

계층간 최적화를 위해 비디오 트레이스에 기반한 무선 MAC 스케줄러

論 文

55D-5-7

A Wireless MAC Scheduler Based on Video Traces for Cross-Layer Optimization

片基鉉[†] · 趙成翊^{*}
(Kihyun Pyun · Seong-Ik Cho)

Abstract - A wireless MAC scheduler that provides a high level of quality-of-service (QoS) for video-on-demand (VOD) applications while achieving a reasonable level of system throughput is proposed. The proposed scheduler considers both channel qualities of mobiles and the urgency of real-time packets coming from VOD applications in a cross-layer approach between application and MAC layers.

Key Words : Quality-of-Service, Wireless Packet Scheduling, Cross-Layer Optimization

1. 서 론

점진적으로, 모바일(mobile) 사용자들은 비실시간 서비스에 대해서 높은 수준의 시스템 성능을 요구할 뿐만 아니라 실시간 서비스에 대해서도 고품질을 요구한다. CDMA2000 1xEV-DO[1]나 와이브로와 같은 진보된 무선 시스템은 모바일의 채널 품질을 고려함으로써 고성능 시스템을 성취한다. 그러나 이런 시스템들은 여전히 주문형 비디오 (VOD: Video-on-demand) 응용과 같이 실시간 응용의 QoS (Quality-of-service)를 만족시키는데 어려움을 갖는다.

현존하는 무선 스케줄러들은 VOD 응용의 QoS를 만족시키는데 부족하거나 혹은 고수준의 시스템 성능을 성취하는데 부족하다. 예를 들어 잘 알려진 PF (proportional fair) 스케줄러는 공평한 방식으로 고수준의 시스템 성능을 성취할 수 있다 [2]. 그러나 PF 스케줄러는 VOD 응용의 QoS를 만족시키는데 불충분하다. 왜냐하면 PF 스케줄러는 비디오 프레임 전달하는 실시간 패킷의 긴급성을 전혀 고려하지 않기 때문이다. 특히 PF 스케줄러는 비디오 프레임 간 크기 차가 큰 VBR (Variable-bit-rate) 비디오 데이터를 전송할 때 실시간 패킷의 시간 요구를 종종 위반할 수 있다. VBR 비디오는 고품질 서비스 제공을 위해서 무선 시스템에서도 점차 중요해지고 있다. 반면 SP (Static priority) 스케줄러[3]는 항상 실시간 패킷에게 비실시간 패킷보다 높은 우선순위를 준다. 이 방식은 VOD 응용의 QoS를 쉽게 만족시키는 장점을 갖지만 모바일의 채널 품질을 전혀 고려하지 않기 때문에 시스템 성능이 현저히 떨어질 수 있다.

이 논문은 VOD 응용에 대해서 합리적 수준의 시스템 성능을 성취하면서도 고품질을 제공하는 MAC 스케줄러를 제안한다. 이 목적을 위해 제안하는 스케줄러는 [4]에 소개된 계층간 최적화 기법을 기반으로 응용과 MAC 계층간에 활용하여, VOD 응용으로부터 전송되는 실시간 패킷의 긴급성과 모바일의 채널 품질을 함께 고려한다. 시뮬레이션 결과는 제안하는 스케줄러가 PF나 SP 스케줄러에 비해 각각 QoS와 시스템 성능 측면에서 더 뛰어날 수 있음을 보인다.

2. 시스템 모델과 문제 설명

전체 시스템은 전통적인 무선 VOD 시스템과 비슷하게, 비디오 서버, 기지국, 그리고 모바일로 구성된다. 각 모바일 i 는 비디오 데이터를 저장하는 고정 크기 버퍼 B_i 를 갖는다. 초기에 모바일 i 는 버퍼 B_i 에 축적된 데이터양이, 재생 시작 위치라 불리는 임계값 s_i 를 넘을 때까지 비디오를 재생하지 않는다. 축적된 데이터양이 s_i 를 넘을 때 모바일 i 는 버퍼 B_i 에 쌓인 데이터를 소모하면서 비디오 재생을 시작한다. 이 버퍼링 기법은 도착하는 비디오 데이터 비율이 순간적으로 소모되는 비율보다 적다할지라도 모바일의 계속적인 비디오 재생을 가능하게 만든다. 그러나 만일 버퍼 B_i 가 비게 되면 비디오 재생은 계속될 수 없다. 이 경우 모바일은 비디오 재생을 중단하고, 다시 버퍼 B_i 에 축적된 데이터양이 임계값 s_i 를 넘을 때까지 버퍼링을 재시작 한다. 우리는 모바일이 데이터를 소모하지 않고 버퍼링만 하는 경우를 모바일이 버퍼링 기간 내에 있다고 말한다. 그렇지 않은 경우, 즉, 모바일이 비디오 데이터를 소모함으로써 비디오 재생을 수행하는 경우 모바일이 재생 기간 내에 있다고 말한다. 주목할 점은 모바일 i 의 행위는 두 인자, 즉, B_i 의 값과 s_i 의 값에 의해서 특성화 된다는 점이다.

