

# 수직적 핸드오프에서의 TCP 성능향상을 위한 수신자기반 슬로우스타트

석우진\*, 이민선\*, 이만희\*\*

## Receiver-Initiated Slow Start for Improving TCP Performance in Vertical Handoff

Woojin Seok\*, Minsun Lee\*, Manhee Lee\*\*

### 요약

TCP의 성능은 연결된 송신자와 수신자간의 전송 경로상의 트래픽 혼잡정도에 의해 영향을 받는다. 트래픽이 급작스럽게 증가하여 혼잡이 발생하면, 이는 데이터 전송의 장애를 유발시킬 수 있다. 데이터 전송의 장애가 발생하여 전송 패킷의 손실이 발생하면, TCP는 슬로우스타트를 수행하여 전송 속도를 낮추어 전송 경로 상의 트래픽 혼잡을 줄이고자 한다. 이동 단말이 이기종 무선 네트워크 간에 수직적 핸드오프를 하게 될 경우, TCP는 트래픽의 급격한 변화를 경험하게 되고, 이에 대해 슬로우스타트를 수행하게 된다. 본 논문에서는 수직적 핸드오프 시에 수반되는 TCP 슬로우스타트를 좀 더 효율적으로 수행하는 방법을 제안한다. 제안하는 방안은 기존의 방식과는 다르게 TCP 수신자가 슬로우스타트를 강제하는 방법이다. 이는 TCP 송신자에서 트래픽 혼잡을 경험한 후 슬로우스타트를 수행하는 기존의 방법보다 성능이 뛰어나다. 향상된 성능에 대해서는 시뮬레이션 방법으로 측정하여 성능을 검증하였다.

**Key Words** : TCP, Vertical handoff, WLAN, Cellular network.

### ABSTRACT

The performance of TCP depends on the degree of traffic congestion between the sender and the receiver. The traffic could increase, and this causes congestion which may cause trouble in data transfer. Then, TCP tries to eliminate the trouble by reducing the transfer speed with slowstart scheme. When a mobile node moves over heterogeneous wireless networks, TCP experiences dramatic change of the amount of traffic, and it performs slowstart. In this paper, we propose the efficient scheme of TCP slowstart that should performs after vertical handoff. In this scheme, TCP receiver forces slowstart, which is different form normal schemes. Its performance is better than the normal schemes in that TCP sender experiences traffic congestion and performs slowstart. We perform simulation to measure and to verify the improved performance.

### I. 서론

TCP는 전송계층의 네트워크 프로토콜이며 이의 성

능은 종단간의 연결 경로상의 네트워크 상태에 의해서 결정된다. 전송 중인 다른 트래픽에게 혼잡영향을 주지 않고 높은 전송률로 전송하기 위해서 TCP는 전

◆ 주저자 : 한국과학기술정보연구원, 국가슈퍼컴퓨팅연구소, wjseok@kisti.re.kr, 정회원

\* 한국과학기술정보연구원, mlecoh@kisti.re.kr

\*\* 한남대학교, manheelee@hnu.ac.kr

논문번호 : KICS2013-02-086, 접수일자 : 2013년 2월 6일, 최종논문접수일자 : 2013년 8월 9일

송 속도를 종단간의 가용한 대역에 맞게 적절히 적응하여야 한다. 이것이 혼잡제어이며 슬로우스타트는 혼잡제어에서 사용되는 중요한 방법 중의 하나이다. 폭주된 트래픽의 전송을 즉각 중단하고 현재의 네트워크 상태에 적합한 전송률을 찾기 위해서 TCP는 전송률을 1에서부터 지수 승으로 증가시킨다<sup>11-31</sup>.

이동 네트워크 환경에서는 이러한 혼잡제어가 트래픽 혼잡으로 오인되는 무선 링크 단락을 감지할 수 있어야 한다. 이동 네트워크에서 종단간의 낮은 TCP 성능을 개선하기 위해서, 많은 TCP 개선안이 제안되었다. 제안된 TCP 개선안들은 대개 핸드오프 혹은 무선 링크 단락으로 인한 패킷 손실과 높은 전송지연에 대한 고려가 포함되어 있다<sup>4-81</sup>.

또한 가까운 미래에는 여러 타입의 무선 네트워크가 혼재한 이기종 무선 네트워크 환경에서는 이동 단말이 두 개 혹은 여러 개의 서로 다른 타입의 무선 네트워크 사이를 핸드오프 하게 된다. 이러한 핸드오프를 수직적 핸드오프라 한다<sup>9-131</sup>. 특히, WLAN과 셀룰러 네트워크가 중첩되어진 무선 네트워크 환경은, 수직적 핸드오프의 전형적인 환경이다. 셀룰러 네트워크와 WLAN은 가격, 성능, 접근성 등에 있어서 상호 보완적인 요소가 있어 상호간의 연동이 점점 많아지고 있다.

하지만 상호 연동으로 셀룰러 네트워크와 WLAN이 연동 환경에서 잘 동작하더라도, 이동 단말은 대역과 링크 전송 시간의 급격한 변화를 경험하게 된다. 이는 WLAN과 셀룰러 네트워크가 구조적으로 다르기 때문이다. 이는 많은 패킷 뒤섞임과 패킷 손실을 발생시키며, TCP 성능의 심각한 저하를 가져온다<sup>14-161</sup>.

수직적 핸드오프로 인한 성능저하를 줄이기 위하여, [17]는 TCP 송신자로 보내어지는 ACK 패킷 숫자를 조절하는 방법을 제안하였다. 이를 위하여, TCP 수신자는 수직적 핸드오프를 감지할 수 있어야 하고, 또한 새로이 접속되어진 무선 네트워크에서 급격하게 변화되어진 대역을 측정할 수 있어야 한다. 그리고 새로이 전송할 전송 속도를 결정하게 되고, 결정된 전송 속도로 인하여 ACK 개수를 조절할 수 있게 된다. 하지만, [17]는 종단간의 대역에 대한 측정값을 알고 있다고 가정하였다는 점에서 문제점으로 제기된다. 왜냐하면, 이 값은 수직적 핸드오프 시에 정확히 측정하는 것은 기술적으로 쉽지 않기 때문이다.

