

# 네트워크 이동 환경에서의 TCP 성능 향상 기법

김 명 섭<sup>†</sup> · 최 명 환<sup>\*\*</sup>

## 요 약

인터넷과 이동 네트워크로 구성된 유무선 통합 네트워크에서는 인터넷 상의 송신자와 이동 네트워크 상의 이동 호스트(MH, Mobile Host) 간 경로는 두 개 이상의 무선링크로 구성된다. 이와 같은 네트워크 이동 환경에서 TCP의 종단간 제어 특성을 유지하면서 무선링크에서의 패킷 전송 오류로 인한 TCP의 성능 저하를 극복하는 방안을 제안한다. 제안하는 방안에서는 수신된 패킷의 시퀀스 번호와 ACK 번호를 근거로 하여 무선링크 상에서의 전송 오류로 인한 패킷 손실 정보를 추출하고 이 정보를 상류의 라우터에게 전달함으로써 상류의 액세스 라우터나 이동 라우터에서 신속히 손실된 패킷을 재전송할 수 있도록 한다. 제안하는 방안은 상류 무선링크에서의 패킷전송 오류를 신속히 극복하는 특성을 가지며 모의실험을 통하여 제안한 방안의 TCP 전송률 성능을 평가하였다. TCP 종단간 제어 특성을 가지며 송신자나 수신자 측에 추가 기능이 요구되지 않는 snoop 방식과 비교할 때 제안하는 방안의 TCP 성능이 크게 개선됨을 모의실험을 통해 확인하였다.

키워드 : 이동 네트워크, 전송제어 프로토콜, 액세스 라우터, 이동 라우터

## TCP Performance Improvement Scheme in Network Mobility Environment

Kim, Myung-Sup<sup>†</sup> · Choi, Myungwhan<sup>\*\*</sup>

### ABSTRACT

In the integrated wired/wireless network which consists of the Internet and moving networks, multiple wireless links are used to connect a fixed host(FH) in the Internet to a mobile host(MH) in the moving network. For use in such an environment, we propose a scheme to overcome the TCP performance degradation due to the packet losses over the wireless links without losing the end-to-end TCP semantics. The proposed scheme in each mobile router(MR) allows to obtain the information regarding packet losses over the upstream wireless links based on the received packet sequence number and the ACK number. This information is delivered to the upstream router, which enables the upstream access router(AR) or MR to quickly retransmit the lost packets. The proposed scheme has the feature to quickly recover the packet losses incurred over the upstream wireless links and the performance of the proposed scheme is evaluated through simulation. It is shown that the significant performance gain can be obtained using the proposed scheme compared with the snoop mechanism which maintains end-to-end TCP semantics and does not require any additional features at the source and/or destination nodes.

Key Words : Network Mobility, TCP, Access Router, Mobile Router

### 1. 서 론

이동통신 가입자의 증가로 이동 호스트를 이용한 인터넷 사용이 증가하고 있으며 이를 위해 주로 Mobile IP가 사용되고 있다. 그리고 많은 인터넷 사용자들은 버스, 자동차, 비행기 그리고 배와 같은 교통수단을 이용하면서 인터넷 서비스를 받고자 한다. 이 때 교통수단은 여러 개의 호스트를 가지고 있는 이동 네트워크의 역할을 할 수 있다. 이에 따라 2002년 결성된 IETF(Internet Engineering Task Force) 네트워크 이동(NEMO, Network Mobility) 워킹 그룹에서는

네트워크를 이동 단위로 보고 Mobile IP를 이에 확장시키기 위한 연구를 진행하고 있다.

네트워크 이동 환경에서는 임의의 네트워크가 이동해서 인터넷과의 접속점이 변경될 경우, 네트워크 내 이동 호스트들은 이의 영향을 받지 않고, (그림 1)과 같이 인터넷(백본 네트워크)의 접속점에 해당하는 액세스 라우터(AR, Access Router)와의 접속을 관장하는 해당 네트워크의 이동 라우터(MR, Mobile Router)를 통해서 인터넷에 접속된다[1-4].

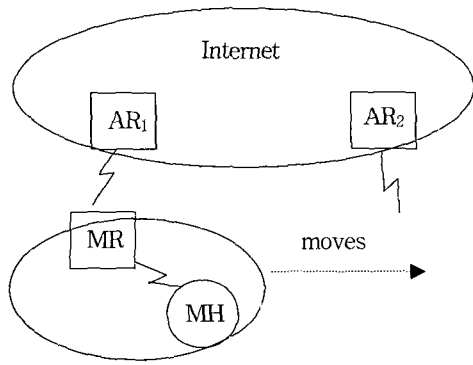
네트워크 이동 환경에서는 중첩(nested) 이동 네트워크를 구성할 수 있어서, 하나의 이동 네트워크 내부에 다수의 이동 호스트와 하위 이동 네트워크와의 통신을 가능하게 하는 여러 개의 이동 라우터가 존재할 수 있다. 이러한 환경에서, 액세스 라우터와 이동 라우터, 그리고 이동 라우터와 이동 호스트 간 또는 또 다른 이동 라우터간 무선 링크에서 발생하는 전송오류는 TCP 성능에 매우 심각한 영향을 끼치게 된다.

※ 이 연구는 2003년도 서강대학교 교내 연구비 지원과 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (ITA-2005-(C1090-0502-0027)).

† 정 회 원 : LG전자 연구원

\*\* 정 회 원 : 서강대학교 컴퓨터학과 교수

논문접수 : 2005년 12월 30일, 심사완료 : 2006년 3월 20일



(그림 1) 네트워크의 이동

TCP에서는 패킷의 손실을 혼잡(Congestion)으로 간주하기 때문에 무선링크에서 발생하는 전송 오류는 불필요한 혼잡 제어를 유발시키며, 이는 TCP의 성능과 효율성을 떨어뜨린다[5-8].

