인터넷에서 종단간 경로의 유효 병목 대역폭 측정 속도 개선

(Speedup in Measuring the Effective Bottleneck Bandwidth of an End-to-End Path in Internet)

유 한 승 † 장 주 욱 ††
(Han-seung Yoo) (Ju-wook Jang)

요 약 패킷 쌍을 이용한 병목 대역 측정 방법 중 Potential Bandwidth Filter를 이용하여 두 개의 패킷 앞에 다른 패킷이 끼어 들어 생기는 시간축 압축현상 잡음을 제거하는 기법의 수렴 속도를 개선하는 방법을 제안한다. 본 논문에서는 Potential Bandwidth Filter에서 고정된 값으로 증가시켜 수렴시키는 방법 대신에 지수적으로 증가시키는 방법을 사용하여 수렴속도를 개선하였다. 또한 효과적인 필터의 성능 유지와 변화되는 대역폭의 측정을 위하여 상한값(MAX)과 하한값(MIN)을 이용하여 Potential Bandwidth(PB)를 조절하였다. 실험 결과 본 논문에서 제안한 알고리즘이 기존의 알고리즘보다 45-89% 빠르게 수렴하는 것을 보였다.

Abstract A new scheme is proposed to speed up a known bandwidth measuring method which employs potential bandwidth for filtering out noises (in estimation) from time compression caused by a packet queueing ahead of two probe packets. Instead of incrementing the potential bandwidth by a fixed amount as in the original method we increase the potential bandwidth exponentially for faster convergence. To retain its filtering capability as well as its agility to adapt to new bottleneck bandwidth, each trial potential bandwidth(PB) is adjusted using MAX and MIN as upper bound and lower bound. An experiment using known bandwidths shows 45-89% improvement in convergence time.

1. 서 론

가용 대역폭이 시시각각으로 변하는 인터넷 환경에 최적으로 적응하는 서비스를 위해서는 가용 대역폭의 측정이 중요하다. 기 제안된 여러 가지 대역폭 측정 기법 중 동시에 보낸 두 패킷의 도착 시간차를 이용하여 대역폭을 측정하는 패킷페어(Packet Pair) 알고리즘이 활발히 연구되어 왔다. 그러나 두 패킷 사이에 다른 패킷이 끼어 들어 생기는 시간축 확장현상(Time Extension)나 두 패킷 앞에 다른 패킷이 큐잉 되어 있는 다른 패킷에 의해 생기는 시간축 압축현상(Time Compression)에 의하여 대역폭을 정확히 측정 할 수 없다는 문제가 있다.

이러한 문제를 극복하기 위해서는 필터를 사용한다. 많이 사용하는 필터는 여러 번의 대역폭 측정을 통해 얻은 값들의 통계적 분산을 통하여 값을 얻는 것이다. 측정한 값들의 유효성은 실제 값의 주위에 몰려 있게 마련이며, 히스토그램을 통해서 쉽게 얻을 수 있다[2]. 그러나 히스토그램을 이용할 경우, 많은 수학적 계산에 따라 속도가 느리다는 단점이 있다.

다른 방법은 Potential Bandwidth(PB)와 Measured Bandwidth(MB)를 이용하여 그래프 적으로 결정하는 Potential Bandwidth Filter(PBF)로 기존의 다른 필터보다 계산이 쉽고, 빠르면서 정확하다는 장점이 있다[1].

그러나 Potential Bandwidth Filter는 필터를 위한 그래프의 기준선, 즉 대역폭을 나타내는 선에 수렴하는데 많은 시간이 소요되는 문제점과 변화된 대역폭을 측정하기 위해서는 처음부터 다시 측정하여야 한다는 문제점이 있다. 따라서 본 논문은 PB를 이용한 대역폭 측정에 있어서, Potential Bandwidth Filter의 대역폭 수렴의 시간적 소요 문제를 개선하고 변화된 대역폭에 적응적인 알고리즘을 제시하여 대역폭 측정을 쉽고, 정확하면서, 빠

† 비 회 원 : 서강대학교 전자공학과
wamozartkr@hotmail.com
†† 종신회원 : 서강대학교 전자공학과 교수
jjang@ccs.sogang.ac.kr
논문접수 : 2000년 3월 28일
심사완료 : 2001년 1월 29일

르게 하고자 하였다.