기존의 연구, [18], [19], [20]는 수직적 핸드오프 직후에 슬로우스타트의 사용하는 것을 제안하였다. 슬로우스타트는, 새로이 접속한 네트워크에서 빠르게 적응

하기 위한 좋은 방법이다. 하지만, 제안한 방법은 TCP 수신부에서 발생한 수직적 핸드오프에 대한 시그널을 송신자에게 전달하여야 한다. 이러한 점이 TCP 송신자와 수신자의 모두의 수정을 요구하게 되며, 이는 제안한 방법의 실제 구현과 배포에 문제점으로 지적된다. TCP를 수정하는 것이 아닌, 다른 네트워크 전송 장비를 수정을 제안하는 방법으로, WLAN의 AP(Access Point)에 일부 기능을 추가하는 방법도 제안되었다<sup>231</sup>.

기존의 연구, [21]은 슬로우스타트 방법을 사용하지 않고, 수신자에서 송신자로 가는 경로를 조절하여 송신자에서의 타임아웃이 발생하지 않도록 하는 제안 방법이다. 이렇게 함으로써 타임아웃으로 인한 성능저하를 막고자 하는 것이다. 하지만 [21]이 제안하는 방법은 무선 네트워크 간의 대역차이로 인한 성능저하는 극복할 수 없다.

기존의 연구, [22]에서 제안된 방법은 상방향 수직적 핸드오프 (Upward Vertical Handoff)의 성능을 개선하는 방법을 제안하였는데, 상방향 수직적 핸드오프에 대한 TCP 성능개선을 위해서, 역시 송신자와 수신자가 MIHF(Media Independent Handoff Function)를 인식하도록 수정하여야 하는 구현상의 단점이 있다.

본 논문에서는, 수직적 핸드오프에 대하여 성능저하를 줄이는 새로운 TCP를 제안하고자 한다. 제안된 TCP는 성능 면에서 우수하고 구현 면에서 실현가능하도록 하였다. 성능 면에서, 수직적 핸드오프 수행 후에, 슬로우스타트를 수행하여 새롭게 도착한 무선 네트워크에서의 가용한 대역폭에 대해서 빠르게 적응하고자 하였다 또한 수직적 핸드오프가 진행될 때, 전송을 중단하여 불필요한 트래픽 발생을 줄이고자 하였다. 슬로우스타트와 일시적 전송중단으로 수직적 핸드오프로 인한 전송성능 저하를 줄이며 또한 다른 트래픽의 전송성능 저하를 줄일 수 있다. 구현 면에서, TCP 수신자만을 수정하고자 하였다. 이동 네트워크 환경에서는 TCP 수신자가 동작하는 이동 단말에서 수직적 핸드오프를 탐지하는 것이 용이하다. 그래서 TCP 수신자에서 수직적 핸드오프를 탐지하고 슬로우스타트를 진행하고 한다. 원래 슬로우스타트는 TCP 송신자에서 수행되는 것이기 때문에, TCP 수신자가 슬로우스타트를 이끌어 내기 위해서 본 제안 방법에서는 ACK 전송을 일정시간동안 일부러 전송하지 않는 방법을 사용한다. 성능 검증을 위해서는, 핸드오프 시의 기존의 TCP와의 성능비교를 하였다. 성능측정을 위해서 NS2를 이용한 시뮬레이션 방법을 사용하였다.

## II. 제안하는 TCP: TCP-FS

이동 단말이 이동하여 새로운 무선 환경에 도착하였을 경우, 이동 단말은 이전의 무선 환경의 대역과 전송지연시간과는 전혀 다른 새로운 대역과 전송지연 시간을 경험하게 되며, 이전에 사용되던 TCP 변수들은 무의미하게 된다. 그 새로운 무선 환경의 대역과 전송지연시간에 적절히 빠르게 대응하기 위해서 수직적 핸드오프 이후에 슬로우스타트를 사용하는 방법을 제안한다.

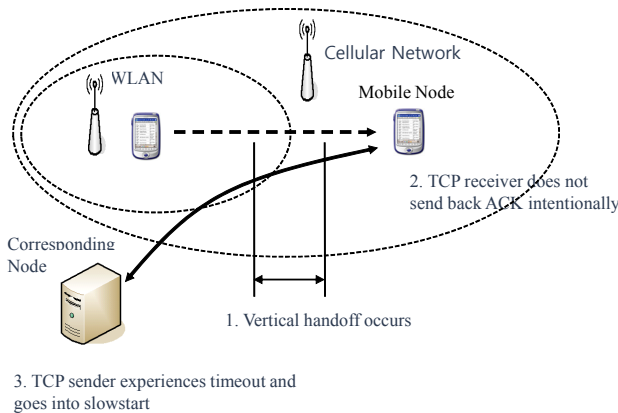


그림 1. TCP-FS 동작  
Fig. 1. The Operation of TCP-FS

제안하는 TCP를 TCP-FS(Forced Start)라 명명하며, 수직적 핸드오프 직후에, 수신부에 의해서 강제적으로 슬로우스타트를 수행된다는 의미이며, 수직적 핸드오프로 인한 TCP 성능 저하를 개선코자 한다. 이동 단말이 이동할 경우, 이동하였다는 사실은 TCP 수신측에서 탐지하게 되며, 이러한 사실을 기반으로 ACK 패킷을 일정시간 전송하지 않음으로써, TCP 송신자로 하여금 타임아웃이 발생하게 하여 슬로우스타트를 수행하게 한다.

