

이질적 무선망 사이의 수직적 핸드오프에서의 TCP 성능 분석

TCP Performance Study in Vertical Handoff across Heterogeneous Wireless Networks

백상헌(白尙憲, Sangheon Pack), 최양희(崔陽熙, Yanghee Choi)

서울대학교 (Seoul National University) *

연구세부분야: 정보통신

주소 : (151-742) 서울시 관악구 신림동 산 56-1 서울대학교 301동 518호 멀티미디어 통신 연구실

전화 : 02-880-1832

FAX : 02-872-2045

E-Mail : 백상헌: shpack@mmlab.snu.ac.kr, 최양희: yhchoi@snu.ac.kr

이질적 무선망 사이의 수직적 핸드오프에서의 TCP 성능 분석

TCP Performance Study in Vertical Handoff across Heterogeneous Wireless Networks

요약

TCP는 웹브라우징, 이메일, 파일 전송 등에서 사용되는 가장 중요한 인터넷 프로토콜 중의 하나이다. 하지만 TCP는 링크 에러율이 극히 적은 유선망에 적합하도록 설계되었기 때문에 무선망에 적용되었을 경우 성능 저하 현상을 가져오게 된다. 이러한 문제의 해결을 위해서 많은 연구가 진행되었다. 하지만 기존 연구의 경우 단일 링크 특성을 가지는 무선망 (즉, 동질적 무선망)에서의 이동성만을 고려하여 TCP의 성능을 분석하였다. 하지만 차세대 무선/이동 통신망의 경우 여러 다양한 무선망이 혼합된 이질적 무선망이 될 것이다. 따라서 이러한 이질적 무선망에서의 TCP의 성능을 분석하는 것이 필요하다. 본 논문에서는 이러한 이질적 무선망에서의 단말기의 이동성에 따른 TCP 성능을 분석한다. 그 결과 기존의 연구 결과에서는 볼 수 없던 다른 형태의 TCP 성능 저하 현상을 경험할 수 있었는데 이는 이질적 무선망 사이의 수직적 핸드오프로 인한 패킷 손실과 링크 특성(대역폭과 링크 지연 시간)의 급격한 변화로 인해 TCP의 재전송 타이머 등이 새로운 링크 특성에 맞지 않게 설정되기 때문에 발생하는 것이다. 다양한 시뮬레이션 분석 결과 기존의 다양한 TCP 개선 기법으로는 이러한 문제를 완전히 해결할 수 없고 새로운 접근 기법이 필요하다는 것을 알 수 있었다.

* 키워드 : TCP, 성능 분석, 수직적 핸드오프, 이질적 무선망, 시뮬레이션

Abstract

TCP (Transmission Control Protocol) is one of the most important Internet protocols, which is widely used in wireless networks as well as wired networks. However, when TCP is deployed for wireless networks, it takes severe performance degradation because TCP was designed for wired network. To overcome this drawback, a number of TCP variants have been proposed in the literature. However, most previous schemes did not consider TCP enhancement over heterogeneous networks. In heterogeneous networks, an mobile node (MN) may move from one access network to another (i.e., vertical handover). In the case of vertical handover, an MN experiences a TCP performance degradation caused by the packet loss and the sudden change of link characteristics between two different access networks. In this work, we investigate the TCP performance degradation occurred in vertical handover across heterogeneous networks. First, we have conducted the measurement study over GPRS-WLAN testbed. In the measurement study, we observed the TCP performance degradation in the case of handover from WLAN to GPRS. In order to study more different TCP behaviors during vertical handover, we performed comprehensive simulations using a network simulator 2 (ns-2). Based on measurement and simulation results, we investigated how to improve TCP performance in vertical handover and we concluded that the existing mechanisms cannot be perfect solutions and new mechanisms are strongly required.

* Keyword : TCP, Performance analysis, Vertical handoff, Heterogeneous wireless networks, Simulation

1. 서론

TCP (Transmission Control Protocol)는 현재 가장 널리 쓰이는 인터넷 프로토콜 중의 하나로 웹 브라우징, 이메일, 파일 전송 등과 같은 여러 프로그램에서 사용되는 아주 중요한 프로토콜이다. 이러한 TCP는 기본적으로 유선망에서 사용되기 위해서 만들어졌기 때문에 무선망에서 사용될 경우 심각한 성능 하락을 가져오게 된다. 이러한 성능 하락의 기본적인 이유는 TCP는 패킷 손실의 이유를 항상 네트워크의 혼잡 (Congestion)이라고 판단하여 네트워크로 보내는 패킷의 전송률을 크게 낮추기 때문이다 [1]. 하지만 무선망의 경우 네트워크의 혼잡이 전혀 없는 경우에도 무선 링크의 높은 에러율과 가변적인 채널 상태 등으로 인해서 패킷 손실이 발생할 수 있다. 이런 경우 TCP의 전송측에서 전송율을 떨어뜨리게 되면 결국 전체 네트워크의 효율성 (Utilization)이 떨어지는 문제가 발생하게 된다.

이러한 문제를 해결하기 위해서 Indirect TCP [2], M-TCP [3], Snoop 프로토콜 [4] 등과 같은 많은 기법들이 제안되었다. 이러한 방법들은 접근 방법에 따라 종단간 기법, 분리형 기법, 지역적 재전송 기법 등으로 나뉘어볼 수 있다 [1]. 하지만 이러한 기법들은 하나의 무선 액세스망으로 구성된 균질한 무선망 (Homogeneous wireless networks)을 대상으로 하였기 때문에 서로 다른 링크 특성을 가지는 무선 액세스망이 혼합된 이질적 무선망 (Heterogeneous wireless networks)에서 발생하는 TCP의 성능 저하에 대해서는 고려하고 있지 않다. 최근의 몇몇 연구를 [5, 6] 통해 이질적 무선망에서의 TCP 성능이 분석되었지만 대부분 TCP 자체의 개선보다는 핸드오프 지연시간이 TCP 성능에 미치는 영향과 지연 시간을 줄이는 방법을 주로 연구하였다. 본 연구에서는 이러한 이질적인 무선 액세스망의 핸드오프 과정에서 발생하는 TCP 성능 저하 문제를 분석하고 TCP의 개선을 통한 성능 향상 기법을 고려하였다.