본 논문 2장에서는 실험을 위한 방법과 실험 시스템에 대해서 살펴본다. 3장에서는 본 논문에서 대역폭 측정시 사용한 패킷페어 알고리즘과 Potential Bandwidth Filter 방법에 대해 살펴볼 것이다. 제 4장에서는 본 논문에서 수행한 수렴 속도 개선 알고리즘과 실험결과에 대해 살펴보며, 제 5장에서는 변화하는 대역폭을 측정하는 방법에 대해 실험과 함께 살펴볼 것이다. 마지막 제 6장에서는 본 논문의 전체적 결론과 추후과제에 대하여 논 할 것이다.

2. 실험 시스템

본 논문은 End-to-End 사이의 병목 대역폭을 측정하는 실험으로 그림 1과 같은 구성도로 실험하였다.

그림 1 실험 시스템의 구성도

그러나, 실제 네트워크에서는 정확한 실제 대역폭을 알 수 없다. 따라서 우리가 알 수 있는 대역폭과, 또한 실험을 위한 대역폭 조절을 위해 리눅스를 운영체제를 사용하는 Pentium-II급 컴퓨터를 이용하여 대역폭을 임의로 조절할 수 있는 가상의 실험 시스템을 구축하였다. 가상의 대역폭 생성 시스템은 송신 측에서 보낸 두 패킷을 원하는 병목 대역폭의 송신율로 수신 측에 보내는 역할을 한다.

3. PBF를 이용한 패킷 페어 알고리즘

3.1 Packet Pair Method

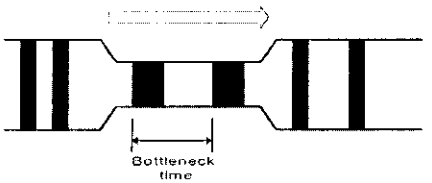

그림 2 Packet Pair 알고리즘 동작 원리

패킷 페어의 기본적인 원리는 다음과 같다. 그림 2와 같이 크기가 같은 연속된 두 패킷은 병목 현상 지역을 통과 할 때는 두 패킷 사이에 t만큼의 간격이 생기면 이 병목 현상의 대역폭은 식 (1)과 같다.

$$b_b = \frac{s_2}{t} \quad (1)$$

여기서 s_2는 두 번째 패킷의 크기이고 b_b는 병목 현상 지역의 대역폭을 나타낸다.

그러나 패킷 페어 알고리즘은 다른 패킷의 방해가 없는 경우를 생각한 것이다. 만약 두 패킷 사이에 다른 s_a 크기를 갖는 패킷이 존재하게 되면 대역폭의 크기는 식 (2)가 될 것이다.

$$b_b = \frac{s_2 + s_a}{t} \quad (2)$$

식 (2)처럼 표현된 대역폭을 시간축 확장현상이라고 한다. 또 다른 문제는 두 패킷 앞에 다른 큰 패킷이 존재하여 두 패킷 간격이 좁아지는 시간축 압축현상 현상이 있다. 시간축 확장현상과 시간축 압축현상 현상은 정확한 대역폭 측정에 어려움을 준다.

3.2 Potential Bandwidth Filter

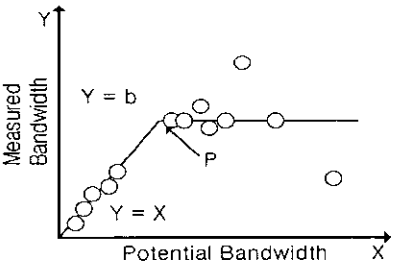

그림 3 Potential Bandwidth Filter

패킷 페어 방법에는 3.1에서 언급되어진 문제에 의해 신뢰성이 떨어지게 된다. 이러한 문제를 해결하기 위해서 필터를 사용하는데, 본 논문에서는 PB와 MB를 이용하는 Potential Bandwidth Filter를 사용하였다. PB는 측정 가능한 대역폭의 최대 값이며 MB는 실제 측정된 대역폭일 때, 그림 3의 Y=b 라인인 실제 대역폭을 중심으로 측정된 값과 비교하며 필터링 하는 것이다.

표 1 Potential Bandwidth Filter 방법

MB = P	Actual Bandwidth
MB > P	Time Compression
MB < P	Time Extension

표 1과 같이 MB 값이 Y=b 라인에 있는 P값과 같으면 실제 대역폭과 같으며 그 외에는 시간축 압축현상이나 시간축 확장현상으로 생각하여 잘못된 값으로 생각하면 된다.