TCP-FS는 TCP 수신자의 수정을 요구하며, TCP 송신자의 수정은 요구하지 않는다. 수신자만을 수정하는 방식으로 구현 및 배포를 용이하게 하고자 함이다. TCP는 송신자와 수신자가 쌍으로 연동되는 구조인데, 송신자와 수신자 모두를 교체하여야 한다면 구현 및 배포가 비현실적일 수밖에 없다. 수신자만을 교체하여야 한다면 이동 단말내의 TCP만 교체함으로써 쉽게 구현과 배포가 이루어질 것이다.

본 장에서는 TCP-FS의 구조에 대해서 설명하고자 한다. 이는 TCP 수신자의 구조에 해당하며, TCP 송신자는 기존의 TCP를 그대로 사용하게 된다. TCP-FS는 제안된 방법이 적용된 TCP 송신자와 수신자를 통

칭해서 일컫는다. 물론, TCP 송신자는 기존의 송신자를 그대로 사용하는 것이다.

### 2.1. TCP-FS의 동작

TCP-FS는 수직적 핸드오프 이후 TCP 송신자가 슬로우스타트를 수행하여 새롭게 이동한 네트워크의 새로운 상태에 빠르게 적응하도록 하는 것을 추구한다. 이를 위해서, TCP-FS는 그림에서 보듯이, 3 단계의 작업을 수행하여야 한다. 그림은 셀룰러 네트워크와 WLAN이 중첩되어 있는 상황에서, 이동 단말이 WLAN 에서 전송을 하다가 셀룰러 네트워크로 이동한 상태이다. 즉 수직적 핸드오프가 발생한 것이다. 수직적 핸드오프가 발생하면, TCP-FS는 3개의 단계를 수행한다.

첫 번째 단계로서는, 그림에서 (1), 핸드오프를 감지한다. 이것을 위해서 TCP-FS는 크로스레이어 방식으로 링크에서의 핸드오프를 직접 감지하여 핸드오프를 탐지하게 된다. 이동 단말은 여러 무선링크 인터페이스를 가지며, 각 무선 인터페이스에서 무선 AP(Access Point) 혹은 BS(Base Station)로부터의 시그널을 감지하여 핸드오프를 탐지할 수 있게 된다.

두 번째 단계로서는, 그림에서 (2), 핸드오프가 탐지되면, ACK 메시지를 'ack-holding' 시간동안 강제로 송신측으로 보내지 않는다. 이는 TCP 송신자에게 타임아웃을 유발시켜 강제로 슬로우스타트를 수행하기 위함이다. 그래서 새롭게 도착한 무선 네트워크의 새로운 네트워크 상황에 빠르게 적응하게 하기 위함이다. 이를 위해서 'ack-holding' 시간 계산이 요구된다.

세 번째 단계로서는, 그림에서 (3), TCP 송신측에서 ACK를 받지 못하게 되므로, 타임아웃이 발생하고 슬로우스타트를 수행하게 된다.

### 2.2. ACK-holding 시간 계산

ACK-holding 시간은 TCP 수신자가 일부러 ACK 메시지를 송신측에 보내지 않는 시간을 말한다. TCP 수신자가 ACK-holding 시간 동안 ACK를 보내지 않으면, 송신자는 타임아웃이 발생하고 슬로우스타트를 수행한다. TCP 수신자는 ACK-holding 시간 이후에 도착하는 데이터에 대해서 ACK를 전송함으로써 송신자에서 수행되는 슬로우스타트가 정상적으로 수행되게 한다.

TCP 송신자에서 타임아웃이 발생하기 위해서는, ACK-holding 시간이 RTO 시간 보다는 길어야 한다 (식1). 본 논문에서는 margin 값으로 0.1 X RTO 값으

로 한다.

$$\text{ACK-holding 시간} = \text{RTO} + \text{margin} \quad (\text{식1})$$

RTO 는 (식2)와 같이 계산되어 진다, 단  $\alpha$  와  $\beta$  는 0 과 1 사이의 상수이다.

$$\text{RTO} = \text{sRTT} + k \times \text{RTTvar} \quad (\text{식2})$$

$$\text{RTTvar} = (1-\beta) \times \text{RTTvar} + \beta \times \text{RTTdiff}$$

$$\text{RTTdiff} = |\text{sRTT} - \text{RTT}|$$

$$\text{sRTT} = (1-\alpha) \times \text{sRTT} + \alpha \times \text{RTT}$$

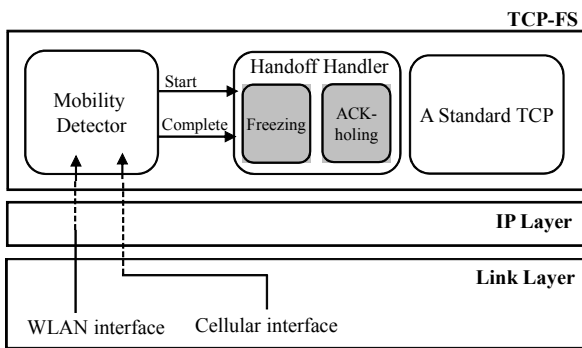


그림 2. TCP-FS 구조  
Fig. 2. The Structure of TCP-FS

### 2.3. TCP-FS 의 구조

TCP-FS는 Mobility Detector, Handoff Handler, 그리고 기존의 TCP 로 구성되어 있다. TCP 송신부분은 전혀 수정이 필요치 않으며, TCP 수신부분만 수정이 요구된다. 기존의 TCP를 교체하지 않고 Mobility Detector와 ACK-handler 부분만 추가하여 구성된다.

Handoff-Handler는 Freeze 기능과 Ack holding 기능을 수행한다. 핸드오프시의 데이터 전송은 트래픽 낭비를 유발하고 다른 트래픽에 방해할 수 있기 때문에 오히려 보내지 않는게 낫다. Freeze 기능은 핸드오프 시작이 탐지되면, 핸드오프가 시작되고 완료될 때까지 트래픽 전송을 잠시 중단하도록 TCP 송신부분에 전송을 멈추게 하는 기능이다. 이를 위해서 zero window를 사용한다.

