

이기종 무선망 환경에서 Freeze DeadLine 기법을 통한 수직 핸드오버의 성능개선

정 현 진*, 최 승 식^o

Performance Enhancement for Vertical Handover in Heterogeneous Wireless Networks Using Freeze DeadLine

Hyeon-Jin Jeong*, Seung-Sik Choi^o

요 약

본 논문에서는 이기종 무선망 네트워크 환경에서 핸드오버가 발생할 때 생기는 문제점을 해결하기 위하여 FDL(Freeze DeadLine) 알고리즘을 제안하였다. 많은 선행연구들은 하위 계층의 정보를 이용하여 핸드오버 과정에서 생기는 문제를 해결하기 집중하였다. 하지만 본 논문에서는 핸드오버 과정에서 생기는 상위계층에 대한 문제를 다루고 있다. 본 논문에서는 핸드오버가 발생할 때 상위 계층인 전송계층에서 생기는 문제점을 분석하고, 성능을 개선하기 위하여 FDL 알고리즘을 제안하였다. FDL 알고리즘을 이용하여 기존에 제안되었던 기법보다 더 나은 성능을 보임을 이론적으로 분석하였고, 이를 NS-2(Network Simulator -2)를 통해 성능 평가 하였다.

Key Words : FDL, Heterogeneous Wireless Network, Transport layer

ABSTRACT

In this paper, we propose that a FDL(Freeze DeadLine) Algorithm to solve handover problems in heterogeneous wireless networks. Many previous studies concentrated on low level layer to solve handover problems. In this paper, however, we concentrate on high level handover problems such as network layer and transport layer. We analyze handover's problems of transport layer in Heterogeneous wireless network, propose the FDL algorithm for better performance than others studies. The Proposed method is analyzed by theoretical frames and we verify that the propose method using the NS-2(Network Simulation - 2).

I. 서 론

최근 LTE-A(Long Term evolution Advance)가 한국에서 상용화 되었다. 이론적으로 LTE-A는 100Mbps 가 넘는 전송속도를 가지는데, 불과 몇 년 전의 3G네트워크의 데이터 전송속도를 고려해 보면, 무선 네트워크 기술이 비약적으로 발전하고 있음을 알 수 있다.

하지만 가장 빠른 전송속도만을 가지는 무선 네트워크만이 사용되는 것이 아니라 각각의 네트워크가 가진 특징을 고려하여, 사용자의 상황에 맞는 네트워크를 선택하여 사용하고 있다. 현재 사용되고 있는 무선 네트워크는 다양한데 그 예로 LTE, 3G, Wireless LAN(WLAN) 등이 있다. 그런데 최근의 이동 단말(Mobile Terminal)은 하나의 무선 네트워크만을 지원

* 본 연구는 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초 연구사업임(2010-0024926)

• First Author : Incheon National University Department of Computer Science & Engineering, oasishjj@incheon.ac.kr, 학생회원

o Corresponding Author : Incheon National University Department of Computer Science & Engineering, sschoi@incheon.ac.kr, 종신회원
논문번호 : KICS2014-01-001, Received January 7, 2014; Revised April 9, 2014; Accepted August 8, 2014

하는 것이 아니라 그림 1에서와 같이 다양한 네트워크를 지원하고 있다. 이렇게 다양한 망이 존재하는 환경을 이기종 무선 네트워크(Heterogeneous Wireless Networks)라고 하며, 이기종 무선 네트워크를 통해 서비스 제공자는 사용자의 상황과, 요구에 맞는 서비스를 제공하게 된다.

그런데 이기종 무선망을 사용하게 되면 여러 가지 문제가 발생한다. 그 중 대표적인 것은 핸드오버(Handover) 문제이다. 이기종 무선망이 존재하는 환경에서 사용자는 끊임없이 이동을 하며, 이 사이에서 사용자는 자신의 요구사항에 맞는 네트워크를 선택하게 되는데, 기존의 동일한 망에서의 이동인 수평적 핸드오버(Horizontal Handover)와는 다르게 이기종 망에서는 각각의 망이 고유한 특성을 가지고 있기 때문에 이를 핸드오버 과정에서 고려해야 한다. 최근 핸드오버문제를 다루기 위해서 하위계층인 물리 및 링크 계층에서 문제를 해결하기 위해 다양한 접근이 시도되었는데 그 중 최근의 대표적인 연구는 IETF에서 연구되고 있는 802.21 MIH 표준^[1]이다. 표준에 따르면 각각의 핸드오버를 하위계층에서 상위 계층과 독립적으로 수행하여 문제를 해결하고 있다. 이를 통해 핸드오버가 발생하게 되면 끊어짐이 없는 서비스를 제공하게 되는데, 이를 통해 핸드오버에 관한 문제를 해결할 수 있다. 하지만 주로 하위 계층의 문제를 다루고 있기 때문에 TCP계층에 발생하는 Time-out, 재전송 등의 문제는 고려하고 있지 않다^[2].