무선링크의 전송 오류로 인한 TCP 전송 성능 저하를 극복하기 위한 연구는 주로 셀룰러 네트워크와 애드혹 네트워크를 대상으로 수행되었다. 셀룰러 네트워크에서는 고정 호스트(FH, Fixed Host)와 기지국 사이는 유선으로 연결되고 기지국과 이동 호스트 간에는 하나의 무선링크로 연결된 구조를 갖는다. 셀룰러 네트워크 유형의 네트워크에서, 유선구역에 있는 송신자로부터 전송된 패킷이 무선링크에서의 전송오류로 인해 유실되는 경우 이 사실을 감추기 위한 여러 가지 방안이 제안되었다. I-TCP[5] 방식은 기지국에서 TCP 연결을 분리함으로써 유선부와 무선부에 두 별개의 TCP 연결을 각각 유지하여 TCP 트래픽을 전달한다. 이러한 접근 방법은 TCP의 종단간 제어 특성을 갖지 못한다. M-TCP[6] 역시 이러한 접근 방식에 해당한다. Snoop 모듈 방식[7]과 WTCP[8]는 기지국에 도착하는 패킷을 기지국에 저장한 후 이동 호스트에게 전달한다. 기지국은 이동 호스트가 성공적으로 수신한 것으로 확인되는 패킷은 기지국의 버퍼에서 삭제하고 해당하는 ACK를 송신자에게 전달한다. 기지국과 이동 호스트가 하나의 무선링크로 연결된 셀룰러 네트워크 유형을 전제로 제안된 Snoop 모듈 방식을 두 홉 이상의 무선링크로 구성된 경로에 적용할 경우 상류의 무선링크에서 발생한 전송오류를 효과적으로 극복하지 못하는 단점이 있으며 이에 대해서는 2.5절에서 상세히 설명한다. WTCP의 경우에는 상류의 무선링크에서 발생한 전송오류를 극복하는

기능이 결여되어 있기 때문에 WTCP를 두 홉 이상의 무선링크로 구성된 경로에 적용할 경우 상류 무선링크에서의 패킷 전송 오류는 결국 TCP 혼잡제어를 유발하는 문제점이 있다. 적응적 순방향 오류 수정기법이나 링크레벨 ARQ 기법을 이용하여 신뢰성 높은 링크 레이어 프로토콜을 사용하고자 하는 접근 방법에서는 무선링크에서 패킷 손실이 발생할 경우 수신자측에서 중복 ACK가 발생되고 이에 의한 송신자의 fast retransmission에 의해 TCP의 성능저하가 일어나는 문제점이 있다[9].

애드혹 네트워크에서는, 송신자와 수신자 또는 이들 간 경로를 구성하는 중간 라우터의 이동으로 무선링크의 전송 능력이 심각하게 나빠짐으로써 송신자와 수신자 간 경로가 단절되고 새로운 경로를 재설정하는 과정에서 발생하는 TCP 전송 성능 저하를 극복하는 방안이 대해 많은 연구가 수행되었으나[10-12] 애드혹 네트워크에서 송신자와 수신자 간 경로를 구성하는 무선링크에서의 전송 오류로 인한 TCP 전송 성능의 저하를 극복하는 방안이 대한 연구는 그다지 많이 수행되지 않고 있다. [13]에서는 네트워크 혼잡에 의한 패킷 손실과 무선링크 상에서의 패킷 손실을 구분할 수 있도록 하고 무선링크 상에서의 패킷 손실로 인해 송신자가 세 개의 중복 ACK를 수신하게 되면 TCP 혼잡제어가 동작하지 않도록 하기 위해 이 세 번째 중복 ACK를 TCP에 전달하지 않도록 하고 단순히 재전송하는 ATCP 기능을 송신자 측에서 사용하도록 하였다. 이 방식의 경우 수신자로부터 세 개의 중복 ACK를 수신한 후에야 무선링크 상에서의 전송오류로 인해 손실된 패킷을 송신자가 재전송할 수 있기 때문에 무선링크 상에서의 전송오류로 인해 손실된 패킷을 재전송할 때까지의 지연시간이 상대적으로 긴 단점이 있다. [14]에서는 송신자와 수신자 사이에 적절한 수의 프록시를 도입함으로써 송신자와 수신자 간 TCP 연결을 여러 개의 구간으로 분리하고 각 구간에서는 TCP를 사용하여 해당 구간에서의 패킷 전송 오류 문제를 해결하도록 하였다. 그리고 TCP의 종단간 제어특성을 유지하기 위하여 수신자가 전송하는 누적 ACK(cumulative ACK)를 송신자가 수신할 경우에만 수신자가 성공적으로 패킷을 수신하였다고 인정하는 Split TCP를 제시하였다. 이 방안은 상대적으로 보다 많은 수의 무선링크를 경유하는 TCP 연결이 무선링크상에서의 전송오류로 인해 겪는 TCP 전송 성능상의 불공평성을 해결하는 데는 매우 유리하다. 그러나 각 구간에서 TCP를 사용하기 때문에 각 구간에서 패킷 손실이 발생할 때 해당 구간

<표 1> 제안방안과의 주요 특성 비교

| 주요 비교 특성                     | 제안방안  | ATCP                      | Split TCP  | Snoop |
|------------------------------|-------|---------------------------|--|-------|
| 무선링크상의 전송 오류 영향 극복 소요 시간     | 매우 양호 | 보통                        | 양호   | 보통    |
| TCP 종단간 제어 특성                | 유지    | 유지                        | 유지   | 유지    |
| 무선링크 구간 마다 해당 방안의 기능 제공의 필요성 | 필요    | 불필요                       | 필요   | 필요    |
| 송신자나 수신자 측에 대한 기능 추가의 필요성    | 불필요   | 송신자측에 ATCP 기능을 제공할 필요가 있음 | 수신자측에 매 100개 정도의 패킷을 수신할 때 마다 누적 ACK를 발생하는 기능 필요 | 불필요   |

에서 혼잡제어가 발생하며 따라서 이 방안은 송신자와 수신자 간의 각 무선링크의 전송오류에 의한 패킷 손실이 송신자와 수신자 간 TCP 전송 성능에 미치는 영향을 줄이는 데는 많은 한계가 있다.

본 연구에서는 (그림 1)과 같이 무선부가 두 개 이상의 무선링크를 경유하는 TCP 연결에 대해, TCP의 중단간 제어 특성을 유지하면서 상류의 무선링크에서 손실된 패킷에 대한 정보를 신속하게 파악하고 손실된 것으로 판단된 패킷에 대해 신속히 재전송이 가능하도록 함으로써 무선링크에서 발생된 패킷의 손실로 인한 TCP 성능저하를 최대한 줄이는 방안을 제시하고자 한다. TCP의 중단간 제어 특성을 유지하면서 두 홉 이상의 무선링크를 경유하는 TCP 연결에 적용 가능한 방안과 본 연구에서 제안하고자 하는 방안의 주요 특성을 비교하면 <표 1>과 같다.