Ack holding 기능은, 핸드오프가 완료되고 나서, 새롭게 진입된 무선 네트워크의 상태에 맞지 않는 TCP 전송을, 슬로우스타트를 통해서 새롭게 적응하게 하기 위해서 슬로우스타트를 강제로 수행하도록 ACK를 잠시 보내지 않게 만드는 것이다. Ack holding 시간은 (식1)과 같다.

Mobility Detector는 핸드오프가 시작되고 완료됨

을 Handoff Handler에게 알려준다. 핸드오프를 탐지하는 방법으로 크로스레이어 방법을 사용한다. 링크단의 인터페이스의 시그널을 직접 전달받아서 핸드오프가 시작되고 완료됨을 판단한다.

### III. 성능 평가

TCP-FS는 슬로우스타트를 강제로 빨리 유발시켜 새로운 망에 적응을 빨리하게 하는 방법이다. 하지만, ACK-holding 시간동안 패킷을 전송하지 못하게 된다. 이러한 특성에 대해서 전체적인 성능이 어떠한지 성능 평가를 하고자 한다.

TCP-FS의 성능평가를 위해서, 이기종 무선 네트워크가 중첩되어 있는 상황에서 수직적 핸드오프가 발생하였을 경우의 성능을 측정한다. ‘upward’ 수직적 핸드오프와 ‘downward’ 수직적 핸드오프를 구분하여 성능을 측정하였다. ‘upward’ 수직적 핸드오프는 상대적으로 낮은 대역으로의 핸드오프이며, ‘downward’ 수직적 핸드오프는 그 반대의 개념이다.

측정인자로써, CWND와 전송률을 사용하였다. CWND는 TCP에서 사용되는 파라미터로써 한 번에 보낼 수 있는 데이터사이즈를 의미하며 이는 곧 TCP의 전송속도를 의미한다. 그리고 전송률은 경과시간에 대한 TCP의 평균 전송 성능을 보여줄 것이다.

TCP-FS는 성능 면에서 우수하고 구현 면에서 실현 가능한 것을 추구하였다. 기존의 연구들 중에서 수직적 핸드오프 시에 발생하는 대역차이를 극복하고자 제안된 알고리즘들([17]-[22])은 구현 면에서 크게 2가지로 구분할 수 있다.

수신부/송신부 모두의 TCP를 수정을 요구하는 방법과 수신부의 TCP만을 요구하는 방법으로 나눌 수 있다. 수신부/송신부 모두의 TCP 수정을 요구하는 방법은 구현과 배포면에서 현실적으로 어려움이 있는 방법이다. 즉, 수신부와 송신부를 모두를 수정한다는 것은 존재하는 모든 TCP의 수정을 요구하는 것을 의미하기 때문이다. 반면에 수신부 TCP 수정을 요구하는 방법은 이와는 달리, 해당 노드의 TCP의 수정으로 성능 향상을 가져올 수 있기 때문에 구현과 배포에 큰 어려움이 없다. 통상, 이동 단말은 TCP 수신부에 해당하며, 대부분의 데이터 전송이 수신부에서의 다운로드(download)에 해당한다. 서버쪽에 해당하는 송신부 TCP의 수정 없이 이동 단말만의 수정을 통해서 성능 향상을 가져올 수 있다면 현실적으로 구현 및 배포가 가능하다고 할 수 있는 것이다.

[18]-[22]는 전자인 수신부/송신부 모두의 TCP 수

정을 요구하는 알고리즘으로 성능 면에서 유리하나 구현 면에서 다소 현실성이 떨어진다. [17]의 경우는 수신부를 수정하는 방법을 제안하고 있으나, 대역에 대한 측정에 대하여 많은 가정을 포함하여 역시 구현 면에서 현실성이 떨어진다. 정확한 대역을 측정하는 것은 어렵고 또한 측정 시간이 오래 걸리는 작업으로 단시간의 측정은 정확성 면에서 오류를 유발하나 [17]에서는 정확한 대역을 측정한다는 가정을 지니고 있어 다소 현실성이 떨어진다.

본 논문은 구현 면에서 현실성을 추구하는 방안을 제시하고 있으며 대역의 변화에 대해서 적절하게 대처하기 위해서 강제적인 슬로우 스타트를 사용하고자 하는 방안을 제안하는 것이며, 이러한 방안에 대한 성능 비교로써, 현재 범용적으로 사용되고 있는 TCP-newreno와 비교하고자 한다.

TCP-FS의 성능을 TCP-newreno와 비교하였다. TCP-newreno는 다수의 패킷 손실에 좋은 성능을 보여주어 수직적 핸드오프 시에 TCP-reno 보다 좋은 성능을 보여준다<sup>[18]</sup>.

### 3.1. 시뮬레이션 환경

이동 단말이 이기종의 무선네트워크의 중첩된 환경에서 이동하는 상황에서의 성능을 평가하고자 한다. 기존의 핸드오프와는 달리 수직적 핸드오프는 이기종 무선 네트워크의 중첩된 특성으로 소프트 핸드오프가 가능하다<sup>[16]</sup>. 소프트 핸드오프를 수행함으로써 이동간의 연결단락이 발생하지 않는다. 본 논문에서는 이기종 무선 네트워크로 WLAN과 셀룰러 네트워크가 중첩되어 있는 상황을 가정한다.

WLAN은 대역이 크고, 가격이 저렴한 장점이 있지만 반경이 작아서 이동 단말에 지속적인 무선 환경을 제공하기는 어렵다. 반면에 셀룰러 네트워크는 반경이 충분히 넓어서 이동 단말에게 광범위한 서비스를 제공해 줄 수 있다. 이동 단말이 셀룰러 네트워크에서 WLAN으로 이동할 경우에는 대역과 가격 측면에서 WLAN의 서비스를 우선적으로 받는 것이 유리할 것이다. 특히 높은 대역으로 TCP를 통한 데이터 전송 시에는 더욱 선호될 것이다. 이동 단말의 이동특성으로 WLAN의 지역을 벗어나게 되면, 셀룰러 네트워크의 접속을 유지하게 될 것이다.