물리 및 링크계층의 접근이 아니라 상위계층인 전송계층에서 핸드오버 문제를 해결하기 위한 접근도 계속적으로 이루어져 왔다^[3-6]. 그 대표적인 예가 Freeze-TCP^[3]가 있으며 이를 통해 이론적으로

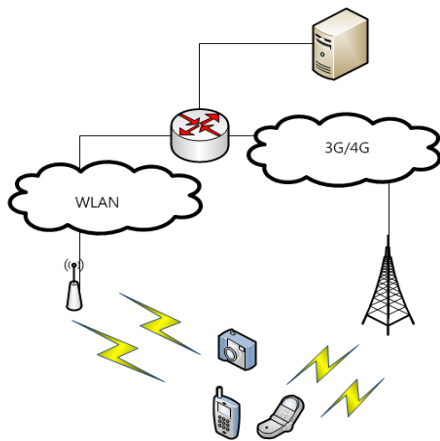


그림 1. 이기종 무선 네트워크의 예
Fig. 1. A Example of heterogeneous Wireless Networks

Transport Layer에서 발생하는 핸드오버 문제를 대부분 해결 할 수 있다.

또 다른 시각으로는 Cross-layer적인 기법들이 연구되고 있다^[4]. Cross-layer적인 접근을 통해 하위 계층에서 발생하는 정보를 이용하여 상위 계층의 핸드오버를 지원하는 방법이다.

그런데 하위 계층의 정보를 토대로 핸드오버를 지원한다고 하더라도, Freeze-TCP가 무선망에 적용된다면 또 다른 문제가 발생하게 된다. Freeze-TCP에서 언제 packet을 freeze 시킬 것인지에 대한 문제이다. 패킷을 너무 빨리 freeze 시킨다면, 데이터를 전송할 수 있음에도 불구하고, 데이터를 전송시키지 못함으로써 문제가 발생할 수 있다. 반대로 너무 늦게 Freeze 시킨다면 Freeze 패킷이 핸드오버 구간에서 손실되어, freeze가 아닌 불필요한 Time-out 및 재전송을 유발하게 되고 이는 핸드오버 처리가 제대로 이루어지지 않게 됨을 의미한다.

본 논문에서는 전송계층에서 하위 계층의 정보를 이용하여, 핸드오버가 언제 발생하는지에 대한 정보를 알고 있다는 가정 하에 Freeze-TCP를 사용함에 있어 언제 패킷을 Freeze 할 것인지에 관한 내용을 제안한다. 이를 Freeze Dead Line(FDL)라 하고, FDL을 통해 Freeze-TCP의 문제점인 불필요한 Time-out 및 세그먼트 재전송을 감소시키고 또한, 패킷 손실 감소를 통해 더 나은 성능을 보임을 확인한다.

핸드오버(Handover)는 이동단말이 하나 이상의 무선 네트워크를 이동할 때 현재의 접속 상태를 유지하기 위한 방법이다^[6]. 핸드오버는 수평적 핸드오버(Horizontal Handover)와 수직적 핸드오버(Vertical Handover)가 있다. 수평적 핸드오버는 같은 망 사이의 이동에서 발생하는 것이며, 수직적 핸드오버는 다른 망 사이에서 발생하게 된다. 수직적 핸드오버는 다시 두 가지 형태로 나누어진다. 첫 번째는 하향식 핸드오버(Downward Vertical Handover)로 이동 단말이 전송범위가 넓은 망에서 좁은 망으로 이동할 때 발생한다. 두 번째는 상향식 핸드오버로 전송범위가 좁은 망에서 넓은 망으로 이동할 때 발생하게 된다. 만약 모바일 단말이 3G/4G 등 무선 이동 통신망을 이용하다가 WLAN(Wireless LAN)망으로 이동하게 되면 이때는 하향식 핸드오버가 된다.

서로 다른 망의 이동인 수직적 핸드오버는 다양한 문제가 발생하게 된다. 첫 번째는 서로 다른 대역폭을 가지고 있기 때문에 대역폭 적응 문제가 발생하게 된다. 3G망에서 WLAN 망으로 이동하는 하향식 핸드오버의 경우에는 급격히 커지는 대역폭으로 인하여

WLAN망에서 빠르게 대역폭에 적응하지 못하는 경우가 발생한다. 이는 과거의 혼잡원도우의 크기를 새로운 망에서도 유지하고 있기 때문에 발생하는 문제이다. 핸드오버의 경우에는 급격히 좁아지는 대역폭으로 인해 많은 패킷이 손실된다. 이 때문에 불필요한 재전송 및 타임아웃 문제가 발생하게 된다.

이와는 다르게 핸드오버 구간에서 발생하는 패킷 손실 문제 및 패킷 순서 문제도 발생하게 된다. 수직적 핸드오버는 서로 다른 망 사이의 이동이기 때문에 이 과정에서 패킷 손실이 발생하게 된다. 만약 많은 패킷 손실이 발생하게 된다면 줄어드는 혼잡원도우의 크기로 인해 낮은 처리량을 가지게 된다. 또한 잘못된 순서의 패킷 전송으로 인해 받은 패킷도 재전송을 하는 경우도 발생하게 된다.

결국 핸드오버에서 발생하는 문제를 해결한다는 것은 핸드오버 과정에서 발생하는 패킷 손실을 줄이고, 이를 통해 새로운 망에 빠르게 적응하여 안정적으로 데이터를 송·수신을 할 수 있게 하는 것이다. 본 논문에서는 이기종 무선 네트워크 환경에서 Freeze-TCP 기법을 기반으로 한 FDL기법을 통해 패킷 손실을 최소화 하고, 핸드오버 이후에 빠르게 대역폭에 적응할 수 있는 기법을 제안한다.