이질적인 무선 액세스망이란 서로 다른 특성을 가지는 무선 액세스망이 서로 연동되어 운영되는 무선망을 의미한다.

다. 유럽식 무선 데이터 서비스 표준인 GPRS (General Packet Radio Service) [7] 와 최근 무선 인터넷 수요가 많은 공공 장소에 설치되고 있는 무선랜 (WLAN: Wireless Local Area Network) [8] 을 연동하여 GPRS-WLAN 망을 구성하는 것이 이러한 이질적인 무선 액세스망의 좋은 예이다. 차세대 인터넷의 경우 비단 GPRS와 WLAN 뿐만이 아니라 블루투스, 3세대 이동통신망 등의 기술이 적용되면 보다 다양한 형태의 이질적인 무선 액세스망이 나타나게 될 것이다.

무선/이동 통신망의 경우 이동 단말기가 네트워크에 대한 자신의 접속 지점을 수시로 바꾸기 때문에 사용자들에게 지속적 서비스의 제공을 위해서는 핸드오프 (Handoff)라는 기술이 필요하다. 이질적 무선 액세스망에서의 핸드오프는 서로 다른 액세스망 사이를 이동하는 것이기 때문에 기존의 핸드오프 (즉, 수평적 핸드오프 (Horizontal Handoff))와 구별하여 수직적 핸드오프 (Vertical Handoff)라고 불린다. 수직적 핸드오프의 경우 서로 다른 액세스망 사이를 이동하기 때문에 AAA (Authentication, Authorizing, and Accounting) 과정 등으로 인해 보다 큰 핸드오프 지연 시간이 발생하게 된다. 앞서 소개한 무선망에서의 TCP 성능 개선을 위한 방법들은 모두 이러한 이질적 무선 액세스망에서의 링크 특성의 변화는 고려하지 않았다. 하지만, 수직적 핸드오프가 발생하는 경우에는 무선 링크 특성의 갑작스런 변화로 인해 새로운 형태의 TCP 성능 저하 현상을 겪게 된다. 본 논문에서는 이러한 이질적 무선 액세스망에서의 수직적 핸드오프로 인해 발생하는 TCP 성능 변화를 분석하고자 한다.

본 논문의 구성은 다음과 같다. 2절에서는 수직적 핸드오프로 인해 발생할 수 있는 TCP 성능 저하의 요인을 소개한다. 3절에서는 GPRS-WLAN 테스트베드에서 관찰한 TCP의 성능 변화를 기술하고 4절에서는 모의실험을 통해 보다 구체적인 형태의 TCP 성능 변화를 분석한다. 5절에서는 이러한 TCP 성능 저하 현상을 완화시킬 수 있는 접근 방법을 제시하고 6장에서 결론을 맺는다.

2. 수직적 핸드오프에서의 TCP 성능 저하 요인

앞에서 언급한 바와 같이 본 연구에서는 서로 다른 링크 특성을 가지는 무선 액세스망 사이를 무선 단말기가 이동하는 경우에 발생하는 TCP의 성능 저하 문제를 다루고 있다. 이러한 성능 저하의 원인은 크게 2가지로 분석해볼 수 있다.

2.1 핸드오프 과정에서의 패킷 손실

한 무선 단말기가 한 액세스 라우터에서 다른 액세스 라우터로 이동하는 경우 새로운 액세스 라우터의 발견과 위치 등록, 인증 과정 등을 거쳐야 하기 때문에 핸드오프 지연 시간이 발생하게 된다. 따라서 이러한 지연 시간 내에 도착한 패킷들은 손실이 된다. 이러한 손실을 최소화하기 위해서 버퍼링, 다중 연결 등의 기법들이 제안되었지만 실제 네트워크에서 실험 결과 이러한 방법들을 사용하는 경우에도 일부 패킷의 손실은 피할 수 없다는 사실을 알 수 있었다. 더군다나 서로 다른 네트워크 사이의 핸드오프 과정에서는 사용자 인증, 등록 등을 위해서 더 많은 시간이 소비되기 때문에 더 많은 패킷 손실이 발생하게 된다. TCP 성능의 관점에서 볼 때 패킷 손실이 발생한 경우 TCP의 전송측은 네트워크에서 혼잡이 발생한 것으로 간주하여 전송율을 크게 떨어뜨리게 된다. 하지만 핸드오프에 의한 패킷 손실은 네트워크의 혼잡에 의해서 발생한 것이 아니기 때문에 이는 결국 네트워크의 효율성을 떨어뜨리는 결과를 가져오게 된다.