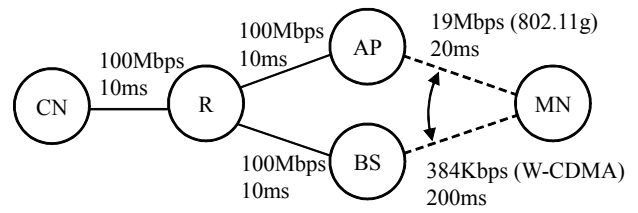


그림 3. 시뮬레이션 토폴로지  
Fig. 3. Simulation Topology

시뮬레이션 토폴로지는 그림 3과 같으며 NS2에 의해서 구현되었다. AP, BS, R은 각각 셀룰러 네트워크에서의 액세스포인트(Access Point), WLAN에서의 베이스스테이션(Base Station), 라우터(Router)를 각각 의미한다. MN과 BS사이의 링크는 3세대 셀룰러 네트워크의 한 타입인 W-CDMA(Wideband Code Division Multiple Access)를 표현한 것이며, MN과 AP는 WLAN의 한 타입인 802.11g를 표현한 것이며, 각각 384Kbps의 대역과 200ms의 전송지연시간, 19Mbps의 대역과 20ms의 전송지연시간을 가진다. 다른 링크들은 유선 상의 링크이며, 100Mbps의 대역과 10ms의 전송지연시간을 가진다. 이러한 시뮬레이션 환경은 [17], [18] 등과 같이 수직적 핸드오프에서의 TCP 성능 평가에 이미 많이 사용되는 방식이다.

MN은 CN으로부터 데이터를 다운로드 받는 상황이며, 이는 이동 네트워크 환경에서 가장 많이 발생하는 상황이다. 이때 TCP 연결이 지속되는 중에, 이동 단말이 이동하게 되어 무선 연결 링크를 변경하게 된다. 이는 수직적 핸드오프를 의미하게 된다. TCP 세그먼트의 최대세그먼트 크기는 1460 byte이다. 셀룰러 네트워크와 WLAN 네트워크에 40초씩 머문다고 가정하며 이동속도는 등속운동을 한다고 가정한다.

### 3.2. 시뮬레이션 결과 토의

그림 4는, 이동 단말이 upward 수직적 핸드오프 시에 TCP-FS와 TCP-newreno의 CWND의 추이를 보여준다. upward 수직적 핸드오프 시에는, 높은 대역과 저지연의 네트워크 환경에서 상대적으로 낮은 대역과 고지연의 네트워크 환경으로 이동하게 된다. 이럴 경우, CWND가 높은 상태에서 낮은 대역의 네트워크 환경으로 이동하게 때문에, 대량의 트래픽을 낮은 대역을 통해서 전송을 시도하게 되는 것이다. 그림 4에서 보듯이, TCP-newreno는 낮은 대역에서 높은 CWND로 대량의 트래픽을 계속 전송하고자 시도하는 가운데 타임아웃이 연속적으로 발생하게 되며, 또한 슬로우스타트 임계값은 최저값인 2로 수렴하게 된

다. 슬로우스타트 임계값이 2인 상태에서 슬로우스타트를 수행하게 되어서, 새로운 무선 네트워크의 대역까지 지수적인 증가를 하지 못하고 선형적 증가를 하게 되어 적절한 CWND 값을 가지기 까지 시간이 많이 소모된다.

반면, TCP-FS는 수직적 핸드오프가 발생한 후 일정시간 동안 트래픽을 임의로 보내지 않아 TCP 송신측의 슬로우스타트를 강제로 발생시킨다. 슬로우스타트 임계값이 유지되어 있는 상태에서 슬로우스타트가 수행되어 새로운 망에 빠르게 적응한다.

그림 5는 이동 단말이 downward 수직적 핸드오프 시에 TCP-FS와 TCP-newreno의 CWND의 추이 그래프를 보여 준다. downward 수직적 핸드오프 시에는, 낮은 대역과 고지연의 네트워크 환경에서 상대적으로 높은 대역과 저지연의 네트워크 환경으로 이동하게 된다. 이럴 경우, 전송상의 대역과 전송지연시간의 차이로 인하여 패킷 뒤섞임 현상이 발생하게 된다.

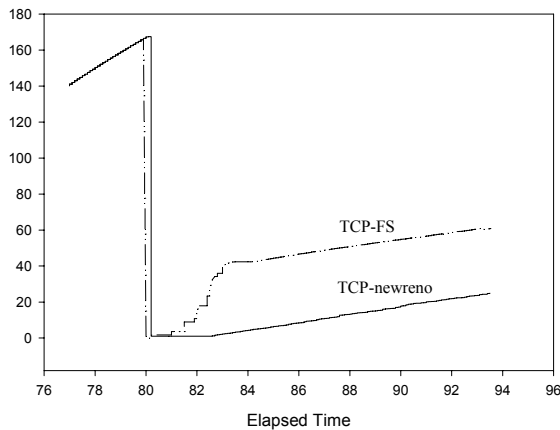


그림 4. CWND 추이 (upward 경우)  
Fig. 4. CWND Progress (upward case)

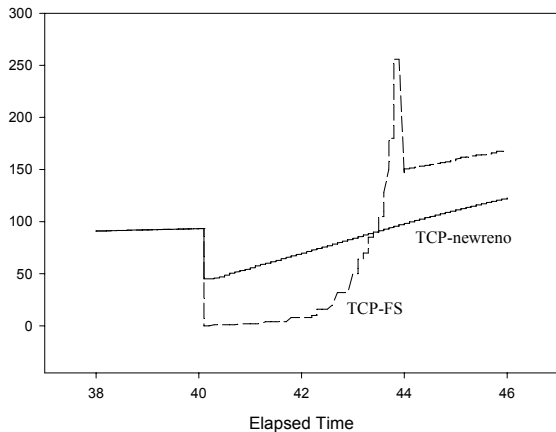


그림 5. CWND 추이 (downward 경우)  
Fig. 5. CWND Progress (downward case)

TCP-newreno는 연속으로 패킷 뒤섞임 현상이 발생하여 'fast retransmission'이 시도된다. 패킷 뒤섞임으로 인한 재전송이 발생하고 CWND 값은 절반으로 줄어들게 된다. 이후에 'congestion avoidance' 모드에서의 선형적 증가 상태로 새로운 고대역의 네트워크에 수렴하게 될 것이다.

