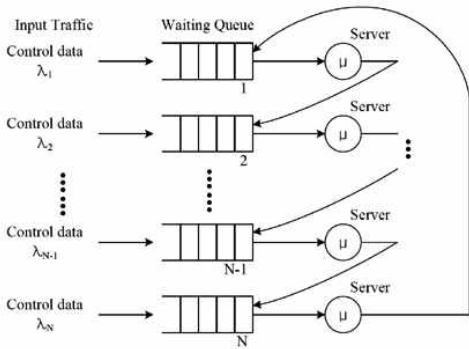
MOST<sup>TM</sup> 네트워크에 연결된 각 장치는 각자 독립적으로 트래픽을 발생시킨다. 이 트래픽은 대부분 제어 메시지가므로, 메시지의 길이는 짧을 것이다. 제어 채널이 기본적으로 32바이트의 고정 길이의 메시지 형식을 따르므로 각 장치는 32 바이트의 고정 길이의 트래픽을 exponential 분포를 갖는 간격으로 발생시킨다고 가정한다.

MOST<sup>TM</sup> 제어 채널은 각 장치에서 발생된 메시지를 다음 인접 장치로 전송해야 한다. 이때 32 바이트의 고정 길이의 메시지를 전송하므로 각 메시지의 서비스 시간은 항상 일정하다. 또한, 각 장치에서 발생하는 프로세싱 지연 시간은 무시하며, 장치 간 메시지 전송 시 링크에서는 어떤 메시지 에러도 발생하지 않는다고 가정한다.

위와 같은 가정을 통해서 원형의 MOST<sup>TM</sup> 네트워크는 (그림 2)와 같이 multi-queue와 multi-server 모델로 정의할 수 있다. N개의 장치는 각각  $\lambda_1, \lambda_2, \dots, \lambda_N$ 의 속도로 메시지를 발생시킨다. 그리고 이렇게 발생된 메시지는 각각 큐에 들어가는데 이 큐에서 서버에서 서비스 받을 때까지 기다리게 된다. 그리고 서비스 받은 메시지는 다음 장치의 서버로부터 서비스를 받게 된다. 인접 장치가 메시지를 수신하는 것을 MOST<sup>TM</sup> 제어 채널의 대역폭을 갖는 서버로 간주할 수 있다. 각 장치에 부여된 이 큐의 길이는 무한개의 메시지를 저장할 수 있다고 가정하겠다.

물론 토큰 링(Token ring)과 같은 multi-queue 모델로 정의할 수도 있겠지만, MOST<sup>TM</sup> 제어 채널의 경우에는 토큰 방식과는 다른 방식

을 따른다. 동시에 하나의 토큰이 존재하는 토큰 링과는 달리 MOST™의 경우에는 망에 여러 개의 토큰이 존재하는 경우와 동일하다. 예를 들어, 제어 메시지가 실린 16 개의 수퍼프레임이 지나가고 다음 17 번째 수퍼프레임에 제어 메시지가 실리지 않을 때, 이전 16 개의 수퍼프레임이 망을 따라 전송되고 있음에도 불구하고 해당 제어 채널에 메시지를 실을 수가 있다[4][5].



(그림 2) MOST™ 제어 채널 모델링

(그림 2)와 같은 모델은 한 장치에서만 관찰했을 때는 (그림 3)과 같은 모델로 재구성할 수 있다. 메시지는 장치에 의해서 생성되는데, 한 장치에서 관찰했을 때, 장치 스스로가 발생시키는 메시지와 다른 장치들이 발생시키는 메시지가 함께 입력으로 들어오게 된다. 그러므로 장치의 개수만큼의 해당하는 메시지가 발생되고 MOST™ 제어 채널 대역폭의 서버를 갖는 M/D/1 모델로 정의할 수 있다. 다만, 원형 구조의 특성 상 송신 채널과 수신 채널을 독립적으로 갖고 있기 때문에, 제어 채널의 물리적 대역폭의 2 배의 대역폭을 실제로 사용 가능하다. 그러므로 해석에 있어서 서버의 용량을 2 배로 늘려 계산해야 한다.