따라서 Potential Bandwidth Filter를 이용 할 경우, MB의 상태 판단이 빠르고, 수학적 계산 또한 적음을 알 수 있다.

4. 수렴 속도 개선

대역폭을 찾기 위하여 그림 3의 Y=b 라는 직선을 찾는 것이다. 그러나 그림 3에서도 나타나 있듯이 측정된 대역폭이 Y=X 직선 상에 있을 경우도 있고, 또한 Y=b 직선 상에 있는지 알 수가 없다. 따라서 얼마나 빨리 Y=b직선을 찾는 것이 중요하게 되었다.

이 Y=b 직선을 찾기 위해 큰 패킷을 이용 할 수 있지만, 초기 큰 패킷의 크기 결정과, 또한 큰 패킷을 이용할 경우 생길 수 있는 패킷 전송 지연과, 패킷 손실을 고려해야 한다. 또한 한 차례의 측정은 3.1에서 언급한 문제에 의한 잘못된 값을 얻을 수 있다는 것도 고려해야 한다. 따라서 최대한 빠르게 Y=b 직선의 시작 지점인 P(Knee Point)점을 찾는 알고리즘이 필요하게 되었다.

기존의 P점을 찾는 방법은 PB를 단계적으로 MB와 비교하여 PB가 증가해도 MB가 증가하지 못하는 점 P점을 찾는 방식이다. 그러나 이 방식에서는 증가하는 단계가 작을 경우 P점을 찾는 시간이 걸리고, 증가하는 단계가 큰 경우 정확한 값을 얻기가 힘들다는 단점이 있다. 따라서 본 논문에서는 처음에는 지수적으로 단계의 크기를 증가시키다가, MB가 PB 보다 작은 값이 되면 증가율을 반으로 감소시키는 방식을 채택하였다.

4.1 제안된 알고리즘

그림 3의 P점을 찾기 위해서는 상한값과 하한값을 이용한다. 하한값은 올바른 대역폭값이 존재할 범위의 최저 값이며, 상한값은 최고값이다. 따라서 다음과 같은 순서로 범위를 좁혀 나감으로써 대역폭을 얻을 수 있다.

[단계 1] 상한값을 찾기 위해 초기의 작은 값에서부터 2의 지수로 값을 찾아 나간다.

[단계 2] 만약 PB보다 작은 MB가 나오면 상한값과 하한값의 1/2의 값을 취하여 다시 대역폭을 측정한다.

[단계 3] 상한값과 하한값의 중간값을 기준으로 측정한 대역폭이, 중간값보다 크면 중간값이 하한값이 되며 중간값보다 작으면 중간값이 상한값이 되어 다시 측정한다.

[단계 4] ③번 과정을 실 대역폭을 찾을 때까지 계속한다.

위의 과정을 통하여, 대역폭을 빠르면서, 정확하게 구할 수 있다.

4.2 실험 및 결과

그림 4는 기존 Potential Bandwidth Filter에서 순차적 증가와 순차적 감소 알고리즘의 2Kbps/step, 5Kbps/step, 10Kbps/step의 증가율에 따른 대역폭 측정과, 본 논문에서 제안된 알고리즘에 따른 대역폭 측정의 결과를 나타낸 그래프이다. 실제 대역폭은 100Kbps이다.

표 2 실험 결과 비교표

알고리즘	순차적 (2Kbps/step)	순차적 (5Kbps/step)	순차적 (10Kbps/step)	지수적
시간 (step)	약 55	약 19	약 11	약 6

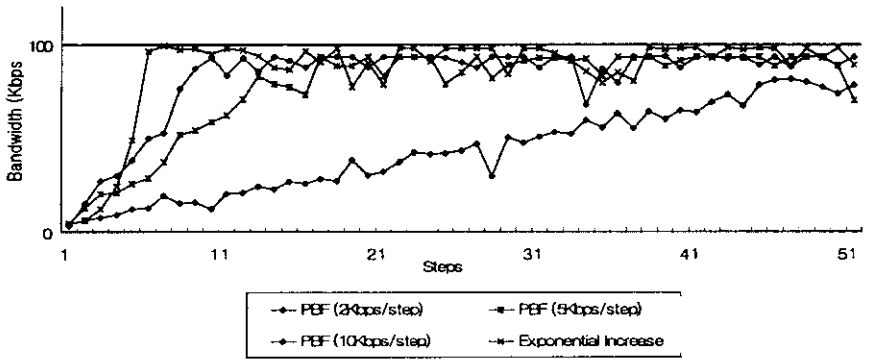

그림 4 순차적 증가와 지수적 증가에 따른 대역폭 측정

그림 4와 표 2의 결과처럼 제안된 알고리즘이 고정된 비율로 증가하는 방법보다 약 45-89% 만큼 성능 향상이 있음을 알 수 있다.