논문의 구성은 다음과 같다. 2장에서는 핸드오버 문제를 해결하기 위한 기존 연구사항을 알아본다. 3장에서는 기존의 Freeze-TCP가 가진 문제 및 이를 수정한 Freeze DeadLine에 대한 내용을 기술하였다. 4장에서는 기존의 연구와 함께 본 논문 내용을 시뮬레이션 결과를 통해 성능을 검증하고, 결론을 맺는다.

II. 관련연구

Freeze - TCP^[3]는 원래의 네트워크에서 다른 네트워크로 이동할 때 이동 단말 자신이 핸드오버 할 것이라는 것을 서버 측에 메시지를 보냄으로써 진행되는 형태이다. 이동 단말은 핸드오버가 일어날 것을 감지하고, 이를 서버 측에 알려 CWND(Congestion Window)를 0으로 만들게 하고, 이를 통해 서버는 데이터 전송을 중지시키는 Freeze 상태로 진입하게 된다. 그리고 이동 단말은 핸드오버가 종료되는 시점에서 다시 서버에게 전송을 재개하라는 메시지를 보내게 되고 이를 수신한 서버는 다시 전송을 시작한다. 이를 통해 핸드오버에서 발생하는 패킷 손실 및 재전송 문제를 해결하게 된다. 하지만 앞에서 언급한 것과 같이 이를 이기종 무선 네트워크에 적용하게 되면 다른 문제가 발생하게 된다. 기본적으로는 핸드오버가

일어난다는 것을 전송 계층에서는 알 수 없다. 하지만 최근 기술의 발달을 통해 이를 알수 있다고 하더라도, 망을 언제 Freeze 할 것이며, 또한 Freeze 메시지를 언제 보내야 하는지에 대한 문제가 발생하게 된다. Freeze를 요청하는 패킷을 너무 빨리 전송하여 서버가 이른 시간에 네트워크를 Freeze시킨다면, 기존의 망을 통해 데이터를 전송할 수 있음에도 불구하고, 데이터를 전송시키지 못하는 상황이 발생하게 된다. 또한 반대로 너무 늦게 freeze 시킨다면 freeze 패킷이 핸드오버 구간에서 손실되어, freeze가 아닌 불필요한 Time-out 및 재전송을 유발하게 되고 이는 전송 계층에서 핸드오버의 처리가 제대로 이루어 지지 않았다는 것을 의미한다.

Freeze-TCP을 개선하여 핸드오버 문제를 해결하기 위한 방법도 제시되었다^[7]. 이 기법은 기본적인 Freeze-TCP위에, 핸드오버 이후 발생하는 Time-out 문제를 해결하기 위하여 새로운 RTT를 계산, 이를 바탕으로 Time-out이 발생하지 않도록 하는 기법이다. 하지만 이 기법은 핸드오버 이후의 성능만 고려하고 있다.

Freeze-TCP가 가지는 문제를 해결하기 위해 Freeze-TCP+^[8]가 제안되었다. 이 기법은 TCP-Vegas를 바탕으로 대역폭의 변화를 예측하고, 남아 있는 데이터의 양을 확인하고, 이를 바탕으로 최적화된 전송률을 지원하기 위한 방법이다. 대역폭의 변화에 따라 새로운 망을 예측을 통해 더 나은 전송률을 제공할 수는 있지만, 적절한 Freeze 시각에 대해서는 기술하지 않고 있다.

최근의 이동 단말은 3G/4G등 이동 통신망의 이용이 가능함은 물론 WLAN등을 이용할 수 있다. 이러한 점에 착안하여 이기종 망으로 Handover할 때 두 개의 망을 모두 활용한 방법들이 제안되었다.

FDCC(Fast adaptive Congestion Control)^[9] 기법은 이기종 무선망 환경에서 저속의 망과 고속의 망을 모두 이용하여 핸드오버를 하는 기법이다. 만약 저속 망에서 고속의 망으로 이동하는 상황에서, 핸드오버가 끝나기 전까지는 저속의 망을 이용하여 데이터를 받고, 대신 ACK 메시지를 고속의 망을 통해서 보낸다. 이를 통해 고속 망의 RTT 정보 및 Bandwidth 정보를 할 수 있으며, 이를 통해 핸드오버 이후에 더 빠르게 대역폭에 적응할 수 있게 한다.

MTCP^[6] 기법은 이기종 무선망 환경에서 이동 단말이 두 개의 망을 동시에 이용가능 할 때, 하나의 연결을 유지하는 것이 아니라 두 개 이상의 망의 연결을 통해 더 높은 처리량을 제공하는 기법이다. 또한 핸드

오버 구간에서도 기존의 망을 끊지 않기 때문에, 핸드 오버에서 발생하는 손실 패킷을 줄일 수 있다.

하지만 두 가지 방법은 두 개의 무선망을 모두 사용이 가능하여야 하며, 만약 하나의 망만이 사용 가능한 경우라면 이용할 수 없는 방법이다. 또한 두 개의 망을 이용함으로써 발생하는 오버헤드(Overhead) 문제도 해결하여야 한다.