(그림 3) MOST™ 제어 채널 약식 모델링

### 4. 성능 평가

본 장에서는 앞 장에서 정의한 MOST™ 제어 채널 모델을 이용하여, 메시지가 서버로부터 서비스 받기 까지 평균적으로 얼마의 지연 시간이 소요되는지 수학적으로 계산하고, 이 계산 값과 시뮬레이션 결과를 비교 분석하였다.

수학적 해석을 위해서 다음과 같은 심볼이 사용된다.

- $\lambda$  평균 메시지 발생률
- $h$  평균 서비스 시간
- $\rho$  제공된 로드
- $W$  평균 지연 시간

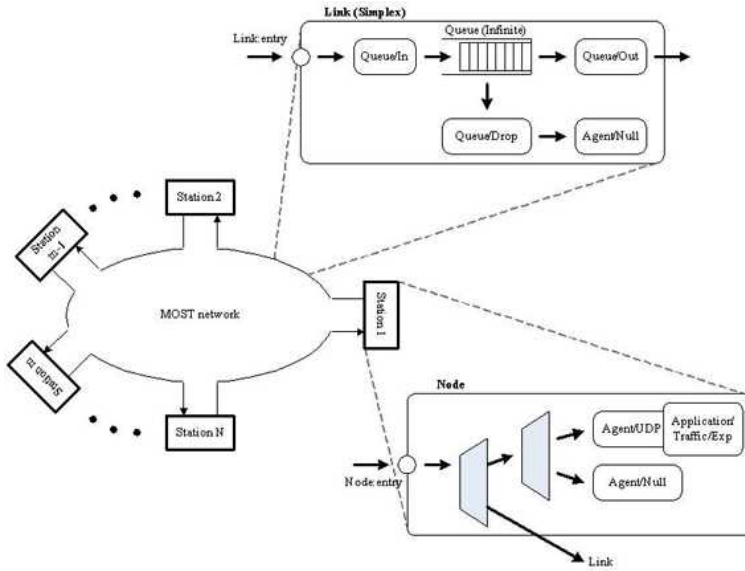
각 장치에서 메시지가 서비스 받을 때까지 소요되는 평균 지연 시간은 아래와 같다[6][7].

$$W = \frac{\rho \cdot h}{2 \cdot (1 - \rho)} \dots \dots \dots (1)$$

수학적 모델과 시뮬레이션 결과를 비교하기 위해서 MOST™ 제어 채널을 다음과 같이 시뮬레이션 하였다. 시뮬레이션을 위해 사용한 툴은 NS-2 2.33 버전을 사용하였다 [8].

시뮬레이션을 위한 토폴로지는 (그림 3)과 같이 원형으로 연결되며, 각각의 노드는 단방향 링크로 연결된다. 연결에 사용된 Link에 포함된 Queue 클래스의 최대 큐 길이값은 4바이트의 최대 정수의 값을 설정하여 사실 상 무한 큐를 구현하였다.

각 노드에는 Agent/Null 에이전트가 붙어 있다. 이 에이전트는 패킷 수신을 위한 것이다. 송신 용 에이전트는 각 노드에 9 개의 Agent/UDP 에이전트를 붙였다. 각 에이전트는 해당 목적지 노드의 Null 에이전트와 연결되어 패킷을 전송한다. 모든 Agent/UDP에는 트래픽 생성을 위해서 Application/Traffic/Exp 트래픽 생성기가 붙는다. Application/Traffic/Exp 는 휴먼 시간이 없이 exponential 분포를 갖는 시간 간격마



(그림 4) 시뮬레이션 모델

다 패킷을 발생시키는 개체로 NS-2 2.33 에서 기본적으로 제공하는 Application/Traffic/CBR 을 바탕으로 수정하였다.

시뮬레이션을 위해서 총 10개의 노드가 연결 되고, 각 노드에는 Null 에이전트와 9개의 UDP 에이전트가 붙어 있으며, 모든 UDP 에이전트에는 Application/Traffic/Exp 트래픽 발생기가 동작한다.