TCP-FS는 패킷 전송을 잠시 멈추고 송신자에서의 타임아웃을 강제하므로 패킷 뒤섞임 현상이 발생하지 않는다. 수직적 핸드오프가 종료된 이후에 슬로우스타트가 수행되어 빠르게 새로운 무선 네트워크 환경에 적응하게 된다.

그림 6과 그림 7은 각각 upward 수직적 핸드오프와 downward 수직적 핸드오프 시의 전송되는 데이터의 일련의 바이트 스트림의 순차적 번호를 보여주고 있다. 그림에서 보듯이, downward 수직적 핸드오프와 upward 수직적 핸드오프 모두 TCP-FS의 CWND가 새로운 무선네트워크에 빠르게 수렴되기 때문에, 전송량의 증가가 새로운 네트워크에서 시작할 때부터 빠르게 증가한다. 반면, TCP-newreno는 핸드오프 이후 점점 증가속도가 빨라지지만 초반에는 다소 느리게 증가한다. 이는 CWND 값이 새로운 망에 최적화되어 수렴되어 있지 않기 때문이며 수렴하는 속도가 선형적 방식으로 증가하기 때문이다.

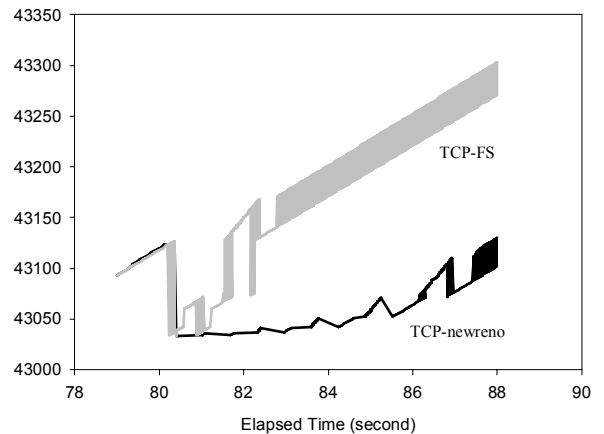


그림 6. 전송 데이터양의 추이 (upward 경우)  
Fig. 6. The Progress of Transferred Data Volume (upward case)

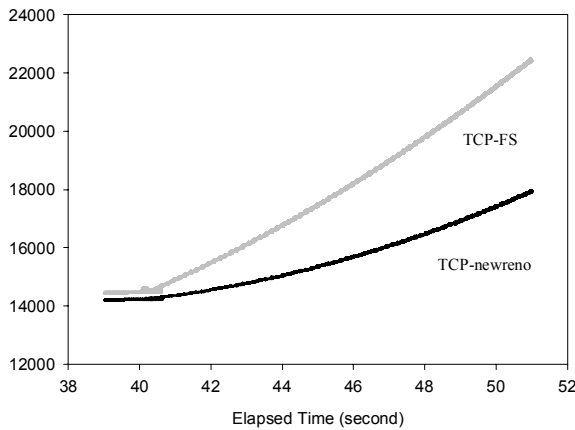


그림 7. 전송 데이터양의 추이 (downward 경우)  
Fig. 7. The Progress of Transferred Data Volume (downward case)

그림 8과 그림 9는, WLAN에서 제공받는 대역의 차이에 대한 이동 단말이 획득하는 전송률을 보여준다. 셀룰러 네트워크에서 제공받는 대역이 384Kbps 일 때, WLAN의 대역이 변화함에 따라, downward 수직적 핸드오프와 upward 수직적 핸드오프 시 이동 단말의 TCP에서의 성능 차이를 보여준다.

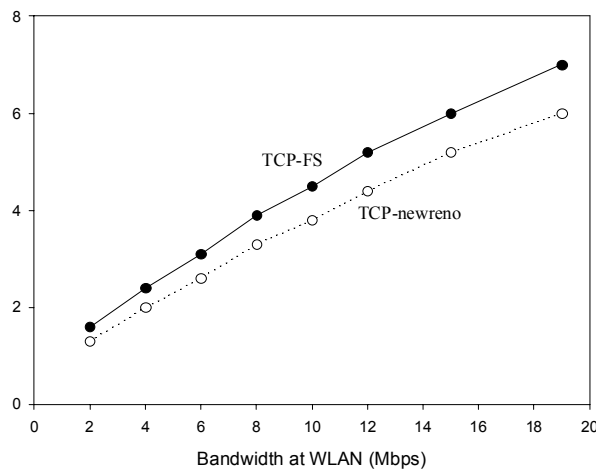


그림 8. 대역 변화에 대한 전송률(throughput)  
Fig. 8. Throughput for Various Bandwidths

그림 8에서 보는 것처럼, downward 수직적 핸드오프시의 TCP-FS와 TCP-newreno의 성능을 비교하였다. downward 수직적 핸드오프 시에는 WLAN에서 획득하는 대역이 많아짐에 따라 이동 단말의 전송률은 높아질 것이다.