비디오 서버는 비디오 데이터 파일을 저장한다. 또한 비디오 서버는 각 비디오 파일에 대응되는 비디오 트레이스 파일

[†] 교신저자, 正會員 : 全北大 工大 電磁情報工學部 助教授 · 工博
E-mail : khpyun@chonbuk.ac.kr

^{*} 正會員 : 全北大 工大 電磁情報工學部 助教授 · 工博
接受日字 : 2006年 3月 31日
最終完了 : 2006年 4月 16日

(video trace file)을 저장한다. 비디오 트레이스 파일은 초당 비디오 프레임 수 (fps : frames per second)를 포함하는 헤더와 비디오 프레임 크기들의 나열로 구성된다. 비디오 트레이스 파일은 대개 실제 MPEG 비디오 대체하는 시뮬레이션 실험을 수행할 때 사용된다. 그러나 이 논문은 계층간 최적화 관점에서 비디오 트레이스 데이터를 활용한다.

제안하는 시스템에서 어떻게 VOD 연결이 설정되고 시작되는지를 살펴보자. 모발 i 가 비디오를 시청하기를 원한다고 가정하자. 이 경우 모발 i 는 기지국에 요청 메시지를 보낸다. 요청 메시지에 덧붙여서, 모발 i 는 행위 인자 B_i 의 값과 s_i 의 값을 기지국에 보낸다. 이 인자들은, 이 논문의 뒤쪽에 기술했듯이, 기지국내의 스케줄러가 모발이 재생 기간인지 혹은 버퍼링 기간인지를 추정하여 판단하는데 사용된다. 기지국은 비디오 프록시 서버 (video proxy server)의 역할을 해서 요청 메시지를 비디오 서버에게 릴레이 한다. 만일 비디오 서버가 그 요청을 승인할 수 있으면, 승인 메시지를 보내기 이전에, 해당 비디오 트레이스 파일을 기지국에 보낸다. 만일 비디오 서버가 모발 i 의 요청을 승인할 수 없으면, 거절 메시지가 기지국에 보내진다. 승인 메시지가 모발 i 에 도착하면, 이 모발은 준비완료 메시지를 기지국에 보내고, 그 메시지는 비디오 서버에게 릴레이 된다. 그러면 비디오 서버는 기지국에 비디오 데이터 전송을 시작하고, 기지국은 그 데이터를 비디오 프록시 버퍼에 버퍼링한 후 모발 i 에게 릴레이 한다. 우리는 기지국내에 각 비디오 연결 당 존재하는 비디오 프록시 버퍼가 항상 비어있지 않다고 가정한다. 그 이유는 비디오 서버와 기지국은 고속의 유선으로 연결되어 있고, 우리는 모발과 기지국간의 병목에 초점을 맞추기 때문이다.

전형적인 무선 VOD 시스템에 비교했을 때 이 논문이 고려하는 시스템은 응용 계층과 MAC 계층간 최적화를 위해서 두 종류의 부가 정보를 활용한다. 첫째로, 우리 시스템은 비디오 트레이스 파일을 활용한다. 둘째로, 모발 i 가 추가적으로 행위 인자 B_i 와 s_i 의 값을 기지국에 보낸다. 그러나 이 추가적인 정보 전달로 인해 발생하는 오버헤드는 크지 않다. 한 시간 분량의 비디오에 대응되는 비디오 트레이스 파일은 설사 텍스트 형태로 쓰인다 할지라도 그 크기는 수백 킬로바이트 정도에 지나지 않는다. 이 경우 비디오 서버로부터 기지국까지 유선 링크를 통한 트레이스 파일의 전송 지연 시간은 매우 짧다. 또한 비디오 트레이스 파일과 행위 인자들은 연결 설정 동안 딱 한번만 전송된다.