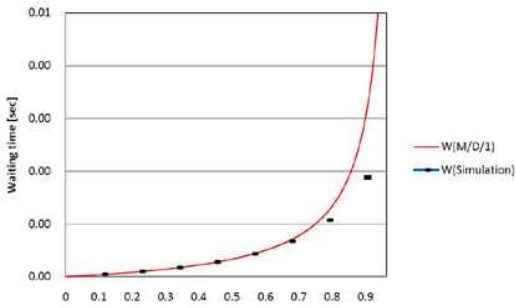
시뮬레이션을 위해서 모든 에이전트는 100 초 동안 패킷을 발생시킨다. 시뮬레이션 결과를 추출하기 위해서 앞부분 5 초에 해당하는 패킷은 initial bias와 로 성능 데이터를 계산하는 데는 포함시키지 않았고, 패킷 발생이 중단되기 5 초 전까지 만을 유효한 데이터로 계산하였다.

시뮬레이션 결과는 수학적 모델과 같이 지연 시간이 부하가 증가함에 따라 대기 큐에서 기다리는 시간이 함께 증가함을 확인할 수 있었지만, 부하가 0.8 과 0.9 의 경우에는 시뮬레이션 값이 수학적 모델의 결과와 차이를 보인다.

부하가 0.112 일 경우의 큐잉 지연 시간은 수학적 해석에 의한 결과에 의해 4.2042E-05 의 값

을 갖고 시뮬레이션 결과에 의해 4.5E-05 의 값을 가져 0.004%의 오차를 보였다. 또한 부하가 0.675 경우에서는 수학적 해석의 결과 값과 시뮬레이션 결과 값이 각각 6.92308E-4와 6.71E-4 로서 0.06% 오차를 보였다. 하지만, 0.9 의 부하를 갖을 경우에는 각각 0.003 과 0.00188 의 값으로 0.29%의 오차를 보이며 시뮬레이션의 큐 대기 시간이 좀 더 작게 나타났다.

높은 부하에서 이와 같이 큐잉 지연 시간이 수학적 모델보다 작게 나오는 이유는 원형 구조 때문인 것으로 추정된다. 모든 노드는 자신을 제외한 모든 노드에게 패킷을 발생시켜서 시뮬레이션 하였다. 그럼 제일 긴 지연 시간을 경험하게 되는 패킷은 바로 송신 노드의 바로 뒤에 위치한 노드가 목적지인 패킷일 것이다. 그러므로 시뮬레이션 상에서 한 노드가 패킷을 발생시켰을 때, 자신의 수신 용 링크는 사용하지 못하는 경우가 발생하게 된다. 결국 전체 네트워크의 혼잡도가 그 만큼 상대적으로 줄어들게 되어 부하가 많은 경우에 있어서 지연 시간이 낮게 나오는 결과를 초래하게 된다.



(그림 5) 결과 그래프

## 5. 결론

MOST™는 현재 유럽에서 주도적으로 상용화되고 있고, 국내와 북미를 비롯한 다른 곳에서는 아직 널리 보급된 상황은 아니다. 하지만, 국내 자동차 제조사들이 MOST™를 차량용 멀티미디어 네트워크로 결정하였고, 2 개의 차종에 적용되어 상용화되었다. 향후 MOST™가 장착된 차량은 점점 늘어날 것으로 보인다.

이에 따라 본 논문에서는 MOST™ 네트워크의 관리 및 제어 메시지를 담당하는 제어 채널을 분석하기 위한 모델을 제시하고 이 모델의 수학적 분석 방법을 제시하였다. 또한, MOST™ 제어 채널만을 시뮬레이션할 수 있는 환경을 개발하여 그 결과를 비교 분석하였다.

시뮬레이션 결과는 높은 부하의 경우에는 원형 구조의 네트워크 형상 때문에 수학적 모델과 차이를 보였지만, 0.7 이상의 부하를 가해서 나오는 큐잉 지연 시간은 실제 구현시 유효하지 않는 범위이므로 0.7 이하의 부하에서 유사한 결과가 도출된 것은 의미있는 것으로 판단된다. 따라서 본 논문에서 제시하는 분석 방법론과 시뮬레이션 결과는 MOST™ 시스템의 제어 채널의 동작을 분석하는데 유용할 것으로 판단된다.