핸드오버로 인해 발생하는 대역폭의 급격한 변화 문제를 해결하기 위한 방법이 제안되었다^[10]. TCP 상태를 FA(Fast Adaptation) 구간과 CA(Collision Avoidance) 구간으로 나누고, 각 구간에서 ACK를 기다리거나, 새로운 혼잡윈도우를 계산해서 빠르게 대역폭에 적응하고, 불필요한 혼잡제어를 막아 더 높은 처리량을 가지는 연구이다. 하지만 이 연구는 핸드오버로 인해 급격하게 줄어드는 bandwidth에 집중하여 기술하고 있으며, 알고리즘이 너무 복잡하여 실제 TCP에 적용하기가 힘들다.

FLA(Fast Link Adaptation) 기법^[11]은 변동하는 RTT(Round Trip Time) 정보를 바탕으로 데이터의 전송률을 조절하는 TFRC 네트워크 환경에서 사용하는 기법이다. 하지만 FLA기법은 핸드오버가 발생할 경우 RTT만을 고려하고 있으며, 이로 인해 핸드오버로 인해 대역폭이 변화한 이후에는 혼잡 회피모드로 전송률을 증가시켜 결국 가용대역폭을 점유하는데 오랜 시간이 걸리게 된다^[6,13].

전송 계층에서 핸드오버 문제를 해결하기 위하여 MIH표준을 이용하여 패킷 손실 및 재전송 문제를 해결하기 위한 연구가 진행되었다^[14]. 이 연구 결과에 따르면 MIH표준을 기반으로 TCP-wsetwood, TCP-Vegas, TCP-NewReno 등의 성능을 향상 시켰다. 하지만 이 방법은 상위 계층의 문제를 고려하고 있지 않다. 본 논문에서는 MIH기반위에 상위 계층접근을 통해 문제를 해결하고자 한다.

III. 본 론

무선망 환경에서 전송계층의 핸드오버 문제를 해결하기 위하여 Freeze-TCP를 적용하게 되면, 전송계층의 핸드오버에 관한 많은 문제들을 해결할 수 있다. 하지만 Freeze-TCP는 기본적으로 수평적 핸드오버를 고려해 제작되었고 그로 인해 최근의 경향인 이기종 망에 적용하기에는 많은 문제가 발생한다. 특히 다른 망으로의 이동이기 때문에 서로 다른 RTT 및 전송 속도로 인해 타임아웃 문제가 발생하게 된다. 이를 해결하기 위해서는 언제 세그먼트 전송을 Freeze 시켜야

하는지에 대해 고려해야 한다.

그림 2는 무선 네트워크에서 핸드오버 상황이 발생했을 때 Fast Freeze를 통해 생기는 문제점을 보여주고 있다. 적절한 Freeze 시간을 Freeze DeadLine(FDL)이라 할 때, 이 시점보다 빠른 Freeze는 패킷을 전송할 수 있는 시간을 낭비하는 결과를 보여준다. 그림 2에서 Fast Freeze 구간부터 FDL구간 사이의 빗금이 그려진 구간은, 정상적으로 Freeze했다면 데이터를 전송할 수 있었던 구간이지만, Fast Freeze를 통해 데이터를 전송할 수 없게 되는 구간이다. 정확한 핸드오버 종료시점을 알 수 있다면, 효율적인 FDL을 설정할 수 있을 것이며, 이를 통해 전송할 수 있는 곳에서 데이터를 전송하게 되어 기존의 방법보다 더 효율적인 데이터 전송을 할 수 있게 된다.

그림 3은 늦은 Freeze 상황을 보여주고 있다. 최신의 Freeze 시점을 FDL이라고 하자. FDL 이후에 전송되는 데이터들은 바뀌는 망으로 인해 패킷 손실이 발생한다고 가정한다. 만약 FDL보다 늦은 시점에 단말 노드가 서버 측으로 Freeze를 요청하는 메시지를 보내게 되면, 이 메시지는 핸드오버 처리구간에서 반드시 손실되게 되고, 서버는 망이 바뀌는지 바뀌지 않는지 알 수 없게 된다. 서버는 망의 변화를 인지하지 못했으므로, 평소와 같이 데이터를 전송하게 될 것이며, 오지 않는 ACK를 RTO만큼 대기한 후, 타임아웃 이벤트가 발생한다. 타임아웃 후 재전송되는 패킷은 마지막으로 받은 ACK의 다음 세그먼트가 될 것이다.

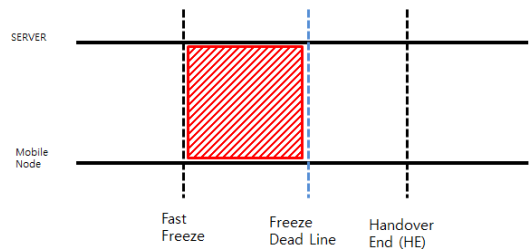


그림 2. 빠른 Freeze
Fig. 2. Early Freeze

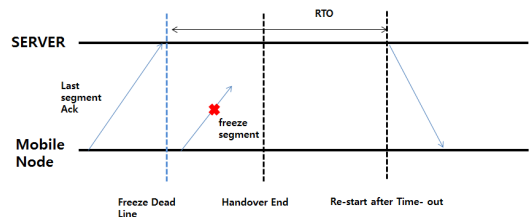


그림 3. 늦은 Freeze
Fig. 3. late Freeze

늦은 Freeze 메시지를 전송하게 되면 핸드오버가 일어나는 것을 정확히 서버에 전달하지 못하게 되고 불필요한 타임아웃 이벤트의 발생을 통해 전체적인 전송률 감소가 발생하게 된다.