2.2 갑작스런 RTT의 변화

앞에서 설명한 핸드오프로 인한 패킷 손실은 널리 알려진 문제이다. 하지만 본 연구에서 초점을 맞춘 부분은 이질적

인 네트워크 사이의 핸드오프에서 발생하는 RTT(Round Trip Time)의 갑작스런 변화에 의한 TCP의 성능 저하 문제이다. TCP는 신뢰성있는 전송을 위해서 데이터를 보내고 그것에 대한 ACK을 받는 과정을 반복하면서 데이터를 전송한다. 이 때 데이터가 제대로 전송되었는지 여부는 ACK이 오느냐 오지 않느냐로 판단할 수 있는데 전송측에서는 하나의 재전송 타이머 (Retransmission timer)를 통해 ACK이 올 때까지 기다리게 된다. 만약 타이머가 다 경과할 때까지 ACK이 오지 않았다면 전송측은 그 패킷이 손실되었다고 가정을 하고 그 패킷을 재전송하고 전송율을 제어하는 cwnd와 ssthresh의 값을 각각 1과 cwnd/2로 줄이게 된다. 이러한 재전송 타이머의 값은 현재 네트워크의 상태를 반영하기 위해서 데이터 전송 과정에서 얻게 된 RTT 값을 이용하여 타이머의 값을 변화시키게 된다 [9]. 바로 이러한 TCP의 동작 방식으로 인해서 서로 다른 특성의 엑세스망을 이동하는 경우에 문제가 발생한다. 예를 들어, 대역폭이 아주 높고 지연 시간이 아주 짧은 네트워크(A) 에서 대역폭이 낮고 지연 시간이 긴 네트워크(B)로 이동하는 경우를 생각해 보자. 이 경우 네트워크(A)에서 처음에 TCP의 재전송 타이머는 짧은 RTT 값으로 인해 아주 짧은 값으로 설정되어 있을 것이다. 하지만 네트워크(B)로 무선 단말기가 이동한 경우에는 패킷 손실이 없다고 하더라도 네트워크의 지연 시간이 아주 길기 때문에 네트워크(A)에서 설정된 재전송 타이머가 경과한 후에야 ACK이 도착하는 문제가 발생하게 된다. 이런 경우에는 전송측은 불필요한 재전송을 하게 되고 cwnd, ssthresh 등의 값을 크게 줄이기 때문에 전체적인 TCP 성능과 네트워크의 효율성이 떨어지는 문제가 발생한다. 이렇게 네트워크의 큰 지연 시간으로 인해 타이머 종료가 성급하게 발생하는 문제를 의심스러운 타임아웃 (Spurious timeout)이라고 한다. 이러한 의심스러운 타임아웃의 문제를 해결하기 위해서 TCP eifel 이란 방법이 제안되었다 [10]. 이 방법은 첫 번째 재전송의 시점을 기억한 뒤에 ACK이 도착하게 되면 기억한 시점과 비교를 해서 앞서 발생한 타임아웃이 의심스러운 타임아웃인지 여부를 결정하게 된다. 만약 의심스러운 타임아웃이라면 타임아웃 이전의 cwnd와 ssthresh 값을 복원해서 뒤에 이어지는 패킷을 전송하게 되고 의심스러운 타임아웃이 아니라면 그대로 전송을 계속하게 된다. 이러한 TCP eifel 기법이 의심스러운 타임아웃의 문제를 해결해 줄 수는 있지만 본 연구에서 대상으로 하는 이질적인 네트워크의 핸드오프 과정에서는 핸드오프로 인한 패킷 손실이 발생하기 때문에 TCP eifel 만으로는 효과적으로 TCP의 성능 저하를 막을 수 없다. 이 문제는 4.3절에서 시뮬레이션 결과를 통해 분석할 것이다.

3. GPRS-WLAN 테스트베드에서의 TCP 성능 측정

3.1 시스템 아키텍처

이질적인 무선 엑세스망에서의 핸드오프 과정에서 발생하는 TCP 성능 저하 현상을 파악하기 위해서 먼저 Fraunhofer FOKUS의 MOBIS 팀에서 구축한 GRPS-WLAN 테스트베드 [11]를 통해 그 현상을 관찰해 보았다.

그림 1은 MOBIS 팀에서 구축한 GRPS-WLAN의 전체적인 아키텍처를 보여준다. GPRS는 유럽의 이동 통신 표준인 GSM에서 무선 데이터 서비스를 지원하기 위해서 확장된 패킷 서비스 전용 무선망이다 [7]. 반면 WLAN은 현재 국내에서도 활발히 서비스가 시작되고 있는 IEEE 802.11b에 기반한 무선랜 시스템이다. 이러한 GPRS-WLAN 테스트베드 상에서 무선 단말기의 이동성을 지원하기 위해서 Mobile IPv4 프로토콜 스택을 설치하였다. Mobile IPv4 프로토콜은 헬싱키 대학에서 배포한 프로토콜 스택을 이용하였다 [12]. 최근에 Mobile IPv6에 대한 표준화 작업이 완료되었지만 현재의 GPRS, UMTS 망의 경우 Mobile IPv4만을 기본적으로 지원하기 때문에 [6] 본 연구에서도 Mobile IPv4를 가정하였다. 향후 연구에서는 Mobile IPv6와 그와 관련된 여러 확장 기법을 통해 이동 통신망과 WLAN을 연동할 수 있을 것이다.

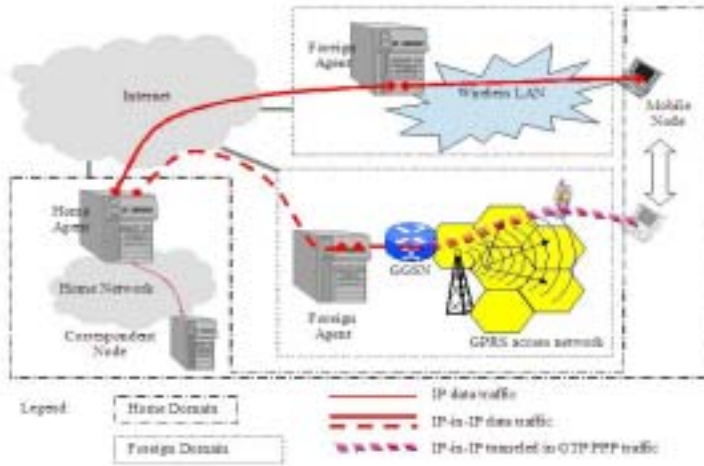


그림 1) GPRS-WLAN 테스트베드 아키텍처

3.2 테스트베드에서의 TCP 성능 분석

아래의 그림 2는 GPRS-WLAN 테스트베드에서의 TCP를 사용하였을 때의 데이터 전송 결과를 보여준다. 여기서는 한 호스트 (CN:Correspondent node)가 이동 단말기 (MN: Mobile node)로 TCP를 사용하여 데이터를 전송하고 MN가 WLAN에서 GPRS로 핸드오프를 하는 경우의 TCP 패킷의 시퀀스를 보여준다.