이 논문에서 제안하는 방식은 적응 변조와 가변 코딩 기법을 사용하는 임의의 무선 시스템에 적용될 수 있다. 그러나 우리는 구체성을 위해 단말과 기지국 사이는 CDMA2000 1xEV-DO 무선 시스템을 고려한다. 이 경우 채널은 타임 슬롯의 연속으로 나누어진다. 한 타임 슬롯의 기간은 1.67 ms이다. 기지국내의 스케줄러는 동일한 수준의 PER (packet-error-rate)을 성취하기 위해서 변조 방식, 코딩 비율, 그리고 타임 슬롯의 개수를 모발의 채널 품질에 따라서 서로 다르게 할당한다. 논의를 단순화시키기 위해서, 실시간 패킷들은 소실되는 경우 재전송되지 않는 것으로 간주한다.

이 논문의 목적은 비실시간 응용에 대해서 시스템 성능이 심각하게 해를 입지 않으면서 VOD 응용에 대해서 고수준의

QoS를 제공하는 스케줄러를 설계하는 것이다. 우리는 모발이 겪게 되는 버퍼링 기간의 수가 적으면 그 모발이 더 나은 QoS를 받았다고 정의한다. 또한, 버퍼링 기간의 수가 동일한 경우 모발의 평균 서비스 단절 시간이 짧을수록 더 나은 QoS를 받고 있다고 간주한다.

3. 제안하는 스케줄러

모발 i 가 N 개의 비디오 프레임으로 구성된 비디오 파일을 요청했다고 하자. 그 프레임의 크기를 각각 F_0, \dots, F_{N-1} 라고 하자. 이 값들은 대응되는 비디오 트레이스 파일에 쓰여진다. 이 트레이스 파일은 헤더에 r_i 로 표기하는 초당 프레임 수, 즉, frs 를 포함한다. 그러면 비디오 프레임 간 시간 구간은 $\tau_i = 1/r_i$ 가 된다. 또, 연결 설정 동안 기지국은 두 행위 인자 B_i 와 s_i 의 값을 모발 i 로부터 받고, 비디오 서버로부터는 해당 비디오 트레이스 파일을 전송받는다.

제안하는 스케줄러의 핵심은 모발 i 의 행위 인자들을 통해서 (기지국과 단말 사이의 전파 속도와 패킷 소실을 무시하면) B_i 에 현재 남아 있는 데이터양을 정확하게 추산한다는 점이다. 이를 위해 제안하는 스케줄러는 세 개의 핵심 변수, t_p , N_t , N_p 를 유지한다. 변수 t_p 는 가장 최근 재생되는 있는 프레임의 재생 시작 시간 시점을 저장한다. 변수 N_t 는 기지국에서 그 모발로 프레임 단위로 전송 완료된 가장 최근 비디오 프레임 번호를 저장한다. 만일 현재 시간 t 까지 기지국에서 모발 i 에게 전송된 양을 $A_i(t)$ 로 나타낸다면, 변수 N_t 의 값은 $\sum_{j=0}^{N_t} F_j \leq A_i(t)$ 를 만족하는 가장 큰 k 의 값으로 계산될 수 있다. 마지막으로, 변수 N_p 는 모발이 재생 기간 내에 있을 때 현재 재생되고 있는 비디오 프레임 번호를 저장한다. 변수 N_p 의 값은 현재 시간 t 까지 모발 i 가 재생 행위를 통해서 소모한 비디오 데이터의 양을 $W_i(t)$ 로 나타낼 때, $\sum_{j=0}^{N_p} F_j > W_i(t)$ 을 만족하는 가장 작은 l 로 계산될 수 있다. 우리는 각 비디오 프레임이 프레임 주기의 끝 시점에서 소모된다고 간주한다.