앞에서 잘못된 Freeze 시점을 통해 발생하는 문제점에 대해 논의하였다. 이 문제를 해결하기 위하여 본 논문에서는 Freeze DeadLine을 제안한다. FDL은 핸드오버가 발생하는 시기와 핸드오버 종료시점을 하위 계층으로부터 전달 받을 수 있다고 가정한다. 이는 서론에서 언급된 MIH나 Cross-layer적 접근을 통해서 해결할 수 있다. 또한, 핸드오버의 시작지점부터 핸드오버의 종료지점까지는 모든 세그먼트가 손실된다고 가정한다.

본 논문에서는 FDL를 그림 4와 같이, 핸드오버가 종료되는 시점에서 이동 단말이 사용하고 있는 현재 망의 RTT/2 전 시점으로 설정한다. 이를 이용하면 위에서 발생하는 문제들을 해결할 수 있다. 이는 그림과 같이 나타낼 수 있으며, 식 (1)으로 표현된다.

$$FDL_{time} = T_{Handover\ End} - \frac{RTT_{current}}{2} \quad (1)$$

모바일 단말은 스스로 다른 망의 정보를 계속적으로 수집하면서 핸드오버가 발생하는지를 관찰하고 있다. 만약 핸드오버가 발생하게 되면 모바일 단말은 과거의 접속 정보 등을 활용하여 핸드오버 종료시점 (HE)을 계산하게 된다. 만약 핸드오버가 결정되면 이 핸드오버 종료시점 이전의 RTT/2 되는 시점 전에 망을 서버 측에 망을 Freeze할 것을 요청하는 메시지를 전송하게 된다. 이 시점에 메시지를 보내는 것은 Freeze를 위한 패킷이 핸드오버 이전의 망을 통해 전송이 안전하게 전달될 수 있기 때문이다. 만약 핸드오버 종료시점에서 새로운 망으로 변환이 된다고 했을 때, 특정 시점에서 보내진 데이터들은 핸드오버 이전

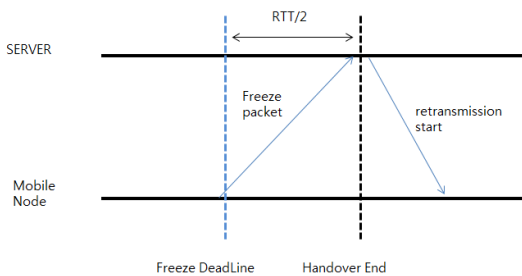


그림 4. 제안 기법
Fig. 4. Propose Method

의 망을 통해 전송되다가 손실된다. 하지만 제안하는 기법은 기존의 망을 통해 Freeze 패킷을 전송하게 하고, 또한 이 패킷의 도착을 일정 시간 전에 전송함으로써 패킷 도착을 보장한다. 이를 통해 늦은 Freeze 패킷 전송 문제를 해결 할 수 있게 한다.

본 논문에서 제안하는 기법인 FDL은 빠른 Freeze 보다 더 나은 성능을 보여준다. 시스템 처리량을 G라고 할 때 식 (2)와 같이 혼잡윈도우(CWND)값을 RTT(Round Trip Time)값으로 나눈 값이라고 하자.

$$G = \frac{CWND}{RTT} \quad (2)$$

이때 FDL을 적용했을 때 얻을 수 있는 시간으로 표현되는 FDL값에서 이른 Freeze가 이루어진 시간의 차를 처리량에 곱해준 값이 이득이 된다. FDL의 이득 값을 $FDL_{benefit}$ 라고 할 때 이는 다음과 같이 표현할 수 있다.

$$FDL_{benefit} = (FDL_{time} - T_{early\ Freeze}) \times G \quad (3)$$

여기에서 FDL_{time} 은 FDL이 시작되는 시간이며, $T_{FastFreeze}$ 는 이른 Freeze가 되는 시점이다. FDL시간과 이른 Freeze시간의 차이는 항상 0보다 크기 때문에 $FDL_{benefit}$ 은 항상 양의 값을 가지게 된다. 이 계산을 통해 본 논문에서 제안하는 기법인 FDL이 기존의 이른 Freeze보다 더 나은 성능을 가진다는 것을 확인할 수 있다.