그림 2의 결과를 보면 핸드오프가 발생한 경우 다수의 타임아웃이 발생하였음을 알 수 있다. 타임아웃이 발생한 시점은 송신측이 재전송을 수행하는 시점과 일치한다. 이렇게 다수의 타임아웃이 발생하는 이유는 WLAN의 경우 네트워크

의 지연시간이 아주 짧아서 TCP의 재전송 타이머가 아주 짧게 설정이 되기 때문이다. 이러한 짧은 재전송 타이머를 네트워크의 지연 시간이 긴 GPRS 네트워크에서 그대로 사용하게 되면 송신측은 성급하게 타임아웃을 발생시키고 해당 패킷을 재전송하게 된다. 그림 2를 보면 송신측이 재전송을 하고 난 뒤에 다수의 중복된 ACK이 도착하는 것을 알 수 있다. 중복된 ACK은 순서가 틀리게 패킷이 도착했거나 중간에 일부 패킷이 손실이 있는 경우에 발생한다. 따라서 여기서는 핸드오프 과정에서 일부 패킷이 손실되었다는 것을 알 수 있다. 여기서 관찰할 수 있는 TCP 성능 저하 현상은 바로 재전송 이후에 cwnd의 크기를 1로 줄여서 slow start (SS)를 시작함으로써¹⁾ 전체 네트워크의 효율성이 떨어진다는 사실이다. 하지만 이 경우의 패킷 손실은 네트워크의 혼잡으로 인해서 발생한 것이 아니고 핸드오프 과정에서 새로운 경로를 설정하고 등록하는 데 걸린 지연시간으로 인해 발생한 것이므로 TCP 전송측이 전송률을 떨어트리는 것은 효율성의 측면에서 효과적이지 못하다. 따라서 이를 위한 TCP의 개선이 필요하다.

4. 시뮬레이션을 통한 TCP 성능 분석

GPRS-WLAN 테스트베드에서의 TCP 성능 분석 후 보다 구체적인 성능 분석을 위해서 네트워크 시뮬레이터인 NS-2 [13]를 이용하여 TCP의 성능을 분석해 보았다. NS-2는 패킷 네트워크 상에서의 시뮬레이션을 위해서 만들어진 시뮬레이션 툴로서 그 정확성과 다양한 프로토콜 지원 특성으로 인해 가장 널리 사용되는 시뮬레이션 툴 중의 하나이다.

4.1 시뮬레이션 방법론

본 연구에서는 앞서 말한바와 같이 NS-2를 이용하여 무선 링크의 특성 변화가 TCP의 성능에 미치는 영향을 분석하고자 하였다. 이를 위해 우선 NS-2에서 지원하는 TCP 모듈을 일부 수정하였다. 즉, 기존의 TCP 모듈의 경우 연결 중간에 하위 링크의 특성이 바뀌는 것을 지원하지 않기 때문에 이에 대한 수정이 필요하다. 이러한 수정을 위해 우리는 문맥 전달(context transfer)의 개념을 이용하였다. 즉, 하나의 TCP 연결을 생성시켜 A 형식의 링크 상에서 동작시킨 다음, 핸드오프가 발생하는 시점을 고려해서 그 TCP 연결을 중단시킨다. 그리고는 중단된 TCP 연결의 다양한 상태 정보(현재의 시퀀스 번호, 마지막 ack 번호, cwnd의 크기, 타이머, ssthresh의 크기 등등)의 값을 얻어낸다. 그런 다음 얻어낸 정보를 B 형식의 링크 상에서 동작하게 될 또 다른 TCP 연결의 초기값으로 설정하게 된다. 그런 다음 B 형식의 링크 상에서의 문맥이 전달된 TCP를 실행하게 되면 기존의 TCP가 핸드오프 과정을 거친 후의 TCP로 동작하는 것과 같은 동일한 효과를 얻을 수 있다. 이러한 과정을 지원하기 위해서 NS-2의 기존 TCP 모듈에 몇 개의 상태 정보를 나타내는 변수를 추가로 선언하였다.

시뮬레이션에 사용한 기본적인 링크 특성은 다음과 같이 설정되었다.

- 링크 A의 특성: 40Kbps의 대역폭과 800ms의 지연시간
- 링크 B의 특성: 4Mbps의 대역폭과 80ms의 지연시간

위의 링크 특성은 링크 A의 경우 GPRS의 링크 특성을, 링크 B의 경우 IEEE 802.11b의 링크 특성을 반영하기 위한 값들이다. IEEE 802.11b는 이론상으로 11Mbps까지의 대역폭을 지원할 수 있지만 여러 단말기가 존재하는 경우 충돌로 인한 대역폭 감소와 무선 링크의 가변적인 특성으로 인해 실제로는 약 3-4Mbps의 대역폭을 얻을 수 있다. GPRS 링크의

경우 더 가변적인 링크 지연 시간을 가지게 되는데 참고 문헌에 의하면 수백 ms의 지연 시간을 가지는 것으로 알려져 있다 [14]. 그리고 awnd는 20, 패킷 사이즈는 1000 바이트로 설정되었다.

4.2 시뮬레이션 결과 I: A 링크에서 B 링크로의 핸드오프

우선 낮은 대역폭의 무선 액세스망에서 높은 대역폭의 무선 액세스망으로 핸드오프를 하는 경우의 TCP 동작을 살펴 보았다. 그림 3은 시뮬레이션에 의한 TCP의 동작 과정을 보여준다. 본 시뮬레이션을 위한 시나리오는 아래와 같다.

표 1 링크 A에서 링크 B로의 핸드오프 시나리오

시간	동작
0.0	tcp0 시작
14.63	이동 단말기가 핸드오프를 시작
14.67	이동 단말기의 핸드오프가 이전 액세스 라우터에 통보되고 버퍼링이 시작
14.71	tcp0에서 tcp1으로의 정보 전달
14.71	tcp1 시작
17.50	종료

4.2.1 핸드오프로 인한 패킷 손실과 포워딩 과정

시뮬레이션 시나리오에 따라 이동 단말기가 구간 C의 시간 14.63에 핸드오프를 시작하게 된다. 따라서 시간 14.63 이후에 전송된 48-61번 패킷은 모두 손실이 된다.

핸드오프가 완료된 시간 14.71에서 이전 액세스 라우터 (old access router: oAR)는 이동 단말기로 버퍼링된 64번 패킷을 보내게 된다. 여기서 64번 패킷만이 버퍼링되어서 전달되는 이유는 이동 단말기가 핸드 오프 이전에 보낸 45-47번에 대한 ACK의 경우 링크 A의 높은 지연 시간으로 인해 시간 14.71 이후에 도착하기 때문에 핸드오프 진행되는 시점에서 송신측이 가지고 있는 가장 높은 ACK 번호는 여전히 44번이기 때문이다. 따라서 송신측의 유효한 윈도우 크기는 45번에서 64번까지 20개(= $\min\{cwnd, awnd\}$)가 되는 것이다. 결국, 14.71 이후에 도착하는 45-47번에 대한 ACK은 누적된 ACK (cumulative ACK) 기법에 의해 무시된다.