MOST™가 점점 차량 내 멀티미디어 네트워크 솔루션으로 보급되면 될 수록 MOST 네트워크를 분석하는 방법에 대한 요구가 증가될 것이고, MOST™ 장치 개발자들은 자신들의 시스템

을 시뮬레이션해보고 싶어 할 것이다. 본 논문은 MOST™ 네트워크 중 제어 채널을 분석하는 방법을 제시하고, 이를 통해 좀 더 신뢰성이 높은 MOST™ 장치 개발이 가능할 수 있으리라 기대한다.

## 참고문헌

- [1] M. Weber and J. Weisbrod, Requiriements Engineering in Automotive Development: Experiences and Challenges, IEEE J. of Software, Vol. 20, Issue 1, pp. 16-24, 2003.
- [2] MOST: The Automotive Multimedia Network, A. Grzember, Franzis Verlag GmbH, ISBN 978-3-7723-5316-1, 2008.
- [3] MOST Cooperation, MOST Specification Rev2.5. <http://www.mostcooperation.com>, 2006.
- [4] H. Akimaru and K. Kawashima, Teletraffic - Theory and Applications, ISBN 0-387-19805-9, Springer-Verlag, 1993.
- [5] ISO 8802/5 Token ring local area networks.
- [6] L. Kleinrock, Queueing Systems - Vol. I: Thoery, ISBN 0-471-49110-1, John Wiley & Sons, Inc., 1975.
- [7] 이호우, 대기행렬 이론 - 확률과정론적 분석, ISBN 89-85922-61-0, 시그마프레스, 1998.
- [8] The Network Simulator - ns-2, <http://www.isi.edu/nsnam/ns>.

## 저자약력



**박부식**

1999년 한국항공대학교 통신정보공학과(학사)  
2001년 한국항공대학교 정보통신공학과(석사)  
2007년~현재 한국항공대학교 정보통신공학과(박사과정)  
2002년~현재 전자부품연구원 선임연구원  
관심분야 : 유무선 통신 MAC, ITS, DSRC, AFDX  
이 메 일 : [parksik@keti.kr](mailto:parksik@keti.kr)



**임기택**

1994년 한양대학교 전자공학과(학사)  
1996년 한양대학교 전자공학과(석사)  
1996년~현재 전자부품연구원 책임연구원  
관심분야 : 지능형자동차, ITS, 엔터테인먼트용 SoC 및 시스템  
이 메 일 : [limkt@keti.kr](mailto:limkt@keti.kr)



**정한군**

2005년 한국항공대학교 정보통신공학과(학사)  
2007년 한국항공대학교 정보공학(석사)  
2008년~현재 전자부품연구원 전임연구원  
관심분야 : 유무선, 네트워크 시스템  
이 메 일 : [junghg@keti.kr](mailto:junghg@keti.kr)



**칙종안**

1985년 경희대학교 전자공학과(학사)  
2002년 서경대학교 컴퓨터과학과(석사)  
2005년 충북대학교 정보통신공학과(박사)  
1985년~1990년 삼성 SDI 종합연구소 주임연구원  
1992년~현재 전자부품연구원 수석연구원, SoC연구센터 센터장  
관심분야 : VLSI 설계, SoC 플랫폼  
이 메 일 : [chojic@keti.kr](mailto:chojic@keti.kr)



**신대교**

1998년 아주대학교 전자공학과(학사)  
2000년 아주대학교 전자공학과(석사)  
2000년~2003년 (주)이엠티티  
2003년~현재 전자부품연구원 선임연구원  
관심분야 : SoC설계, 디지털 모뎀 설계  
이 메 일 : [dukeshin@keti.kr](mailto:dukeshin@keti.kr)



**윤종호**

1984년 한양대학교 전자공학과(학사)  
1986년 KAIST 전기및전자공학과(석사)  
1990년 KAIST 전기및전자공학과(박사)  
1995년~1996년 U. of Arizona 방문교수  
1992년~현재 한국항공대학교 항공전자정보통신공학부 정교수  
관심분야 : 이더넷, 프로토콜, AFDX  
이 메 일 : [yunch@kau.ac.kr](mailto:yunch@kau.ac.kr)