FDL은 이동 단말에 의해 핸드오버를 주도한다. 만약 이동 단말 Freeze를 해야 한다고 판단하게 되면 서버 측에 메시지를 전송해 망을 Freeze 하게 되는데, 이때 늦은 Freeze는 불필요한 타임아웃(Time out)을 발생시킨다. 제안하는 FDL은 이미 하위 계층의 정보 등을 이용하여 핸드오버 종료시점을 알고 있다. 이를 서버 측에 Freeze를 요청할 때 같이 전송하게 된다. 이를 통해 FDL은 핸드오버가 종료되는 시점에서 데이터를 전송할 수 있지만, 늦은 Freeze는 전송계층에서 핸드오버가 발생했는지의 여부를 알 수 없기 때문에 망이 변경된 것과 상관없이 마지막으로 받은 ACK부터 타임아웃 값으로 설정된 시간까지 기다렸다가 전송이 가능하게 된다. 늦은 Freeze와 FDL을 비교 했을 때의 이득은 식 (4)와 같다.

$$FDL_{benefit} = (RTO - (T_{handover\ end} - FDL_{time})) \times G \quad (4)$$

RTO는 TCP에 의해 설정된 Time out시간이며, $T_{handoverEnd}$ 는 핸드오버가 끝나는 시간이다. 늦은 Freeze에 비해 FDL이 가지는 이득은 그림과 같이 표현될 수 있다. 그림 5에서 빗금으로 표현된 부분이 늦은 Freeze와 비교했을 때 FDL이 가질 수 있는 이득이다. 늦은 Freeze에서는 전송계층에서는 핸드오버가 발생했는지 알 수 없기 때문에 타임아웃이 발생하게 되고, 이를 통해 데이터를 보낼 수 있음에도 데이터를 전송하지 못하는 결과가 발생하게 된다. 이 구간은 TCP에 설정된 RTO에서 FDL이 적용된 후 망이 바뀐 이후 데이터가 전송되는 시간의 차이로 정의된다. 제안된 FDL을 통해 이 구간에서 데이터 전송이 가능하게 된다.

앞의 경우와는 다르게 핸드오버 종료 후 서버 측에 데이터를 요청하는 패킷을 보내어 데이터를 수신하는 방법을 생각해 볼 수 있다. 하지만 이러한 방법도 제안한 FDL에 비해 낮은 성능을 보인다. 그림 5에서와 같이 이동 단말이 핸드오버가 종료되었다는 것을 인지하고 서버 측에 전송 재개 요청을 보내게 된다. 이 메시지는 새로운 망의 $RTT/2$ 이후에 도착할 것이고, 이 메시지 도착 후에 데이터를 전송하게 될 것이다. 하지만 제안하는 기법인 FDL은 새롭게 바뀐 망에서도 이동 단말 측의 메시지를 기다리지 않고, Freeze를 요청하는 시점에서 이미 핸드오버 종료시간을 알고 있기 때문에, 이 시간 이후 바로 데이터를 전송할 수 있게 된다. 여기에서 발생하는 이득은 다음과 같다.

$$FDL_{benefit} = \left(\frac{RTT_{current}}{2} \right) \times G \quad (5)$$

$RTT_{Current}$ 를 새로운 망의 RTT라고 했을 때, FDL에 의한 방법은 핸드오버 이후 이동 단말이 데이터를 요청하는 시점보다 최대 $RTT/2$ 전에 데이터를 전송할 수 있게 한다. 이를 통해 FDL기법이 기존의 방법에서 전송하지 못하는 구간에 데이터를 전송할 수 있게 됨

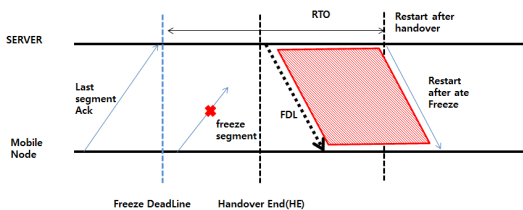


그림 5. 늦은 Freeze와 FDL
Fig. 5. late Freeze and FDL

으로써 더 나은 성능을 보여주는 것을 확인할 수 있다.

FDL기법은 그림 6과 같은 절차로 이루어진다. 이동 단말 측은 핸드오버를 수행하기 위한 파라미터를 측정하고 있다. 만약 파라미터의 값이 임계값 보다 낮아지면 핸드오버를 시작하게 된다. 이후 이동 단말 측은 과거의 핸드오버 정보 등을 활용하여 핸드오버 종료 시간을 예측하게 되고, 이를 서버 측에 Freeze 메시지와 함께 보내게 된다. 이를 받은 서버는 데이터 전송을 일단 중지하고, 이동 단말 측으로부터 받은 메시지 안에 있는 핸드오버 종료시간인지 확인한 후 시간이 지났다면 다시 데이터를 전송하게 된다. 만약 핸드오버 종료시간 전이라면 서버 측은 데이터 전송을 다시 지연시킨 후 시간이 지난 다음에 전송하는 절차로 이루어진다.

FDL을 활용하기 위해서는 핸드오버의 종료 시점 ($T_{handover end}$)을 알고 있어야 한다. 이를 위해서는 다양한 방법들을 적용할 수 있으나, 본 논문에서는 과거의 접속 정보를 활용하여 핸드오버 종료 시점을 예측하고, 이를 FDL 기법에 적용한다. 핸드오버의 종료 시점은 MIH의 Information Service(MIIS)를 이용하여 정의한다. MIIS는 핸드오버 간에 필요한 네트워크들의 정보를 이용하여 단말과 네트워크 간의 정보를 전달하는 기능을 가지는데, 여기에서 전달될 수 있는 정보는 그림 7과 같다¹⁵⁾.