4.2.2 중복된 ACK의 수신과 재전송

송신측은 시간 14.87216에 14.71에서 전송한 패킷 64번에 대한 ACK을 받게 되는데, 이 경우 48-61까지의 패킷 손실이 있었기 때문에 64번이 아닌 47번에 대한 ACK을 받게 된다. 그러면 송신측의 입장에서는 47번에 대한 ACK은 새로운 ACK이므로 이에 대해서 다시 65, 66, 67번 패킷을 보내게 되고 이를 받은 수신측은 다시 중복된 ACK인 47번에 대한 ACK을 보내게 된다. 결국 65, 66, 67번 패킷에 대해서 계속해서 중복된 ACK이 오게 되므로 시간 15.03848에서 세 번째 중복된 ACK을 받게 된다. 그러면 송신측은 세 번째 중복된 ACK에 대해서 Fast retransmission을 수행하게

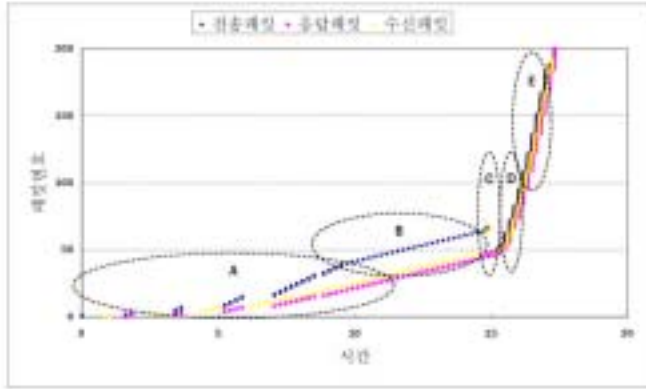


그림 2) 링크 A에서 링크 B로의 핸드오프 시뮬레이션 결과

되므로 시간 15.03848에 48번 패킷을 재전송하고 TCP Tahoe 알고리즘에 의해 SS 상태로 들어가서 cwnd는 1이 되고 ssthresh는 $10(=20/2)$ 이 된다.

4.3 시뮬레이션 결과 II: B 링크에서 A 링크로의 핸드오프 - 패킷 무손실의 경우

다음으로는 링크 B에서 링크 A로의 핸드오프가 발생할 때의 TCP의 성능을 분석하였다. 우선 핸드오프로 인한 패킷 손실이 없는 경우를 살펴보자. 이를 위한 시뮬레이션 시나리오는 표 2와 같다. 우선 시간 0.0에서 tcp0이 시작하고 시간 2.07에서 TCP의 수신측 (즉, 이동 단말기)이 핸드오프를 시작한다. 그러면 이동 단말기가 이동한 사실은 시간 2.10에 송신측에 감지되어 송신측은 전송을 중단한다. 핸드오프를 시작한 무선 단말기는 시간 2.15에 핸드오프를 완료하고 tcp0에서의 TCP 관련 정보를 얻은 다음에 새로운 tcp1을 시작하게 된다. 즉, 이 경우에서의 핸드오프 지연 시간은 2.15-

2.07=0.08s=80ms가 된다. 이와 같은 시나리오에서 발생한 TCP 동작은 그림 4에 나와있다. 그림 4에서 각 부분의 TCP 동작을 살펴보자.

시간	동작
0.0	tcp0 시작
2.07	이동 단말기이 핸드오프를 시작
2.10	이동 단말기의 핸드오프가 이전 액세스 라우터에 통보되고 버퍼링이 시작
2.15	tcp0에서 tcp1으로의 정보 전달
2.15	tcp1 시작
39.0	종료

2 A B ()

4.3.1 패킷 버퍼링을 통한 무손실 핸드오프

이전 시뮬레이션과 마찬가지로 A구간과 B구간에서는 각각 SS와 혼잡 방지 (Congestion Avoidance: CA)에 의해서 패킷을 전송하게 된다. 구간 C의 시간 2.07에 이동 단말기는 핸드오프를 시작한다. 하지만 송신측이 수신측으로 보냈던 모든 패킷은 시간 2.07 이전에 도착하기 때문에 핸드오프로 인한 패킷 손실은 발생하지 않는다. 패킷 손실을 막기 위해서 이동 단말기는 시간 2.10에 핸드오프를 시작했다는 사실을 이전의 액세스 라우터에 알리게 된다. 그러면 이전 액세스 라우터는 그 이후에 도착하는 패킷은 전송하지 않고 버퍼링을 시작하게 된다.

4.3.2 타임아웃의 발생과 ACK 모호성 문제

시간 3.15에서 이동 단말기는 새로운 액세스 라우터 (new access router: nAR)에 연결이 되고 HA와 CN에 대한 바인딩 업데이트를 완료하게 된다. 그런 다음에 이전 액세스 라우터에 버퍼링되었던 20개의 패킷(192번부터 211번)이 이동 단말기로 일괄적으로 전송된다. 이 상황에서 송신측은 여전히 192번 패킷에 대한 ACK을 기다리고 있다. 이 때 192번에 대한 ACK을 기다리고 있는 재전송 타이머는 링크 지연 시간이 아주 짧은 링크 B에서 측정된 RTT에 기반하여 설정이 된 것이기 때문에 상당히 짧게 설정되어 있다. 따라서 송신측은 충분한 시간 동안 ACK을 기다리지 않고 성급하게 시간 3.15에서 타이아웃을 발생시키고 패킷을 재전송하게 된다. (RFC 2988 [15]에 의하면 아주 짧은 재전송 타이머가 설정되는 것을 방지하기 위해서 재전송 타이머의 최소값을 1.0으로 규정하고 있고 NS-2도 이 값을 따르고 있다.)