고정된 비율로 증가하는 방법의 결과에 따르면 증가율을 증가시킬수록 빨리 실제 대역폭에 도달함을 알 수 있다. 그러나 4.1절에서 언급했듯이 증가율에 따른 소요시간 문제와, 정확히 측정하기 어렵다는 문제를 안고 있다. 또한 그림 4의 그래프처럼 진동 현상이 나타남을 알 수 있다.

5. 변화하는 대역폭 찾기

네트워크의 대역폭은 시시각각 계속 변한다. 따라서 기존의 패킷페어 방식에서 값의 유효성을 찾기 위해 수십 또는 수백 번의 측정을 이용하여 대역폭을 찾는 것은 현실적으로 어렵다. 본 논문은 4.1에서 제안된 알고리즘을 수정하여 빠르게 변하는 네트워크 대역폭을 측정하였다.

5.1 방법(1)

첫 번째 방법은 변화된 대역폭을 측정하기 위하여 측정된 대역폭을 중심으로 어느 시간 경과 후 다시 대역폭 측정을 시작하는 방법이다.

4장에서 언급한 초기 수렴으로 인하여 대역폭을 측정한 후 정해진 시간만큼 기다린 후 상한값은 기존의 두 배로 설정하고, 하한값은 0으로 설정한 다음 4장에서 언급한 제안된 알고리즘을 통하여 대역폭을 측정한다.

그러나, 이 방법은 변화된 대역폭을 찾기 위하여 재설정하는 시간만큼 기다려야 하는 문제점이 있다. 즉, 기다리는 동안 대역폭의 변화가 생긴다면, 기다리는 시간만큼 대역폭 찾는 속도가 늦어진다.

5.2 방법(2)

5.1의 방법은 다시 대역폭을 측정하기 전에 대역폭이 변화하였을 경우 변화 대역폭을 찾기 위한 재 설정 시간까지 기다려야 하는 문제점이 있다. 이를 극복하기 위하여 두 번째 방법에서는 측정을 지연시키는 재 설정 시간 대신에 대역폭의 변화를 감지하는 방법을 이용하였다.

방법(2)는 다음과 같다. 초기 수렴 알고리즘을 통하여 대역폭의 측정이 끝난 후 좁아진 상한값과 하한값은 우리가 대역폭의 측정 시 설정해 놓은 대역폭 오차 범위 내에 있다.

따라서 이 오차 내에서 움직이는 값은 모두 대역폭을 인식하기에 대역폭의 변화를 감지 할 수 없다. 또한 상한값이 PB이기 때문에 상한값을 벗어난 값은 우리가 얻을 수 없다. 따라서 이 상한값을 대역폭 오차 범위보다 약간 위에 두고 감시를 하면 된다. 만약 측정된 대역폭이 상한값에 접근하거나, 너무 멀어지면, 대역폭의 변화가 있음을 감지하고 대역폭의 측정을 다시 한다.

5.3 방법(3)

세 번째로 제안된 방법은 4.1의 제안된 알고리즘에서 상한값의 움직임을 하한값에 적용한 방법이다. 5.2의 방법(2)에서 대역폭의 값을 얻기 위하여 상한값과 하한값이 어느 일정한 범위 안으로 들어와야지만, 확실한 대역폭으로서 값을 얻었다.

그러나 일정한 범위 안으로 들어오기까지의 시간적 지연은, 그 사이에 일어나는 대역폭의 변화를 감지 할 수 없다.

방법(3)은 표 3의 코드와 같이 측정된 대역폭이 상한값에 접근하면, 상한값을 두 배 해주고, 하한값보다 아래에 오면, 전 하한값의 반을 하한값으로 취하는 방법이다. 만약 측정된 대역폭이 상한값과 하한값 사이에 존재하면, 중간값을 기준으로 상한값과 하한값을 변화시켜 재 측정하게 된다. 따라서 상한값과 하한값 사이의 차가 줄어들어 실제 대역폭에 접근하게 된다.