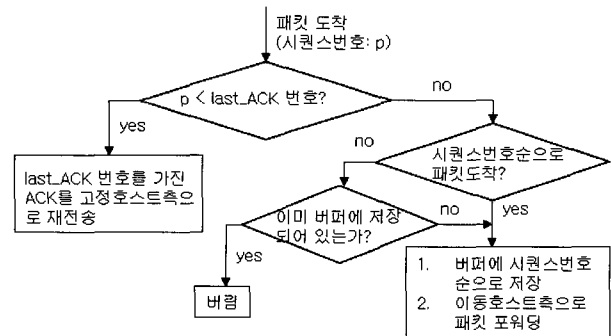
본 논문은 다음과 같이 구성되어 있다. 2장에서는 제안하는 방안의 동작 방식에 대해 설명한다. 3장에서는 모의실험 환경과 실험 결과를 보이며 4장에서 결론을 맺는다.

## 2. 제안된 알고리즘의 동작

유선부의 한 송신자가 이동 네트워크를 구성하고 있는 임의의 한 이동 호스트와 TCP 연결을 만든 상태라고 가정하자. 이 경우 이 연결의 무선부는 액세스 라우터와 1개 이상의 이동 라우터를 거쳐 이동 호스트에 연결되고 따라서 2개 이상의 무선링크를 거치게 된다. 본 논문에서는 액세스 라우터와 이동 라우터 사이의 무선링크, 이동 라우터와 이동 라우터 사이의 무선링크, 그리고 이동 라우터와 이동 호스트 사이의 무선링크에서 전송 오류에 의해 발생하는 패킷 손실에 대비해 재전송이 가능하도록 각 라우터에서는 수신한 패킷을 저장한 후 하류의 라우터나 이동 호스트에게 전송한다.

각 라우터에서는 수신한 패킷의 저장을 위한 순환 버퍼의 구현을 위해 두 개의 인덱스, 즉 `bufhead`과 `buftail`를 사용한다. `buftail`은 버퍼에 저장된 패킷 중 가장 작은 시퀀스 번호를 가진 패킷의 저장 위치를 가리키도록 한다. `bufhead`는 가장 큰 시퀀스 번호를 가진 패킷이 저장된 위치의 다음 비어있는 저장 위치를 가리키도록 한다. 따라서 패킷이 시퀀스 번호 순으로 도착하면 `bufhead`가 가리키는 위치에 저장된다. 패킷이 순서대로 도착하지 않으면 버퍼에 저장된 패킷을 재정렬해서 버퍼에 저장된 패킷이 시퀀스 번호순으로 저장되도록 한다. 여기서 `bufhead-1`에 저장된 패킷의 시퀀스 번호보다 큰 시퀀스 번호를 가진 패킷이 도착하면 시퀀스 번호 순으로 패킷이 도착하는 것으로 간주한다.

액세스 라우터나 이동 라우터는 이동 호스트나 하류의 이동 라우터가 전송한 ACK를 받게 되면 이동 호스트가 수신한 것으로 확인된 패킷을 해당 버퍼로부터 삭제한다. 그리고 이동 라우터는 ACK 번호 `a`와 자신이 저장하고 있는 패킷의 시퀀스 번호 중 가장 작은 시퀀스 번호 `b`를 비교한다. 만약 `b-a`가 양수이면 자신과 상류 라우터 사이의 무선링크에서 `b-a`개의 패킷이 손실된 것으로 판단하며 해당 라우터는 ACK를 상류 라우터에게 전송할 때 이 정보를 TCP 헤



(그림 2) 고정 호스트로부터 액세스 라우터나 이동 라우터에 도착하는 데이터 패킷의 처리 흐름도

더의 예비 필드에 기록해서 함께 보낸다. 이는 상류의 무선링크에서 손실된 패킷에 대한 정보를 신속하게 파악하고 손실된 것으로 판단된 패킷에 대해 신속히 재전송이 가능하도록 하는 역할을 한다. 이 패킷 손실 정보를 수신한 상류의 라우터는 손실된 것으로 판단된 패킷을 재전송한다.

제안한 알고리즘의 동작 과정은 고정 호스트로부터 액세스 라우터나 이동 라우터에 도착하는 데이터 패킷의 처리와 액세스 라우터나 이동 라우터에 도착하는 ACK의 처리 두 과정으로 크게 나눌 수 있으며 자세한 동작 과정은 다음과 같다. 아래의 설명에서 `last_ACK` 번호는 각 라우터에서 고정 호스트측으로 보낸 ACK 번호 중 가장 큰 값을 나타낸다.

### 2.1 고정 호스트로부터 액세스 라우터나 이동 라우터에 도착하는 데이터 패킷의 처리

라우터에 도착하는 데이터 패킷은 다음과 같이 네 가지의 경우로 분류되며 각 경우에 대한 처리 방식은 다음과 같다. 그리고 (그림 2)는 데이터 패킷의 처리 흐름도를 나타낸다.

**[경우 1] 버퍼가 비어 있거나 순서에 맞는 패킷이 도착 (새로운 패킷의 도착)**

`bufhead`가 가리키는 위치에 해당 패킷을 저장하고 이동 호스트 방향으로 패킷을 전송한다. 저장 후 라우터는 `bufhead`를 1만큼 증가시킨다.

**[경우 2] `last_ACK` 번호 보다 더 작은 시퀀스 번호를 가진 패킷이 도착**

이 경우는 라우터가 이전에 보낸 ACK가 분실된 경우이다. 라우터는 다시 한번 ACK를 고정 호스트에게 전송한다. 라우터는 수신한 패킷을 버린다.

**[경우 3] 이미 저장되어 있는 데이터 패킷이 도착**

이 경우는 고정 호스트에서 재전송이 발생한 경우이다. 라우터는 수신한 패킷을 버린다.

**[경우 4] 버퍼에 저장되지 않은 패킷이 순서대로 들어오지 않은 경우.**

라우터는 도착한 패킷을 이동 호스트 방향으로 전송하고, 시퀀스 번호 순으로 저장한다. 이를 위해 패킷을 저장할 위치를 찾은 후 버퍼에 저장된 패킷을 이동시켜 새로 들어온 패킷을 적절한 위치에 저장한다. 물론 이때 `bufhead`는 1만큼 증가된다.