시간 3.966에 송신측은 패킷 192에 대한 ACK을 받게 된다. 하지만 송신측은 이 ACK이 처음에 보낸 192번에 대한 ACK인지 재전송한 192번에 대한 ACK인지를 구별할 수 없다. 즉, 이 경우에 ACK 모호성 (ambiguity) 문제가 발생한다. 결국 송신측은 192번 ACK을 재전송에 의한 ACK으로 간주하여 cwnd를 1에서 2로 증가시켜 193, 194 패킷을 보내면서 SS

를 시작한다.

4.3.3 의심스러운 타임아웃 (spurious timeout) 현상

D 구간의 시간 8.126에 송신측은 패킷 211에 대한 중복된 ACK을 받게 된다. 이는 시간 3.15에서 송신측이 성급하게 패킷 192를 재전송하였기 때문에 발생한 중복된 ACK이다. 이러한 211에 대한 중복된 ACK은 시간 8.334, 8.542에도 계속 수신된다. 하지만 이 경우 3번째 중복 ACK에 대해서 Fast retransmission은 적용되지 않는다. 왜냐하면 이미 3.15에서 타임아웃이 발생했었고 타임아웃이 발생한 시점에서 유효한 데이터 (outstanding data, 패킷을 보냈고 아직 ACK을 받지 못한 데이터)는 192번부터 211번까지이기 때문이다. 즉, TCP에서는 중복된 ACK에 의해서 여러 번 패킷이 재전송되는 현상을 막기 위해서 재전송되는 시점에서의 유효 데이터에 대해서 모두 ACK을 수신할 때까지는 추가적인 재전송을 금지하고 있다. 이와 같은 중복된 ACK은 시간 12.078까지 계속되는데 이는 실제 핸드오프 과정에서 패킷 손실이 없었는데도 불구하고 시간 3.15에서 타임아웃이 걸린 뒤에 이미 제대로 전송된 데이터를 다시 재전송을 했기 때문에 발생한 것이다. 즉, 이는 의심스러운 타임아웃에 의한 현상이다.

4.3.4 RTT의 갱신과 혼잡 방지 과정

시간 12.286에는 패킷 212에 대한 새로운 ACK이 도착하고 이로 인해 cwnd는 2로 증가한 후 SS에 의해서 전송을 계속하게 된다. 또한 새로운 ACK을 수신하였으므로 송신측은 현재 링크의 RTT를 갱신하게 되고 이후부터는 링크 A에 대한 RTT값에 기반한 새로운 재전송 타이머가 사용된다. 시간 13.742에서는 219번 패킷에 대한 새로운 ACK이 도착하고 cwnd는 ssthresh와 같은 5가 되면서 이후부터 송신측은 CA 과정을 수행하게 된다.

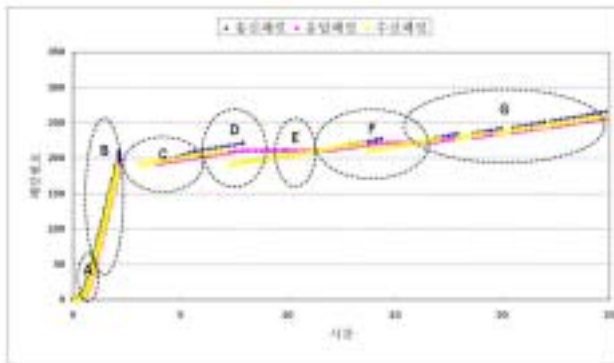


그림 3) 링크 A에서 링크 B로의 핸드오프 시뮬레이션 결과 (패킷 무손실)

4.4 시뮬레이션 결과 III: B 링크에서 A 링크로의 핸드오프 -패킷 손실의 경우

세 번째 시뮬레이션은 두 번째 시뮬레이션과 거의 유사한 시나리오에서 이루어졌다. 다만 무선 단말기가 핸드오프를 시작하는 시점이 시간 2.06이기 때문에 이 경우 이동 단말기는 189-191, 3개의 패킷을 받지 못하게 된다. 이외의 시나리오는 패킷 손실이 없는 경우와 동일하다.

표 3 링크 B에서 링크 A로의 핸드오프 시나리오 (패킷 손실)

시간	동작
0.0	tcp0 시작
2.06	이동 단말기가 핸드오프를 시작
2.10	이동 단말기의 핸드오프가 이전 액세스 라우터에 통보되고 버퍼링이 시작
2.14	tcp0에서 tcp1으로의 정보 전달
2.14	tcp1 시작
39.0	종료

4.4.1 패킷 손실의 발생

세 번째 시뮬레이션에 이동 단말기는 시간 2.06에 핸드오프를 시작한다. 시간 2.06까지는 172번에서 188번까지의 패킷만이 수신측에 도착했기 때문에 결국 189-191, 3개의 패킷은 손실되는 결과를 초래한다. 시간 2.10에 이전 액세스 라우터는 수신측이 핸드오프한 사실을 알게 되고 그러면 이전 액세스 라우터는 수신측이 위치 등록 과정을 마칠 때까지 수신측으로 패킷을 보내는 것을 중단하고 그 이후에 오는 패킷(192-208)을 모두 버퍼링하게 된다. 여기서 총 17개의 패킷만이 전송되는 이유는 앞선 과정에서 3개의 패킷이 손실되어 송신측에 도달한 ACK 패킷의 개수가 총 17개이기 때문이다.

4.4.2 타임아웃의 발생

시간 2.14에 수신측은 새로운 액세스 라우터로 핸드오프를 완료한다. 그러면 이전의 액세스 라우터는 버퍼링된 17개의 패킷을 수신측으로 전송한다. 이 때 송신측은 189번에 대한 ACK을 기다리고 있다. 하지만 앞에서 말한대로 189번부터 191번까지의 패킷이 손실되었기 때문에 189번에 대한 ACK은 도착하지 않는다. 뿐만 아니라 링크 B 상에서 설정된 재전송 타이머가 사용되기 때문에 재전송 타이머는 아주 짧은 값을 유지하게 된다. 따라서 시간 3.14에 첫 번째 타임아웃이 발생하고 구간 C의 시간 3.956에 패킷 188에 대한 ACK이 도착한다. 이는 버퍼링 후 수신측으로 전송된 192번에 대한 ACK인데 189-191번 패킷이 손실되어 수신되지 못했기 때문에 수신측은 계속해서 188번에 대한 ACK을 보내는 것이다. 이러한 모든 ACK들은 새로운 ACK이 아니기 때문에 무시되고 시간 5.14(=3.14*1.0*2)에 두 번째 타임아웃이 발생한다.