표 3 방법(3)의 Pseudo 코드

```
MB : Mesured Bandwidth
PB : Potential Bandwidth
MAX : Maximum of the Range
MIN : Minimum of the Range

MAX, PB = α
MIN = 0

Do
   If MB > PB Then
        Error "Time Compression"
   Else If (MB - PB) < δ Then
        MIN = MAX
        MAX = 2 * MAX
        PB = MAX
   Else If MB < MIN Then
        MAX = MIN
        MIN = MAX / 2
        PB = MAX
   Else
        IF MB > (MAX + MIN) / 2 Then
           MIN = (MAX + MIN) / 2
        Else
           MAX = (MAX + MIN) / 2
           PB = MAX
        Endif
   Endif
End do
```

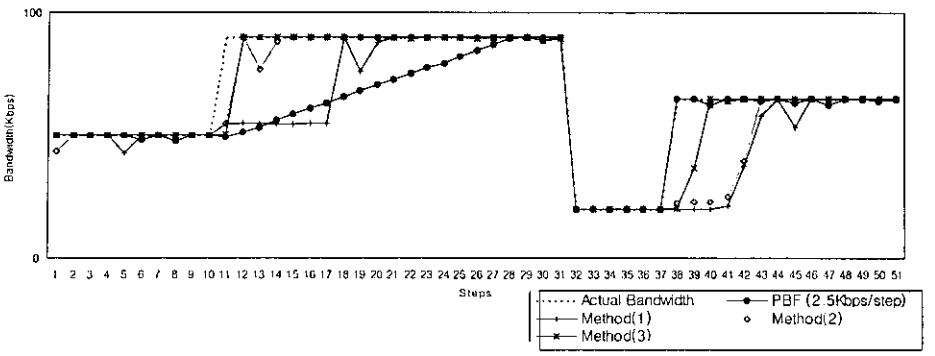

그림 5 변화하는 대역폭 측정

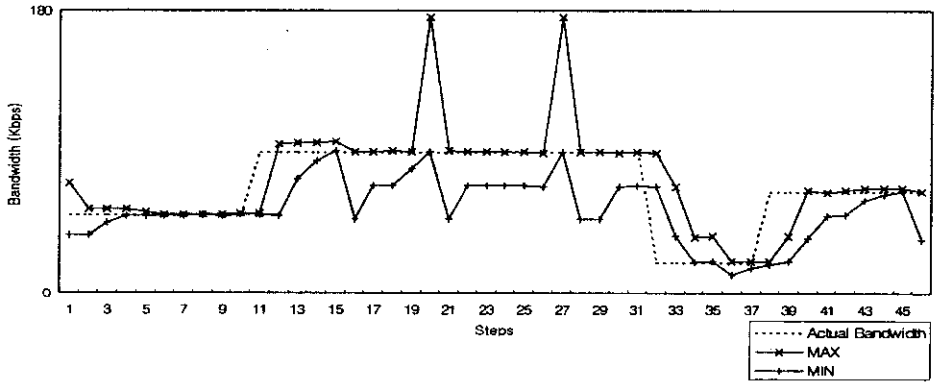

그림 6 대역폭 변화에 따른 MAX와 MIN의 변화 모습

방법(3)은 대역폭의 변화에 민감하여, 잘못 측정된 대역폭에도 반응하는 문제점이 있다. 그러나, 변화된 대역폭에 빠르게 적응해나가는 모습을 볼 수 있다. 따라서 언제 대역폭이 변하더라도, 대역폭의 변화를 감지할 수 있으며, 계속적인 대역폭의 측정을 할 수 있다.

5.4 실험 및 결과

그림 5는 위에서 5.1에서부터 5.3까지 제시한 방법과 기존의 고정된 증가율로 측정하는 방법을 통하여 실험한 결과이다. 실제 대역폭이 50Kbps, 90Kbps, 20Kbps, 65Kbps로 변화할 때 각 방법에 의해 측정된 대역폭 변화 모습들이다. 그림 6을 통하여 알 수 있듯이, 50Kbps에서 90Kbps로 대역폭이 바뀌었을 때, 방법 (1)은 7 step만에 변화된 대역폭을 찾은 반면, 방법 (2), (3)은 1 step만에 변화된 대역폭을 찾았다. 방법 (1)에서는 대역폭을 찾기 위해서 다시 찾기 위한 재 설정 시간까지 기다려야 하기 때문에, 결과처럼 약 6, 7 step 늦게 대역폭을 찾았다.