본 논문에서는 Access Network Information에 새로운 파라미터인 Handover Time(HT)을 정의한다. HT는 핸드오버의 시작 시점에서 종료시점까지를 측정한 평균값으로 식 (6)과 같다.

$$HT = \alpha \times HT_{current} + (1 - \alpha) \times HT_{old} \quad (6)$$

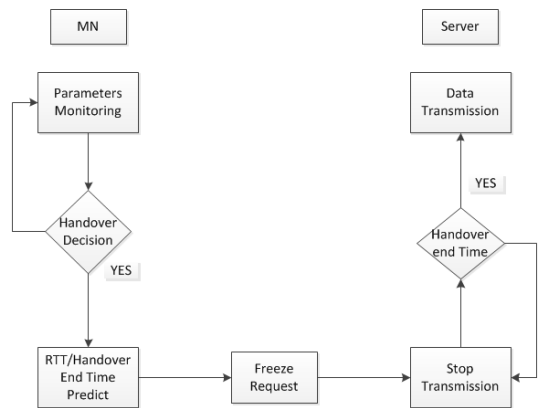


그림 6. FDL의 진행 절차
Fig. 6. A Process of the FDL

General Information	Access Network Information	PoA Specific Information
Operator ID Network Type Country Code ...	Network ID Network QoS Handover Time ...	PoA Link Addr PoA Location PoA System Info ...

그림 7. MIIS의 파라미터
Fig. 7. Parameters of the MIIS

FDL의 진행절차는 그림 8과 같다. 이동 단말은 현재의 네트워크와 통신을 하고 있다. 이 순간에 MIIS는 현재의 네트워크뿐만 아니라 새로운 망의 네트워크 정보 또한 수집하고 있다(Information Update). 현재의 망에서 새로운 망으로 이동하기 위해 주위의 Beacon 또는 네트워크 정보를 수집하는 이동 단말은 이동할 새로운 망에 대한 정보를 얻기 위하여 MIIS에 Information Request 메시지를 전송하게 되고, 이를 받은 MIIS는 이동 단말이 요청한 정보를 보내주게 된다. 이때 식 (6)에서 계산한 타겟 네트워크의 HT도 전송하게 된다. 만약 MIIS에서 망의 정보가 없다면, 비슷한 망의 정보를 검색하여 전송하게 된다. MIIS로부터 전송 받은 자료를 토대로 이동단말은 FDL을 설정하고, HT를 서버로 전송하게 된다. 그리고 핸드오버 절차가 진행되며, 이때 서버는 이동 단말로 부터 받은 HT를 토대로 데이터 전송을 중지 시켰다가, HT이 지난 후 데이터 전송을 재개한다.

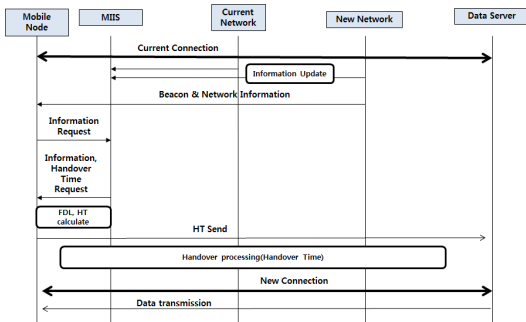


그림 8. FDL 처리 흐름도
Fig. 8. Flow-chart of the FDL Process

IV. 실험

제안하는 기법의 성능 평가를 위하여 NS-2 (Network Simulator -2)^[11]를 이용하여 성능평가를 진행하였다. 실험환경은 그림 9와 같이 하나의 서버와 하나의 라우터, 그리고 3G 기지국과 무선 랜

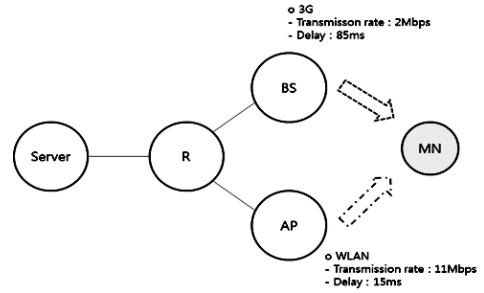


그림 9. 시뮬레이션 토폴로지
Fig. 9. Simulation Topology

AP(Access Point)가 존재하는 환경에서 실험을 진행하였다. 서버와 라우터, 그리고 BS(Base station), AP는 유선으로 연결이 되어 있으며, 모바일 단말과는 무선으로 연결이 되어있다. 시뮬레이션이 시작되면 서버는 현재의 망 중에서 모바일 단말과 연결된 망을 통해 연결절차를 시작한다. 그리고 2초 후인 10초 지점부터 새로운 망의 연결이 이루어져 이때부터 데이터를 전송할 수 있게 된다. 이때, 패킷 사이즈는 1000Byte 로 FTP 트래픽을 이용하였다. 3G 네트워크의 대역폭은 2Mbps의 가용대역폭과 85ms의 지연시간을 설정하였고 WLAN은 가용대역폭 11Mbps와 15ms의 지연시간을 가진다. 전송계층의 프로토콜은 총 4가지로 구성되고 진행되었으며 이는 Freeze-TCP, FDL기법을 적용한 TCP-Reno, 그리고 TCP-Reno, TCP-SACK을 통해 진행되었다. TCP-SACK은 에러가 발생하는 환경에서 제안하는 기법인 FDL과 비교를 위해 사용되었다. 실제 무선 네트워크 환경에서는 많은 에러가 발생하기 때문에 제안하는 기법이 어떠한 성능을 보이는지 확인하기 위하여, 에러가 발생하는 상황과 그렇지 않은 상황을 각각 나눠 실험을 진행하였다. 에러의 발생은 1%로 설정하였다. 실험은 모바일 단말이 3G망에서 WLAN망으로 이동하는 하향식 핸드오버 상황과, 이와는 반대로 WLAN망에서 3G망으로 이동하는 환경을 나눠서 실험을 진행하였다. 그리고 CWND (Congestion Control Window)를 측정하여 핸드오버 이후 망의 대역폭에 빠르게 적응할 수 있는지를 확인하였다. 이 파라미터들은 관련연구에서 사용된 값을 참고하여 적용하였다^[12,13].