두번째 타임아웃 후에는 $cwnd$ 는 1로 $ssthresh$ 는 기존 $cwnd$ 의 값의 1/2인 0.5가 되어야 하지만 NS-2의 TCP 구현에서 사용되는 최소 $ssthresh$ 값이 2이기 때문에 2로 설정된다. 두 번째 타임아웃 이후에도 계속해서 188번에 대한 중복된 ACK이 도착하다가 시간 7.492에서 189번에 대한 새로운 ACK이 도착하게 된다. 그러면 $cwnd$ 는 SS 과정에 의해서 2가 되고 190과 191, 두 개의 패킷을 전송하게 된다. 그러면 $ssthresh$ 값이 앞에서 2로 설정되었기 때문에 TCP 송신측은 그 이후로는 CA를 수행하게 된다.

4.4.3 CA과정과 추가의 중복된 ACK 수신

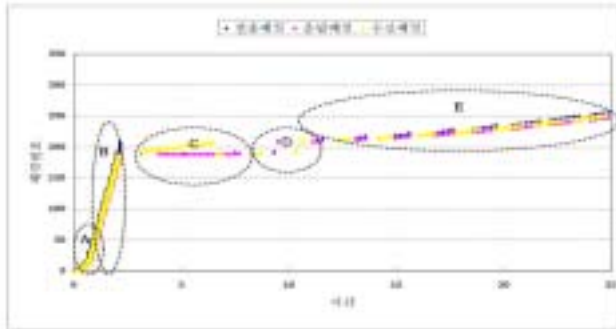


그림 4) 링크 B에서 A로의 핸드오프 (패킷 손실)

시간 9.308에는 190번에 대한 새로운 ACK을 받게되므로 $cwnd$ 는 CA에 의해 $2.5(=2+1/2)$ 이 되고 192번 패킷을 전송한다. 여기서 $cwnd$ 가 2.5이기 때문에 2개의 패킷을 보낼 수 있지만 현재 sliding window의 범위에 있는 것이 191, 192번이므로 192번만 보내게 된다. 이는 191번에 대한 ACK은 아직 받지 못했기 때문이다. 시간 9.516에는 208번에 대한 ACK이 도착하는데 그러면 송신측은 208번까지의 모든 데이터는 올바르게 수신되었다고 가정하고 209, 210번 패킷을 보내게 된다. 구간 E의 시간 11.124에는 또 다른 중복 ACK인 208번에 대한 ACK이 도착한다. 이는 시간 9.308에서 192번을 재전송했기 때문에 발생하는 것인데 실제 192번은 손실되지 않고 제대로 수신이 되었기 때문에 수신측은 192번 패킷을 받은 뒤에 중복 ACK을 발생시키게 되는 것이다. 즉, 재전송된 191번까지의 패킷을 수신한 뒤에 수신측은 189번에서 208번까지의 모든 패킷을 제대로 수신하였는데 그 뒤에 다시 192번이 도착했기 때문에 중복된 ACK을 발생시킨다. 하지만 송신측은 첫 번째 중복 ACK이므로 이를 무시하게 되고 시간 11.332에 209번에 대한 새로운 ACK을 받게 되면 211, 212번 패킷을 보내면서 CA에 의해 전송을 계속하게 된다.

5. TCP 성능 개선을 위한 방안

본 논문에서는 이질적인 무선 액세스망에서 수직적 핸드오프가 발생할 때의 TCP 성능 저하 현상을 파악하였다. 이러한 성능 저하의 주된 원인은 4장에서 제시한 시뮬레이션 결과에서도 알 수 있듯이 핸드오프 과정에서 발생하는 패킷 손실과 링크 특성의 변화로 인해 적합하지 않은 재전송 타이머의 값이 사용되기 때문이다.

4.2장과 4.4장의 시뮬레이션 결과에서 볼 수 있듯이 핸드오프 과정에서의 패킷 손실은 TCP 성능 저하를 초래하게 된다. 이러한 패킷 손실을 최소화하기 위해서는 현재 IETF에서 논의되고 있는 핸드오프 기법을 활용할 수 있다. 즉, 지역 에이전트 (Local Agent)를 통해 홈 에이전트 (Home Agent)로의 등록 지연 시간을 줄이는 Hierarchical Mobile IP, Regional registration 등과 같은 지역적 이동 관리 기법이나 링크 계층에서의 핸드오프에 대한 힌트를 IP 계층에서의 핸드오프 처리 과정에 활용하는 Fast Handover, Low latency handoff 등의 기법이 그 대표적인 예이다. 뿐만 아니라 이러한 기법들은 엑세스 라우터 간의 터널링이나 동시 바인딩 (Simultaneous binding) 등의 기법을 활용하여 패킷 손실을 줄이게 된다.

4.3장에서 살펴보았듯이 핸드오프 과정에서의 패킷 손실뿐만 아니라 부적절한 재전송 타이머 값 역시 TCP 성능 저하의 중요한 한 요인이다. 이러한 재전송 타이머로 인한 성능 저하를 줄이기 위해서는 발생한 타임아웃이 의심스러운 타임아웃인지를 우선 구별할 필요가 있다. 이를 위해서 TCP Eifel [10, 16], Freeze [17] 등의 기법을 이용할 수 있다. 이 방법의 경우 발생한 타임아웃이 의심스러운 타임아웃인 경우 TCP의 상태값을 이전의 값으로 복구함으로써 TCP 성능 저하를 막게 된다. 하지만 링크 지연시간이 짧은 곳에서 큰 곳으로 핸드오프를 한 경우 이러한 의심스러운 타임아웃을 판별하기 이전에 다수의 재전송이 발생할 수도 있기 때문에 TCP Eifel만으로도 TCP 성능 저하를 완벽하게 막을 수는 없다. 따라서 수직적 핸드오프 과정에서 링크 특성의 변화를 빠르게 감지할 수 있는 기법이 필요하다. 뿐만 아니라 TCP Eifel은 핸드오프에 의한 패킷 손실은 고려하지 않았기 때문에 수직적 핸드오프에서의 TCP 성능 저하를 막기 위해서는 앞에서 말한 개선된 핸드 오프 기법과 의심스러운 타임아웃을 모두 고려하는 기법이 필요하다.