그림 6의 90Kbps에서 잠시 20Kbps로 간 다음 다시 65Kbps로 변화였을 때는 방법 (1)과 (2)는 7 step만에 변화된 대역폭을 찾았고, 방법 (3)은 3 step만에 대역폭을 찾았다. 방법 (1)은 재 설정 시간에 의해 방법 (3)보다 4 step이 늦었다. 방법 (2)는 5.2에서 언급하였듯이, 대역폭을 확신하기 위한 상한값과 하한값의 값을 좁혀나가는 중에, 대역폭의 변화가 있었으나, 이를 감지하지 못하기 때문에 변화된 대역폭을 찾는데, 시간이 걸린 것이다.

결론적으로 본 논문에서 제안한 방법(3)은 다른 방법보다 빠르게 대역폭을 찾음과 동시에 초기값 설정에서도 빠르게 동작하는 방법이다. 다음 그림 6은 제안된 방법 (3)에서 사용한 상한값과 하한값의 변화도이다. 그림 6을 통해서 알 수 있듯이 상한값과 하한값 사이에 실제의 대역폭이 존재함을 알 수 있으며 또한 측정된 대역

폭은 이 사이에 있음을 알 수 있다.

6. 결 론

대역폭의 측정이 중요해짐에 따라 패킷페어 알고리즘을 이용한 대역폭의 측정 연구가 활발해 지고 있다. 또한 정확한 대역폭 값을 얻기 위하여 필터의 모델이 제안되었다. 기존의 Potential Bandwidth Filter는 계산이 쉽고, 빠르다는 장점이 있는 반면, 필터를 위한 기준선을 찾는데 느리다는 문제점이 있다. 따라서 본 논문에서는 PB의 기준선에 더욱 빠르게 수렴하는 알고리즘을 제안하였다. 본 논문에서 제안된 알고리즘은 기존 순차적 증가율을 사용한 알고리즘보다 약 45-89% 빠르게 Potential Bandwidth Filter의 기준선에 수렴하는 것을 보았다.

빠르게 변화하는 네트워크 대역폭에 본 논문에서 비교 분석 결과 방법(3)이 다른 방법보다 변화된 대역폭을 빠르게 찾는데 적합함을 보였다.

향후 연구방향은 실제 네트워크에 적용하여 네트워크 부하가 큰 지역에서의 동작에 대하여 제안된 알고리즘의 성능을 평가해야 할 것 같다.

참 고 문 헌

[1] Kevin Lai, Mary Baker, "Measuring Band- width," In proceedings of the IEEE INFO- COMM99, pp.235-245, 1999

[2] Robert L. Carter, Mark E. Crovella, "Measuring Bottleneck Link Speed in Packet-Switched Networks," Technical Report BU-CS-96-007, Boston University, 1996

[3] Robert L. Carter, Mark E. Crovella, "Dynamic Server Selection using Bandwidth Probing in Wide-Area Networks," Technical Report Bu-CS-96-006, Boston University, 1996

[4] Vern Paxson, "End-to-End Internet Packet Dynamics," In Proceedings of SIGCOMM, 1997

[5] Srinivasan Keshav, "A control-theoretic approach to flow control," In Proceedings of SIGCOMM, 1991

[6] W. Richard Stevens, TCP/IP Illustrated Vol.1, Addison-wesley, 1994

유 한 숭

1997년 서강대학교 전자공학과 졸업(공학사). 1997 ~ 현재 서강대학교 전자공학과 대학원 재학중. 관심분야는 Internet Protocol, Traffic Analysis, Multimedia

장 주 욱

1983년 서울대학교 전자공학과 학사. 1985년 학국과학기술원 전기 및 전자 석사. 1993년 Univ. of Southern California, Computer Engineering Ph.D. 1993년 ~ 1995년 삼성전자 컴퓨터개발실 선임연구원. 1995년 ~ 현재 서강대학교 전자공학과 부교수. 관심분야는 Internet Protocol, Multicast, Multimedia