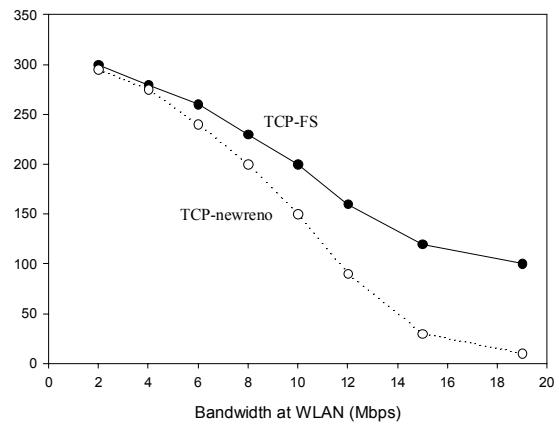


그림 9. 대역변화에 대한 전송률(throughput)  
Fig. 9. Throughput for Various Bandwidths

또한 그림 9에서 보는 것처럼, upward 수직적 핸드오프 시에는 WLAN에서 획득했던 대역이 높을수록 384Kbps 대역의 셀룰러 네트워크로 이동해 들어오면, 전송에 대한 트래픽 혼잡 정도가 높아져서 오히려 성능이 떨어진다. 모두의 경우에서, TCP-FS가 더 좋은 성능을 보여주며 대역의 차이가 클수록 더 큰 차이를 보여준다. 또한 downward 수직적 핸드오프보다 upward 수직적 핸드오프에 대해서 TCP-FS가 더 큰 성능을 보여준다.

특히, 그림 9의 upward 수직적 핸드오프의 경우, 갑자기 큰 전송률을 가지고 새로운 망에 들어가는 TCP-newreno의 경우 큰 혼잡으로 많은 손실을 가져온다. 대역의 차이가 클수록 이러한 혼잡의 정도가 크게 발생하므로 TCP-FS와의 성능차가 커지고 있다.

표 1은 대역 및 전송지연시간의 차이에 대해서 수직적 핸드오프 수행시간 과 ack-holding 시간동안 TCP-FS가 보내지 못하는 데이터양과 그동안 TCP-newreno가 보내는 데이터양과 그 이후에 보내는 데이터양을 비교한 것이다. 핸드오프를 거치는 동안의 시간과 핸드오프이후에 강제로 ACK를 보내지 않아 TCP 송신측에 슬로우스타트를 유발시키는 시간이다. 이 시간 동안 TCP-FS는 데이터를 보내지 않는다.

표 1. 데이터 전송량 비교  
Table 1. Transferred Data Volume Comparison

	The period that TCP-FS cannot send data	5 Seconds after the period
TCP-newreno	200Kbytes	1018Kbytes
TCP-FS	0	1387Kbytes

반면에 TCP-newreno는 패킷 뒤섞임 등을 경험하

여 'fast retransmission'을 경험하겠으나, 데이터를 보낼 수는 있다. 전송지연시간이나 대역의 차이가 증가할수록 패킷 뒤섞임 정도가 높아지거나 혼잡이 가중되어 전송 데이터량은 줄어드는 모습을 보여준다. TCP-FS가 데이터를 전송하지 못하는 동안, TCP-newreno가 데이터를 전송하더라도, 수직적 핸드오프 이후 TCP 송신측에서의 슬로우스타트가 시작되면 TCP-FS는 새롭게 도착한 무선 네트워크의 대역에 빠르게 적응하게 된다. 이러한 이유로 슬로우스타트 이후에 전송하는 데이터량은 그림에서 보듯이 TCP-FS가 더 많아진다. 결국 전체적인 전송량으로써, TCP-FS가 더 좋은 성능을 보여준다.

#### IV. 결 론

수직적 핸드오프는 갑작스런 대역과 전송지연시간의 변화로 인하여 TCP의 성능을 저하시킨다. 이러한 성능저하를 극복하기 위하여 본 논문에서는 수신자 기반의 슬로우스타트 방안을 제안하였다. 제안된 TCP는 성능 면에서 우수하고 구현 면에서 실현가능하도록 하였다.

TCP-FS는 수직적 핸드오프가 발생한 후 TCP 송신측의 슬로우스타트를 강제로 발생시키는 방법으로, 새로운 망에 빠르게 적응하도록 하였다. TCP 전송 속도를 나타내는 CWND에서 보듯이, TCP-FS는 TCP-newreno에 비해서 이동된 무선 네트워크의 대역에 빠르게 적응하였으며, 이로 인하여 전송되는 TCP 일련번호도 빠르게 증가함을 보였다. 특히 이러한 TCP-FS 특성으로, WLAN에서 획득되는 가용 대역이 클수록 TCP-FS는 TCP-newreno에 비해서 더 좋은 성능을 보여주었다.

이러한 성능의 향상은 수직적 핸드오프가 발생할 때 데이터를 보낼 수 있음에도 불구하고 보내지 않아 송신측의 슬로우스타트를 유발시켜 오히려 빠른 새로운 망에 적응하는 것이 더욱 좋은 성능을 보여주는 것이다.

TCP-FS는 수직적 핸드오프가 잦게 발생하는 이동 단말에서 좋은 성능을 보여줄 것으로 기대되며, TCP 수신자만 수정되는 구조를 가지고 있어, 구현 및 배포에도 용이할 것이다.

#### References

[1] J. B. Postel, *Transmission Control Protocol*, IETF RFC 793, Sep. 1981.

[2] P. Karn and C. Patridge, "Improving round-trip time estimates in reliable transport protocols," in *Proc. SIGCOMM '87*, pp. 2-7, Stowe, U.S.A., Aug. 1987.

[3] V. Jacobson and V. M. Karels, "Congestion avoidance and control," *ACM Comput. Commun. Review*, vol. 18, no. 4, pp. 314-329, Aug. 1988.

[4] A. Bakre and B. R. Badrinath, "Handoff and system support for indirect TCP/IP," in *Proc. Second Usenix Symp. Mobile Location-Independent Computing (MLICS '95)*, pp. 11-24, Ann Arbor, U.S.A., Apr. 1995.

[5] R. Caceres and L. Iftode, "Improving the performance of reliable transport protocols in mobile computing environments," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 5, pp. 850-857, June 1995.

[6] H. Balakrishnan, S. Seshan, E. Amir, and R. H. Katz, "Improving TCP/IP performance over wireless networks," in *Proc. ACM Annu. Conf. Mobile Computing Networking (MobiCom '95)*, pp. 2-11, Berkeley, U.S.A., Nov. 1995.