2.2 하류의 이동 라우터로부터 액세스 라우터나 이동 라우터에 도착하는 ACK의 처리

다음과 같이 세 가지의 경우로 나누어서 ACK를 처리하며 (그림 3)은 그 처리 흐름도를 나타낸다.

[경우 1] 새로운 ACK가 도착

패킷 손실 정보의 유무를 확인한다. 패킷 손실에 대한 정보가 있는 경우, 손실된 패킷을 이전에 재전송한 적이 있는지 확인한다. 만약 재전송한 적이 없으면 ACK를 수신한 라우터와 하류의 이동 라우터 사이에서 패킷이 손실된 것으로 판단해서 재전송한다. 재전송시 재전송 타이머를 설정해서 재전송된 패킷 중 손실되는 패킷에 대한 재전송이 가능하도록 한다. 재전송한 적이 있는 경우에는, 재전송 타이머가 완료되지 않은 경우에는 재전송한 패킷이 아직 이동 호스트에 도착하지 않은 것으로 판단하고 재전송하지 않으며 재전송 타이머가 완료된 경우에는 재전송한다.

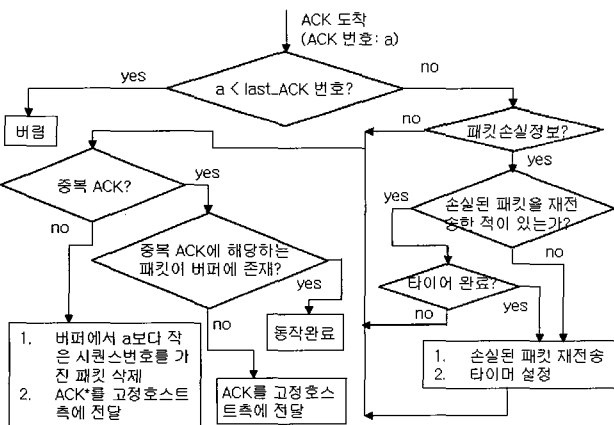
재전송에 대한 처리가 끝나면 버퍼에 저장된 패킷 중 ACK의 번호보다 작은 시퀀스 번호를 가지고 있는 패킷을 지우고 ACK를 고정 호스트에게 전송한다. 단, ACK를 수신한 라우터가 이동 라우터인 경우에는 현재 자신의 버퍼에 저장된 패킷에 대한 시퀀스 번호 중 가장 작은 시퀀스 번호  $b$ 와 수신한 ACK의 번호  $a$ 를 비교한다. 두 번호의 차,  $b-a$ 를 해당 라우터와 상류 라우터 사이에서 발생한 패킷 손실 정보로 사용한다.  $b-a$ 가 양수이면 이 값을 패킷 손실 정보로 하여 ACK와 함께 상류의 라우터로 전송한다.

[경우 2] 받은 ACK의 번호가 last\_ACK 번호보다 작은 경우

액세스 라우터는 ACK를 고정 호스트에게 전달하지 않고 제거한다.

[경우 3] 중복 ACK의 도착

수신한 중복 ACK에 포함되어 있는 패킷 손실 정보를 확인한 후 경우 1)에서와 같은 방식으로 재전송에 대해 처리한다. 그리고 라우터의 버퍼에 중복 ACK의 번호에 해당하는 패킷이 없으면 ACK를 고정 호스트 방향으로 전달한다. 그렇지 않으면 라우터가 이미 고정 호스트 방향으로 해당하는 ACK 정보를 전달했으므로 ACK는 전달하지 않는다.



(그림 3) 하류의 이동 라우터로부터 액세스 라우터나 이동 라우터에 도착하는 ACK의 처리 흐름도. 이동 라우터에 도착하는 ACK의 처리인 경우, ACK\*에는 손실정보가 포함됨

2.3 이동 호스트로부터 이동 라우터에 도착하는 ACK의 처리

이동 라우터는 받은 ACK의 종류와 패킷 손실 정보에 따라 다음과 같이 동작하며 그 처리 흐름도는 (그림 4)와 같다.

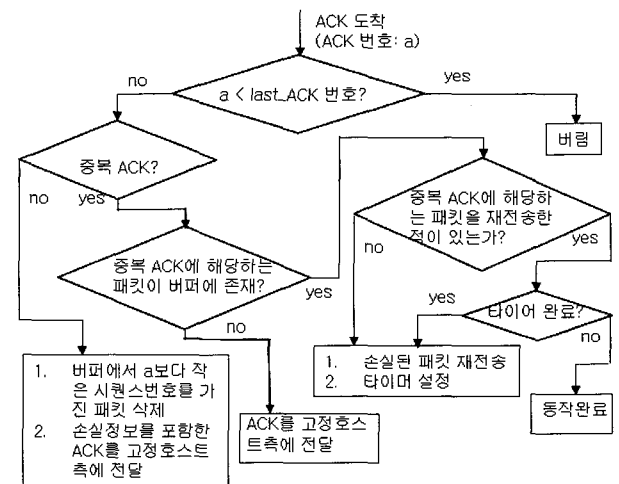
[경우 1] 새로운 ACK가 도착

이동 라우터는 버퍼에 저장된 패킷 중 ACK의 번호보다 작은 시퀀스 번호를 가지는 패킷을 지우고 ACK를 상류의 라우터에게 전송한다. 이때 이동 라우터는 현재 자신의 버퍼에 저장된 패킷의 시퀀스 번호 중 가장 작은 시퀀스 번호  $b$ 와 ACK의 번호  $a$ 를 비교한다. 두 번호의 차,  $b-a$ 를 해당 라우터와 상류의 라우터 사이의 무선링크에서 발생한 패킷 손실 정보로 사용한다.  $b-a$ 가 양수이면 이 값을 손실된 패킷 손실 정보로써 ACK에 포함되어 상류의 라우터에게 전송된다.

[경우 2] 받은 ACK의 번호가 last\_ACK 번호보다 작은 경우 ACK를 제거한다.