이와 같은 수직적 핸드오프에서의 TCP 성능을 개선하기 위해서는 [18]에서 논의된 바와 같이 Cross layer의 개념을 적용할 수 있을 것이다. 즉, 링크 계층의 정보를 이용하여 핸드오프를 예측하여 핸드오프로 인한 패킷 손실을 줄이고 링크 계층의 지연 시간을 고려해서 새로운 링크에 맞는 RTT를 설정하는 등의 기법을 적용할 수 있을 것이다. 전자에 해당하는 기법으로는 현재 IETF에서 논의되고 있는 low latency handoff, fast handover 등이 있고 후자에 해당하는 기법으로는 임시 재전송 타이머(Temporary retransmission timer)의 개념을 생각해볼 수 있다. 즉, 임시 재전송 타이머의 경우 WLAN에서 GPRS로 핸드오프하였을 경우 짧은 재전송 타이머로 인해 여러번 재전송이 발생하는 것을 막기 위해서 이동 단말기가 GPRS로 이동한 후 링크 계층의 연결을 설정하는 과정에서 파악한 링크 지연 시간에 기반하여 임시로 사용될 새로운 재전송 타이머 값을 추정하여 설정하는 것이다. 추후 연구에서 이러한 임시 재전송 타이머의 성능을 TCP Eifel 등의 기법과 비교할 것이다.

6. 결론

본 논문에서는 이질적인 무선 액세스망에서 무선 단말기가 링크 특성이 서로 다른 네트워크 사이를 이동하는 경우(즉, 수직적 핸드오프)에 발생하는 TCP 성능 저하 문제에 대해서 살펴보았다. 기존 연구 [5]에서 시뮬레이션을 통해 이러한 영향을 분석해 보았지만 [5]에서는 링크 지연 시간이 큰 곳에서 작은 곳으로 핸드오프하는 경우만을 살펴보았기 때문에 재전송 타이머의 부적절한 값에 의해 다수의 재전송이 일어나는 경우는 살펴보지 않았다. 이에 비해 본 연구에서는 다양한 핸드오프 과정에서 발생하는 TCP 성능 저하 현상을 관찰하였다.

이러한 성능 저하 현상을 막기 위해서 몇몇 방법들이 제안되었지만 수직적 핸드오프의 특성, 특히, 링크 특성 변화로 인한 부적절한 재전송 타이머의 사용 문제를 해결할 수 있는 방법은 아직 제안된 바가 없다. 따라서 향후 연구에서는 IETF에서 제안된 개선된 핸드 오프 방안과 임시 재전송 타이머의 개념을 결합하여 수직적 핸드오프에서의 TCP 성능 저하를 막을 수 있는 방법을 제안할 것이다.

참고문헌

- [1] H. Balakrishnan, V. Padmanabhan, S. Seshan, R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *IEEE/ACM Trans. on Networking*, Vol. 5, No. 6, Dec. 1997.
- [2] A. V. Bakre, B. R. Badrinath, "Implementation and Performance Evaluation of Indirect TCP," *IEEE Trans. on Computers*, Vol. 46, No. 3, March 1997.
- [3] K. Brown and S. Singh, "M-TCP: TCP for mobile cellular networks," *ACM Computer Communication Review*, Vol. 27, No. 5, Oct. 1997.
- [4] H. Balakrishnan, S. Seshan, R. H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks," *ACM Wireless Network*, Vol. 1, No. 4, Dec. 1995.
- [5] W. Hansmann, M. Frank, and M. Wolf, "Performance Analysis of TCP Handover in a Wireless/Mobile Multi-Radio Environment," in *Proc. of IEEE LCN 2002*, Nov. 2002.
- [6] R. Chakravorty, P. Vidades, K. Subramanian, I. Pratt, and J. Crowcroft, "Practical Experiences with Wireless Integration using Mobile IPv6," *ACM Mobile Computing and Communication Review*, 7(4), October 2003.
- [7] B. Ghribi and L. Logrippo, "Understanding GPRS: the GSM packet radio service," *Computer Network*, Vol. 34, No. 5, Nov. 2000.
- [8] B.P. Crow, I. Widjaja, J.G. Kim, and P.T. Sakai, "IEEE 802.11 Wireless Local Area Networks," *IEEE Communication Magazine*, Sep. 1997.
- [9] W.R. Stevens, "TCP/IP Illustrated," Vol. 1, Addison-Wesley, 1996.
- [10] R. Ludwig and R. H. Katz, "The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions," *ACM Computer Communication Review*, 30(1), January 2000.
- [11] Fraunhofer FOKUS: <http://www.fokus.gmd.de/fokus/>
- [12] HUT Mobile IP: <http://www.cs.hut.fi/Research/Dynamics/>
- [13] UCB/LBNL/MINT Network Simulator – ns, <http://www.isi.edu/nsnam/ns/>
- [14] R. Chakravorty, J. Cartwright, and I. Pratt, "Practical Experience with TCP over GPRS," in *Proc. of*

Globecom 2002, Dec. 2002.

- [15] V. Paxson and M. Allman, "Computing TCP's Retransmission Timer," IETF RFC 1988, Nov. 2000.
- [16] A. Gurtov and R. Ludwig, "Responding to Spurious Timeouts in TCP," in Proc. INFOCOM 2003, June 2003.
- [17] 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. of INFOCOM 2000, June 2000.
- [18] G. Carneiro, J. Ruela, and M. Ricardo, "Cross-Layer Design in 4G Wireless Terminals," IEEE Wireless Communications, vol. 11, no. 2, April 2004.

1) 본 연구에서는 여러 TCP 버전 중에서 TCP Tahoe를 사용하였다.