위의 주요 변수들을 이용하면 모발 i 가 얼마나 오랫동안 비디오를 끊임없이 재생할 수 있는지를 계산할 수 있다. 구체적으로, 모발 i 가 현재 시간 t 로부터 끊임없이 재생을 계속할 수 있는 시간을 $C_i(t)$ 로 나타내자. 그러면, $C_i(t)$ 는 버퍼 내에 존재하는 비디오 프레임의 수와 현재 비디오 프레임 재생 구간의 끝 시점까지 남은 시간으로부터 계산될 수 있다. 즉,

$$C_i(t) = \max(0, (N_t - N_p - 1) * \tau_i + (\tau_i - (t - t_p))). \quad (1)$$

식 (1)에서 $C_i(\cdot)$ 의 값이 감소할수록, 모발 i 가 재생 기간에서 버퍼링 기간으로 전환할 잠재적 위험성이 증가한다. 이러한 위험을 피하기 위해서는, 만일 $C_i(\cdot)$ 의 값이 작아지면, 모발 i 는 채널 품질의 좋고 나쁨과 관계없이 급히 더 많은 실시간 패킷을 받을 필요가 있다. 반면, 만일 $C_i(\cdot)$ 의 값이

충분히 큰 경우 무선 자원은 다른 모발들에게 할당되어도 무관하다. 모발 i 가 이와 같은 잠재적 위험을 갖는 지 그렇지 않은 지를 결정하기 위해서, 우리는 각 모발 i 에 대해 안정 시간 ST_i 를 정의한다. 우리는 어떤 시간 t 에 대해 $C_i(t)$ 가 ST_i 보다 작으면, 모발 i 는 서비스 중단에 대한 잠재적 위험을 갖는 것으로 간주한다. ST_i 의 값은 경험적으로 몇 개의 비디오 프레임 재생 시간으로 설정한다.

제안하는 스케줄러는 실시간 VOD 응용들뿐만 아니라 비실시간 응용들도 고려한다. 그러나 양쪽 타입의 응용들은 서로 상충하는 성능 목적을 추구하므로 이 목적들을 동시에 최대화시킬 수는 없다. 따라서 제안하는 스케줄러는 비실시간 응용에게 너무 피해를 주지 않으면서 실시간 응용들에게 이점을 준다. 이 목적을 위해, 제안하는 스케줄러는 두 가지 모드, 즉, 실시간 모드와 공평 모드를 갖는다. 만일 적어도 하나 이상의 모발이 재생 기간 내에 있으면, 스케줄러는 실시간 모드로 동작한다. 그렇지 않으면 공평 모드로 동작한다. 실시간 모드에서 임의의 시간 t 에 VOD 응용을 실행하는 모발 i 는 다음과 같이 실시간 우선순위 값으로 $P_i(t)$ 를 갖는다.

$$P_i(t) = DRG_i(t) \cdot \frac{ST_i}{C_i(t)}. \quad (2)$$

식 (2)에서 $DRG_i(t)$ 는 시간 t 의 모발 i 에 대한 채널 품질에 비례해서 전송할 수 있는 데이터양을 나타낸다. 제안하는 스케줄러는 실시간 모드로 동작할 때 가장 큰 실시간 우선순위 값을 갖는 모발을 선택한다. 제안하는 스케줄러가 공평 모드로 동작할 경우 비실시간 응용의 목적을 만족시키기 위해서 PF 스케줄러와 동일하게 동작한다. 다시 말해, 각 모발 i 는 다음과 같은 공평 우선순위 값 $Q_i(t)$ 를 갖는다.

$$Q_i(t) = \frac{DRG_i(t)}{R_i(t)}. \quad (3)$$

식 (3)에서 $R_i(t)$ 는 현재 시간 t 로부터 과거 일정한 고정된 시간 구간 동안 전송되었던 평균 데이터양을 의미한다. 제안하는 스케줄러는 공평 모드로 동작할 경우 가장 높은 공평 우선순위 값을 갖는 모발이 선택된다. VOD 응용을 실행하는 모든 모발은 실시간 우선순위 값과 공평 우선순위 값을 갖지만, 다른 모발들은 공평 우선순위 값만 받는다.

4 시뮬레이션 결과 및 결론

시뮬레이션 실험은 CDMA2000 1xEV-DO를 가정하고, 한 개의 중앙 셀과 이를 둘러싸는 6개의 셀이 있어 중앙 셀을 간섭하는 신호를 생성한다. 채널 모델은 경로 감소 지수가 4인 Rayleigh fading을 따른다고 가정하였다. 각 실험은 총 20개의 모발이 서로 다른 시간 구간에 대해서 서로 다른 속도로 중앙 셀 내를 이동한다. 20개의 모발 중 VOD 응용을 실행하는 모발의 수, 즉, VOD 모발의 수는 실험마다 변경된다. 각 VOD모발은 평균 54 Kbps의 전송 대역폭을 갖고 frs가