[경우 3] 중복 ACK의 도착

이동 라우터의 버퍼에 중복 ACK의 번호에 해당하는 패킷이 없으면 ACK를 고정 호스트 방향으로 전달한다. 그렇지 않으면 이동 라우터가 이미 고정 호스트 방향으로 해당하는 ACK 정보를 전달했으므로 ACK는 전달하지 않으며 이동 라우터는 해당 패킷을 이전에 재전송한 적이 있는지 확인한다. 만약 재전송한 적이 없으면 이동 라우터와 이동 호스트 사이에서 패킷이 손실된 것으로 판단해서 재전송한다. 재전송시 이동 라우터는 재전송 타이머를 설정해서 손실되는 패킷에 대한 재전송이 가능하도록 한다. 재전송한 적이 있는 경우에는, 재전송 타이머가 완료되지 않은 경우에는 재전송한 패킷이 아직 이동 호스트에 도착하지 않은 것으로 판단하고 재전송하지 않으며 그렇지 않은 경우에는 재전송한다.



(그림 4) 이동 호스트로부터 이동 라우터에 도착하는 ACK의 처리 흐름도

2.4 재전송 타이머

액세스 라우터와 이동 라우터에서는 재전송되지 않은 패킷에 대해 rtt(round trip time)를 측정하고 이를 바탕으로 식 (1)과 같이 srti(smoothed round-trip time)를 계산한다.

$$srtt = (1 - \alpha) \times srtt_{old} + \alpha \times rtt_{curr} \quad (식1)$$

여기서  $\alpha = 0.25$ ,  $srtt_{old}$ 는 기존의  $srtt$ 를,  $rtt_{curr}$ 는 현재 측정된  $rtt$ 를 나타낸다. 재전송 타이머는  $srtt$ 를 바탕으로 하여 설정한다. 데이터 패킷 전송 후  $srtt$ 의 두 배의 시간이 지나도록 ACK를 받지 못하면 재전송 타이머가 만료됨으로써 해당 패킷을 재전송한다.

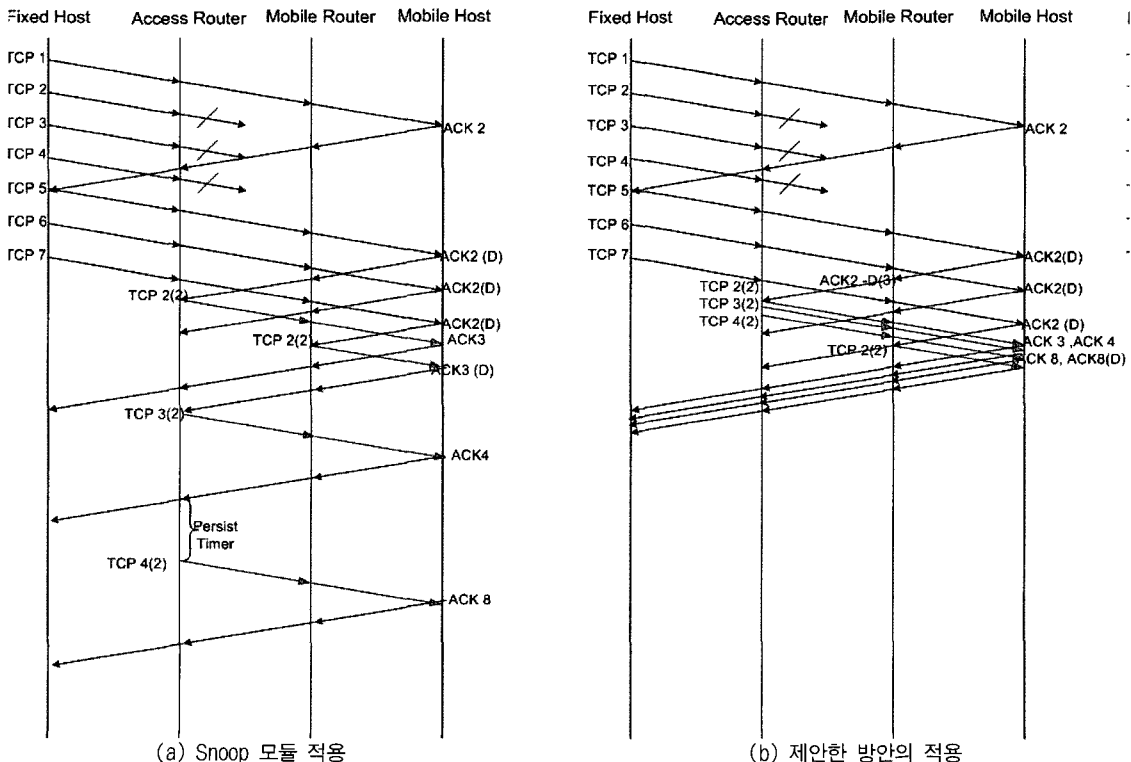
2.5 제안 알고리즘의 동작 예

(그림 5)는 액세스 라우터와 이동 호스트가 하나의 이동 라우터를 통해 경로가 설정된다고 가정하고 액세스 라우터와 이동 라우터 사이에서 2번, 3번 그리고 4번의 패킷이 연속적으로 분실되는 경우에 대해 제안하는 알고리즘의 동작 과정과 snoop 모듈 방식을 적용하였을 경우에 대한 동작 과정을 보이고 있다. (그림 5)에서 ACKs는 ACK 번호가 s인 ACK, ACKs(D)는 중복된 ACKs, ACKsD(L)은 패킷손실정보 L이 포함된 ACKs(D), 그리고 TCPs(n)은 해당 노드에서 패킷 시퀀스 번호가 s이고 n번째 전송되는 패킷을 각각 나타낸다.