[7] R. Ludwig and R. H. Katz, "The Eifel algorithm: making TCP robust against spurious retransmissions," *ACM Comput. Commun. Review*, vol. 30, no. 1, pp. 30-36, Jan. 2000.

[8] T. Goff, J. Moronski, D. S. Phatak, and V. Gupta, "Freeze-TCP: a true end-to-end TCP enhancement mechanism for mobile environments," in *Proc. 9<sup>th</sup> Annu. Conf. IEEE Comput. Commun. Soc. (INFOCOM 2000)*, pp. 1537-1545, Tel Aviv, Israel, Mar. 2000.

[9] M. Stemm and R. H. Katz, "Vertical handoffs in wireless overlay networks," *ACM Mobile Networking Applicat.*, vol. 3, no. 4, pp. 335-350, 월 1998.

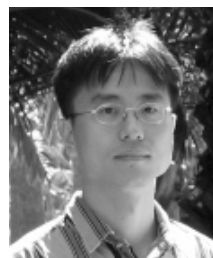
[10] Q. Zhang, C. Guo, Z. Guo, and W. Zhu, "Efficient mobility management for vertical handoff between WWAN and WLAN," *IEEE Commun. Mag.*, vol. 41, no. 11, pp. 102-108, Nov. 2003.

[11] M. Buddhikot, G. Chandranmenon, S. Han, Y.



- W. Lee, S. Miller, and L. Salgarelli, "Integration of 802.11 and third-generation wireless data networks," in *Proc. 22<sup>nd</sup> Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM 2003)*, vol. 1, pp. 503-512, San Francisco, U.S.A., Mar.-Apr. 2003.
- [12] J. McNair and F. Zhu, "Vertical handoffs in fourth-generation multinet network environments," *IEEE Wireless Commun.*, vol. 11, no. 3, pp. 8-15, June 2004.
- [13] Q. Song and A. Jamilpour, "A network selection mechanism for next generation networks," in *Proc. IEEE Int. Conf. Commun. (ICC 2005)*, vol. 2, pp. 1418-1422, Seoul, Korea, May 2005.
- [14] S. E. Kim and J. A. Copeland, "TCP for seamless vertical handoff in hybrid mobile data networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM '03)*, vol. 2, pp. 661-665, San Francisco, Dec. 2003.
- [15] A. Gurtov and J. Korhonen, "Effect of vertical handovers on performance of TCP-friendly rate control," *ACM Mobile Computing Commun. Review*, vol. 8, no. 3, pp. 73-87, July 2004.
- [16] R. Chakravorty, P. Vidales, K. Subramanian, I. Pratt, and J. Crowcroft, "Performance issues with vertical handovers - experiences from GPRS cellular and WLAN hot-spots integration," in *Proc. 2nd IEEE Annu. Conf. Pervasive Computing Commun. (PerCo, 2004)*, pp. 155-164, Orlando, U.S.A., Mar. 2004.
- [17] Y. Matsushita, T. Matsuda, and M. Yamamoto, "TCP congestion control with ACK-pacing for vertical handover," in *Proc. IEEE Wireless Commun. Networking Conf. (WCNC 2005)*, vol. 3, pp. 1497-1502, New Orleans, U.S.A., Mar. 2005.
- [18] S. E. Kim and J. A. Copeland, "Interworking between WLANs and 3G networks : TCP challenges," in *Proc. IEEE Wireless Commun. Networking Conf. (WCNC 2004)*, vol. 2, pp. 1252-1257, Atlanta, U.S.A., Mar. 2004.
- [19] L. J. Chen, G. Yang, T. Sun, M. Y. Sanadidi, and M. Gerla, "Enhancing QoS support for vertical handoffs using implicit/explicit handoff notifications," in *Proc. IEEE 2<sup>nd</sup> Int. Conf. Quality Service in Heterogeneous Wired/Wireless Networks*, pp. 8-37, Lake Vista, U.S.A., Aug. 2005.
- [20] K. Tsukamoto, Y. Fukuda, Y. Hori, and Y. Oie, "New TCP congestion control schemes for multimodal mobile hosts," *IEICE Trans. Commun.*, vol. E89-B, no. 6, pp. 1825-1836, June 2006.
- [21] H. Rutagemwa, S. Pack, X. Shen, and J. W. Mark, "Robust cross-layer design of wireless-profiled TCP mobile receiver for vertical handover," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3899-3911, Nov. 2007.
- [22] S. Kim and J. Jang, "Improved TCP Performance during the upward vertical handover using an IEEE 802.21 MIHF Framework," in *Proc. 5<sup>th</sup> Int. Conf. Convergence Hybrid Inform. Technol. (ICHIT '11)*, pp. 197-204, Daejeon, Korea, 2011.
- [23] J. Lee and S. Song, "TCP aware link layer agent for seamless vertical handoff in the cellular/WLAN integrated network," *IEEE Commun. Lett.*, vol. 14, no. 11, pp. 1017-1019, Nov. 2010.

석우진 (Woojin Seok)



1996년 경북대학교 컴퓨터공학과 학사

2003년 Univ. North Carolina, Computer Science 석사

2008년 충남대학교 컴퓨터공학과 박사

2003년~현재 한국과학기술정보

연구원 선임연구원

<관심분야> 무선/이동 QoS, TCP 성능 분석

이 민 선 (Minsun Lee)



1989년 숙명여자대학교 물리  
학과 학사

1997년 Univ. of Nebraska-  
Lincoln, Electrical Engineering  
석사

2012년~현재 한국과학기술정  
보연구원 기술원

<관심분야> 가상화 테스트베드, 병렬코드 등

이 만 희 (Manhee Lee)



1995년 경북대학교 컴퓨터공학  
과 공학사

1997년 경북대학교 공학석사

2008년 Texas A&M 대학교  
컴퓨터공학과 공학박사

<관심분야> 네트워크/시스템/스  
마트폰 보안, 고성능 시스템,

컴퓨터교육