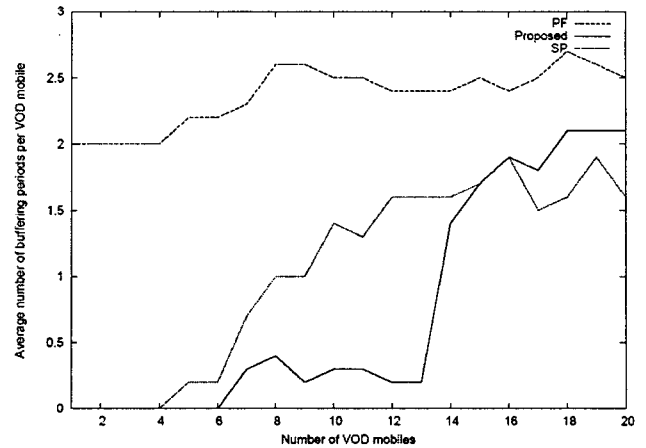


그림 1 VOD 모발 당 버퍼링 기간의 평균 수
Fig. 1 Average number of buffering periods per VOD mobile

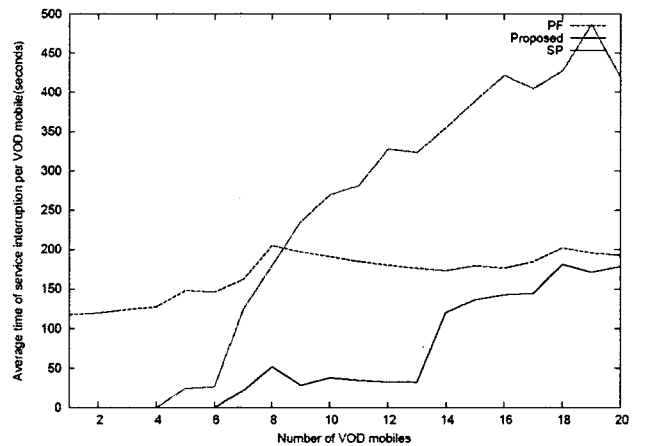


그림 2 VOD 모발 당 평균 서비스 중단시간
Fig. 2 Average service interruption time per VOD mobile

24인 동일한 VBR 비디오를 요청한다. VOD 모발의 버퍼 크기는 800 킬로바이트이고, 재생 시작 위치는 400 킬로바이트로 설정되었다. 안정 시간은 100 비디오 프레임 구간, 즉, $100/24 = 4.17$ 초이다. VOD 모발이 아닌 다른 모든 모발들은 배경 트래픽을 생성하기 위해 항상 충분한 패킷을 생성한다고 가정한다. 시뮬레이션 시간은 각 실험별로 17분이고, 세 종류의 스케줄러, 즉, PF, SP, 그리고 제안하는 스케줄러를 고려하였다.

그림 1은 총 20개의 모발 중 VOD 모발의 수의 비중이 증가함에 따라, VOD 모발 당 버퍼링 기간의 평균 수 (최초 버퍼링은 제외)를 나타낸 것이다. 그림 2는 그림 1에 대응되도록, 버퍼링의 재시작으로 인한 VOD 모발 당 평균 서비스 중단 시간을 나타낸다. 두 그림에서 제안하는 스케줄러는 다른 스케줄러들보다 더 나은 QoS, 즉, 버퍼링 기간 수의 수와 서비스 중단 시간 측면에서 다른 스케줄러보다 우수하다. SP 스케줄러는 VOD 모발의 수가 적을 때만, 예들 들어, 두 그림에서 VOD 모발의 수가 5를 넘지 않을 때 까지만 가장 좋은 QoS를 제공하고 있다. 그렇지만, VOD 모발의 수가 그 이

상으로 증가하면 심지어 PF 스케줄러보다 더 나쁠 수도 있다. 예를 들어, 그림 2에서 VOD 모발의 수가 9개 이상인 경우 SP 스케줄러는 제안하는 스케줄러뿐만 아니라 PF 스케줄