제안하는 방안의 동작 과정은 다음과 같다 ((그림 5) (b) 참조). ACK2를 받은 이동 라우터는 버퍼로부터 첫 번째 패킷을 삭제한 후 ACK2의 번호와 버퍼에 저장된 패킷의 시퀀스 번호 중 가장 작은 시퀀스 번호를 비교한다. 이동 라우터는 액세스 라우터와 이동 라우터 사이에서 패킷 손실이 없는 것으로 판단하게 되고 따라서 패킷 손실 정보를 0으로 표시해서 액세스 라우터에게 ACK2를 전송한다. 액세스 라우터는 ACK2를 받아서 첫 번째 패킷을 자신의 버퍼에서

제거한다. ACK2(D)를 받은 이동 라우터는 현재 자신의 버퍼에 두 번째 패킷이 없으므로 액세스 라우터에 전송한다. 전송하기 전 이동 라우터는 ACK의 번호와 자신이 저장하고 있는 패킷의 시퀀스 번호 중 가장 작은 시퀀스 번호와의 차이를 액세스 라우터와 이동 라우터 사이의 무선 링크에서의 패킷 손실 개수로 판단한다. 따라서 중복 ACK2에 손실 정보 3을 포함시킨 ACK2D(3)를 전송한다. 중복 ACK2를 받은 액세스 라우터는 두 번째 패킷을 손실로 판단, 재전송한다. 손실정보가 3이므로 연속된 패킷 손실로 판단하고 세 번째 패킷과 네 번째 패킷을 전송한다. 다음 ACK2(D)를 수신한 이동 라우터는 ACK를 고정 호스트 방향으로 전달한다. 다음 ACK2(D)를 수신한 이동 라우터는 전송한 패킷이 손실된 것으로 판단, 재전송한다. 그러나 이 경우는 불필요한 재전송에 해당된다.

액세스 라우터와 이동 라우터에 snoop을 사용하는 경우의 동작 과정은 다음과 같다((그림 5)(a) 참조). ACK2를 받으면 이동 라우터와 액세스 라우터는 첫 번째 패킷을 삭제한다. 이동 라우터가 중복 ACK2를 받으면 이동 라우터는 현재 자신의 버퍼에 두 번째 패킷이 없기 때문에 해당 ACK2를 액세스 라우터에게 전송한다. 처음으로 중복 ACK2를 받은 액세스 라우터는 패킷 손실을 알고 두 번째 패킷을 재전송한다(TCP2(2)). 그 후에 도착한 중복 ACK2는 무시한다. 일곱 번째 패킷이 이동 호스트에 도착함으로써 생성된 중복 ACK2를 받은 이동 라우터는 액세스 라우터에서 재전송한 두 번째 패킷이 손실된 것으로 판단해서 재전송한다. 이 경우는 불필요한 재전송에 해당된다. 이동 라우



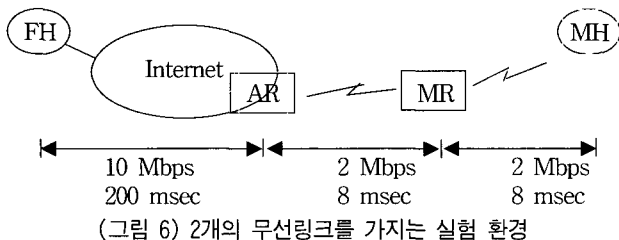
(a) Snoop 모듈 적용  
(b) 제안한 방안의 적용  
(그림 5) Snoop과 제안 모듈에서 연속된 패킷 손실에 대한 처리 비교

터가 재전송한 두 번째 패킷에 의해서 중복 ACK3이 전송된다. 중복 ACK3을 받은 이동 라우터는 현재 자신이 세 번째 패킷을 가지고 있지 않기 때문에 중복 ACK3을 액세스 라우터에게 전송한다. 중복 ACK3을 받은 액세스 라우터는 손실로 판단 재전송한다. 액세스 라우터가 재전송한 세 번째 패킷에 의해서 ACK4가 전송된다. ACK4가 전송된 후 액세스 라우터에서는 중복 ACK도 오지 않고 재전송된 경우도 없기 때문에 persist 타이머가 만료됨으로써 재전송된다. 이와 같이 snoop 방식에서는 상류의 무선링크에서의 패킷 손실 상태를 극복하는 기능이 미약하기 때문에 제안한 방안과 비교할 때 액세스 라우터와 이동 라우터 간 패킷 전송 오류를 극복하는 데 보다 많은 시간이 요구된다.

### 3. 성능 평가

#### 3.1 모의실험 모델

모의실험에 사용된 기본 모델은 (그림 6)과 같은 구성을 가지며 2개 이상의 무선링크를 경유하여 액세스 라우터와 이동 호스트간 경로가 형성되도록 하였다. 고정 호스트와 액세스 라우터 사이의 유선링크는 10 Mbps의 이더넷으로 가정하고 전송지연은 200 msec로 하였다. 액세스 라우터와 이동 라우터 그리고 이동 라우터와 이동 호스트 사이의 무선링크는 2 Mbps의 IEEE 802.11 무선랜으로 가정하고 전송지연은 8 msec로 하였다.



고정 호스트와 이동 호스트는 일반적으로 사용하는 TCP Reno를 사용하였으며 최대 TCP 세그먼트의 크기는 536 바이트로 일반적인 TCP의 설정을 따랐으며 주요 모의실험 파라미터는 <표 2>와 같다. 본 모의실험에서는 고정 호스트가 1 Mbyte의 정보를 전송하며 따라서 1,956개의 TCP 세그먼트가 생성된다. 액세스 라우터와 이동 라우터에서 버퍼 오버플로우가 발생하지 않도록 버퍼 크기는 충분히 크게 하였다. 제안하는 방안의 성능을 실험하기 위하여 network simulator v.2.26을 사용하였으며 무선 링크 오류의 연직성을 모델링하기 위해 two-state markov 모델을 사용하였다. 이 모델에서 무선채널이 양호 상태에 머무를 때는 BER이  $10^{-6}$ , 불량 상태에 머무를 때에는 BER이  $10^{-2}$ 인 것으로 가정한다. 양호 상태에 머무르는 시간은 평균 1초의 exponential 분포를 따르고, 불량 상태에서는 평균  $x$  msec의 exponential 분포를 따른다. 모의 실험에서는 평균불량기간

(MBP, Mean Bad Period)  $x$ 를 4 msec에서 128 msec까지 변화시킴으로써 <표 3>에서와 같이 무선링크에서의 BER이  $10^{-5}$ 에서  $10^{-3}$ 까지 변화도록 하였다. 그리고 제안하는 방안과의 성능 비교를 위해 <표 1>에서 보인 접근 방안 중에서 수신자나 송신자 측에 별도의 기능 추가가 요구되지 않는 snoop 방식과 액세스 라우터와 이동 라우터에서 전송오류에 대한 복구 기능을 전혀 제공하지 않는 방식(TCP Reno)에 대한 모의실험을 수행하였다.