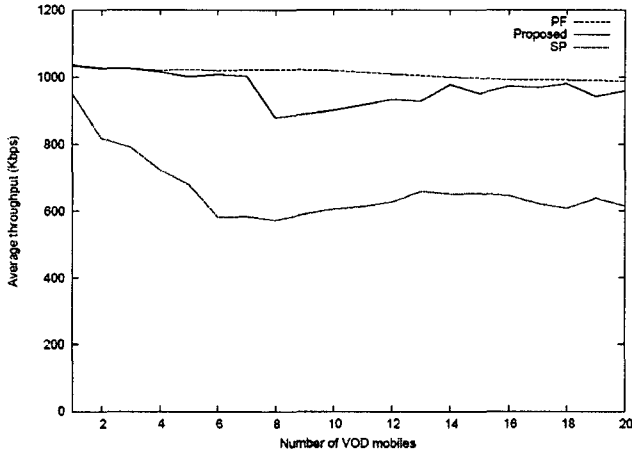


그림 3 시스템 성능 비교
Fig. 3 Comparison of system throughput

러보다도 더 긴 평균 서비스 중단 시간이 발생된다. 주 이유는 VOD 모발이 일단 한번 재생을 중지하면, 그 모발은 재생 기간 내에 있는 VOD 모발이 모두 존재하지 않을 때까지 패킷을 받을 수 없기 때문이다.

그림 3은 평균 시스템 성능을 비교한다. PF 스케줄러는 가장 높은 성능을 성취한다. 그러나 제안하는 스케줄러와 비교했을 때 VOD 응용의 QoS를 만족시키는데 불충분하다. 반면 SP 스케줄러는 무조건적인 실시간 패킷 선호 때문에 PF 스케줄러보다 훨씬 낮은 성능을 낳는다. 제안하는 스케줄러는 합리적인 수준의 시스템 성능을 제공하여 PF 스케줄러와의 성능차이가 작으면서도 그림 1과 2에 보인바와 같이 QoS 제공에 더 나은 이점을 갖는다.

5 결 론

본 논문은 VOD 응용에 대해서 합리적 수준의 시스템 성능을 성취하면서도 고품질을 제공하는 MAC 스케줄러를 제안하였다. 제안한 스케줄러는 계층간 최적화 기법을 기반으로 응용과 MAC 계층 간에 활용하여, VOD 응용의 QoS 제공 측면에서 PF 스케줄러나 SP 스케줄러보다 더 우수하면서도 시스템 성능이 크게 떨어지지 않음을 보였다.

감사의 글

이 연구는 정보통신부 정보통신연구진흥원의 2005 IT 기초기술연구지원사업(과제번호: B1220-0501-0327)의 지원에 의하여 이루어진 연구로서, 관계부처에 감사드립니다.

참 고 문 헌

[1] "CDMA2000 High Data Rate Packet Data Air Interface Specification," October 2002, 3GPP2 C.S0024-0 v4.0.
 [2] M. Andrews, "Instability of the Proportional Fair Scheduling Algorithm for HDR," *IEEE Tran. Wireless Communications*, vol. 3, no. 5, pp. 1422-1426, 2004
 [3] H. Zhang, "Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks," *Proc. IEEE*, vol. 3, no. 4, pp. 391-430, 1995.
 [4] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, "Cross-Layer Design for Wireless Networks," *IEEE Communications Magazine*, vol. 41, no. 10, pp. 74-80, 2003.

저 자 소 개



편 기 현 (片基鉉)

1995년 인하대학교 전자계산공학과 학사 졸업. 1997년 KAIST 전산학과 석사 졸업. 2003년 KAIST 전산학과 박사 졸업. 2004년 KAIST 전자공학 Post doc. 2004~현재 전북대학교 전자정보공학부 조교수로 재직 중. <주관심분야: 유무선 고품질 서비스, 패킷 스케줄링, 임베디드 시스템>
 Tel : 063-270-4139
 Fax : 063-270-2394
 E-mail : khpyun@chonbuk.ac.kr



조 성 익 (趙成翊)

1987년 전북대학교 전기공학과 학사 졸업. 1989년 전북대학교 전기공학과 석사 졸업. 1994년 전북대학교 전기공학과 박사 졸업. 1994년~2004년 Hynix 반도체 메모리 연구소 책임연구원. 2004년~현재 전북대학교 전자정보공학부 조교수. <주관심분야: 저전압/고속 Graphic DRAM, Low-voltage Low-power analog circuit, High speed data Interface circuit, ADC/DAC, Filter, PLL/DLL >
 Tel : 063-270-4137
 Fax : 063-270-2394
 E-mail : sicho@chonbuk.ac.kr