<표 2> 주요 모의실험 파라미터

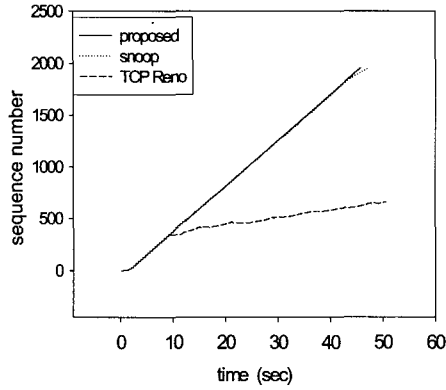
| Parameter                              | Value       |
|--|-------------|
| Max Window size                        | 20 segments |
| Buffer size at access router           | 2000 bytes  |
| Buffer size at mobile router           | 2000 bytes  |
| TCP Payload                            | 536 bytes   |
| TCP Segment Size (including header)    | 556 bytes   |
| IP datagram Size (including header)    | 576 bytes   |
| Ethernet Frame Size (including header) | 586 bytes   |
| Wired link BER                         | 0.0         |
| Wireless link BER in Good state        | $10^{-6}$   |
| Wireless link BER in Bad State         | $10^{-2}$   |
| Mean Good Period                       | 1 sec       |

<표 3> MBP에 따른 무선 링크의 평균 BER

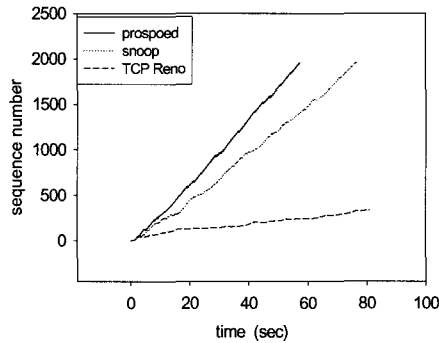
| Mean Bad Period (MBP) | Average BER |
|-----------------------|-------------|
| 4 ms                  | 4.0E-05     |
| 32 ms                 | 3.0E-04     |
| 128 ms                | 1.1E-03     |

#### 3.2 실험 결과

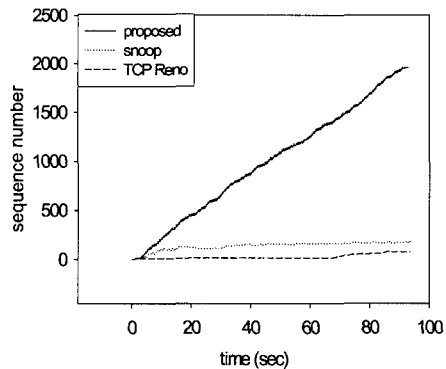
(그림 7)부터 (그림 10)은 액세스 라우터와 이동 호스트 간 경로가 2개의 무선링크로 구성되어 있는 경우에 대한 성능을 나타내고 있다. (그림 7)은 MBP가 4 msec(평균 BER =  $4 \times 10^{-5}$ ) 일 때 이동 호스트에서의 수신 패킷 시퀀스 번호의 증가 상태를 보이고 있으며 이를 통해 제안한 방식이나 snoop 모듈의 시퀀스 증가 속도가 이러한 방안을 적용하지 않은 순수한 TCP Reno에 비해 크게 향상됨을 알 수 있다. (그림 7)에서와 같이 BER이 상당히 낮을 때는 1Mbyte를 전송하는 데 소요되는 시간은 제안한 방식을 사용할 경우 45.73초, snoop 모듈을 사용할 경우 47.08초로 이들 두 방안의 성능차이는 크지 않다. 그러나 BER이 증가되면 상황은 달라진다. (그림 8)과 (그림 9)는 평균 BER을  $3 \times 10^{-4}$  과  $1.1 \times 10^{-3}$ 으로 증가시킬 경우 제안한 방식과 snoop 모듈을 사용하는 경우와 비교할 때 시퀀스 번호 증가 속도 차이가 점점 더 많이 커짐을 보이고 있다. 이는 (그림 5)에서 설명하였듯이 액세스 라우터와 이동 라우터 간 무선링크에서의 연속된 패킷 손실을 제안한 방안이 보다 효과적으로 복구하기 때문이다. (그림 10)은 이들 방식에 대한 TCP 처리율을 보이고 있다.



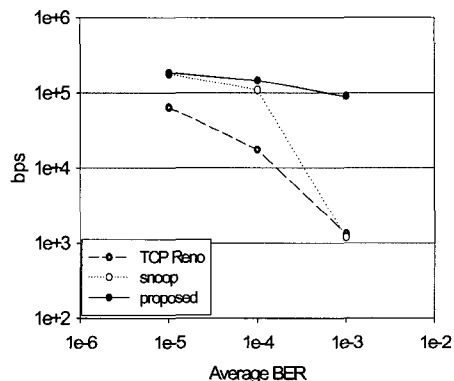
(그림 7) 이동 호스트에 도착한 시퀀스 번호 (MBP = 4 msec, 두 개의 무선링크)



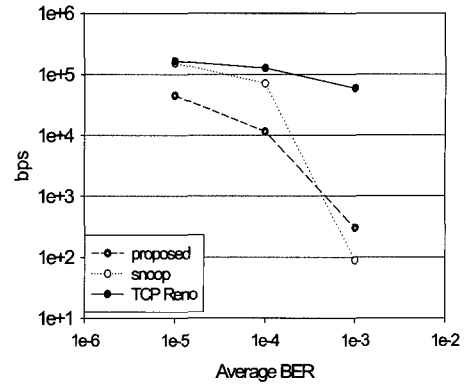
(그림 8) 이동 호스트에 도착한 시퀀스 번호 (MBP = 32 msec, 두 개의 무선링크)



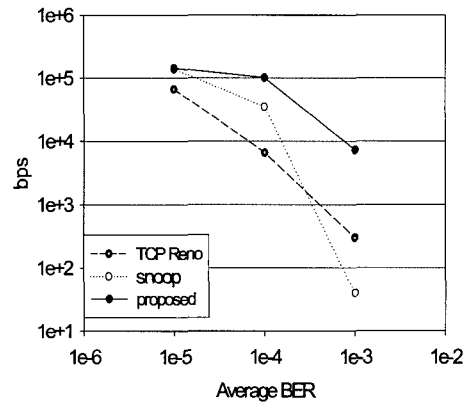
(그림 9) 이동 호스트에 도착한 시퀀스 번호 (MBP = 128 msec, 두 개의 무선링크)



(그림 10) BER에 따른 throughput 비교 (두개의 무선 링크)



(그림 11) BER에 따른 throughput 비교 (세개의 무선링크)



(그림 12) BER에 따른 throughput 비교 (네개의 무선링크)

(그림 11)과 (그림 12)는 3개의 무선 링크와 4개의 무선 링크를 가지는 경우에 대한 TCP 전송률을 보이고 있으며 무선링크의 개수가 2개인 경우와 비슷한 경향을 보이고 있다. (그림 10)과 (그림 11) 그리고 (그림 12)를 살펴보면 주어진 BER 값에 대해 무선링크의 개수가 증가함에 따라 TCP 처리율이 감소함을 알 수 있다. 이는 2개 이상의 이동 라우터를 거쳐 액세스 라우터와 이동 호스트가 연결되는 경우에는 액세스 라우터와 이동 호스트 간 무선링크의 수가 3개 이상으로 증가하며 따라서 무선링크에서의 패킷전송 오류 가능성이 이에 비례해서 증가하고, 더욱이 2단 이상의 상류 무선링크에서의 패킷 전송 오류를 극복하는 데는 더욱 많은 시간이 요구되기 때문이다.

#### 4. 결 론

본 논문에서는 단대단 경로가 여러 개의 무선링크로 구성될 수 있는 네트워크 이동 환경에서 상류 무선링크에서의 전송 오류로 인한 패킷 손실에 대한 정보를 제공하여 상류의 액세스 라우터나 이동 라우터에서 신속히 손실된 패킷을 재전송할 수 있도록 함으로써 무선링크에서의 전송 오류로 인한 TCP 성능 저하 문제를 해결하는 한 방안을 제시하였다. 제안한 방안은 TCP의 종단간 제어 특성을 유지하고 있으며 특히 송신자나 수신자 측에 별도의 기능 추가가 요구되지 않기 때문에 TCP를 사용하는 송신자나 수신자에게 아

무런 영향없이 제안하는 방안을 적용할 수 있는 장점이 있다. 네트워크 이동 환경의 특성 상 액세스 라우터와 이동 호스트 사이에 있는 무선링크의 개수가 3개를 넘지 않는 경우가 일반적이라고 예상되나 무선링크의 개수가 증가할 경우 제안한 방안의 TCP 전송 성능은 이에 비례해서 감소하게 된다. 이와 같이 무선링크의 개수가 많아짐으로써 야기되는 TCP 성능 저하 문제를 해결하기 위해서는 애드혹 네트워크 환경에서 사용되도록 제안되고 있는 Split TCP와 같은 방식을 대안으로 사용할 수도 있을 것이다.

### 참 고 문 헌

[1] J. Jeong, K. Lee, J. Park, and H. Kim, "Route Optimization based on ND-Proxy for Mobile Nodes in IPv6 Mobile Networks," *Proc. of VTC 2004-Spring, Vol.5*, pp.2461~2465, May 2004.

[2] T. Ernst, "Network Mobility Support Goals and Requirements," draft-ietf-nemo-requirements-05, October 24, 2005.

[3] T. Ernst, K. Mitsuya, and K. Uehara, "Network Mobility from the Internetcar Perspective," *Journal of Interconnection Networks*, Vol.4, No.3, pp.329~344, September 2003.

[4] E. Perera, V. Sivaraman, and A. Seneviratne, "Survey on network mobility support," *ACM SIGMOBILE Mobile Computing and Communications Review*, pp.7~19, 2004.

[5] A. Bakre and, B.R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts," *Proc. 15th International Conference on Distributed Computing Systems*, pp.136~143, 1994.

[6] K. Brown and S. Singh, "M-TCP: TCP for Mobile Cellular Networks," *ACM Computer Communications Review*, Vol.27, No.5, pp.19~43, 1997.

[7] H. Balakrishnan and S. Seshan and R. H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," *ACM Wireless networks Journal*, Vol.1, No.4, pp.469~481, December 1995.

[8] K. Ratnam and I. Matta, "WTCP : An Efficient mechanism for Improving TCP Performance over Wireless Links," *Proc. of the Third IEEE Symposium on Computers and Communications(ISCC '98)*, pp.74~78, June 1998.

[9] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *IEEE/ACM Trans. on Networking*, Vol.5, No.6, pp.756~769, Dec. 1997.

[10] G. Holland and N. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," *Wireless Networks* 8, pp.275~288, 2002.

[11] X. Yu, "Improving TCP Performance over Mobile Ad Hoc Networks by Exploiting Cross-Layer Information Awareness," *Proc. MobiCom'04*, pp.231~244, 2004.

[12] D. Kim, Toh, C.-K. and Y. Choi, "TCP-BuS: Improving TCP performance in wireless ad hoc networks," *Proc. ICC 2000*, pp.1707~1713, June 2000.

[13] J. Liu and S. Singh, "ATCP: TCP for Mobile Ad Hoc Networks," *IEEE Journal on Selected Areas in Comm.*, Vol.19, No.7, pp.1300~1315, July 2001.

[14] S. Kopparty, S. V. Krishnamurthy, M. Faloutsos, and S.K. Tripathi, "Split TCP for Mobile Ad Hoc Networks," *Proc. GLOBECOM 2002*, pp.138~142, 2002.

[15] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The Impact of Multihop Wireless Channel on TCP Throughput and Loss," *Proc. IEEE INFOCOM 2003, Vol.3*, pp.1744~1753, 2003.



### 김 명 섭

e-mail : AnneKim@lge.com

2003년 한양대학교 전자컴퓨터공학부(학사)

2005년 서강대학교 컴퓨터학과(석사)

2005년 인티큐브 프로그래머

2006년~현재 LG전자 연구원

관심분야: Network Mobility, 텔레메틱스, 애드혹 네트워크



### 최 명 환

email : mchoi@sogang.ac.kr

1978년 서울대학교 전기공학과(학사)

1980년 한국과학기술원 전기 및 전자공학과(석사)

1990년 University of Massachusetts, Amherst, 전기 및 컴퓨터공학과(박사)

1980년~1984년 삼성전자 선임연구원

1989년~1991년 Integrated Network Corp, NJ, 연구원

1991년~1993년 삼성전자 수석연구원

1993년~현재 서강대학교 컴퓨터학과 교수

관심분야: 네트워크 및 이동통신망, 애드혹 네트워크, 이동멀티미디어 방송