그림 10은 하향식 핸드오버 상황에서 FDL, TCP-Reno, Fast-freeze 기법의 처리량을 비교한 그래프이다. 그래프에서 5초 이후의 상황을 보면 Fast-freeze 기법은 너무 일찍 망을 freeze 하면서, 데이터를 전송할 수 있는 시간에 전송하지 못하는 것을 확인할 수 있다. 하지만 제안하는 기법인 FDL 기법은 핸드오버가 종료하는

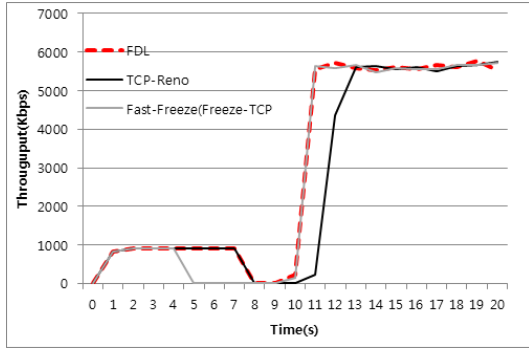


그림 10. 하향식 핸드오버 상황에서의 처리량
Fig. 10. Throughput in Downward Vertical Handover

시점을 알고 있고, 이 시점을 통해 메시지를 언제 보내야 하는지 알고 있으므로, 5초 이후에도 데이터를 전송하고 있는 것을 확인할 수 있다. 그리고 10초 이후에는 TCP-Reno 기법에 비해 제안하는 기법이 더 빠르게 데이터를 전송하는 것을 확인할 수 있다. TCP-Reno는 핸드오버가 언제 발생할지, 언제 종료되는지 알지 못한다. 단지 RTO를 통해 패킷에 대한 ACK 패킷이 수신이 되지 않는다면, 타임아웃을 통해 혼잡제어를 수행할 뿐이다. 그렇기 때문에 핸드오버 구간에서 계속적으로 타임아웃이 발생하게 되고 이를 통해 CWND가 낮아져 핸드오버 이후에도 낮은 속도로 데이터를 전송할 수밖에 없다.

그림 11은 상향식 핸드오버 상황에서 FDL, TCP-reno, Fast-Freeze 기법의 처리량을 비교한 그래프이다. 하향식 핸드오버 상황과 마찬가지로 5초 이후에는 Fast-freeze로 인해 데이터 전송을 하지 않는다. 하지만 제안하는 기법은 전송 가능한 범위에서 최대한 데이터를 전송하게 되므로, 더 많은 데이터를 전송할 수 있다. 핸드오버 이후에는 TCP-reno보다 더 빠르게 데

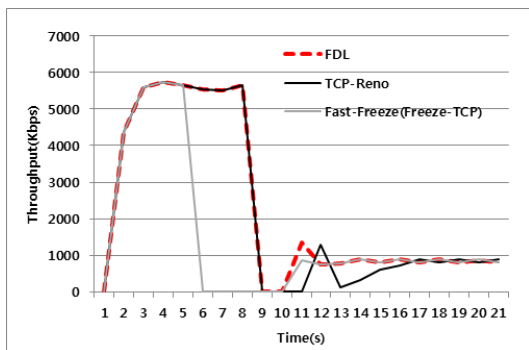


그림 11. 상향식 핸드오버 상황에서 처리량
Fig. 11. Throughput in Upward Vertical Handover

이터를 전송하게 되는데, 이는 앞서와 같이 CWND 값이 핸드오버 이후에도 유지되기 때문이다.

앞의 실험과 동일한 조건에서 확률적으로 에러가 발생하는 환경에서의 시뮬레이션을 진행하였다. 본 실험을 진행한 이유는 실제 무선 네트워크를 사용하는 환경에서는 회절, 간섭, 등의 이유로 패킷이 많이 손실되기 때문이다. 이 실험의 에러율은 1%로 설정하여 실험을 진행하였다. 실제 환경에서는 에러율이 일정하지 않으나, 실험의 편의를 위해 1%로 설정하였다. 그림 12는 하향식 핸드오버가 발생할 때 TCP-SACK과 FDL기법의 처리량 비교이다. 처음의 망에서는 비슷한 성능을 보이고 있다. 하지만 핸드오버 이후에는 FDL기법이 더 빠르게 데이터 전송을 시작한다. 이 이유는 앞서 설명한 것과 같이 CWND의 크기 및 제안하는 알고리즘이 Freeze패킷을 보낼 때 핸드오버 종료시간을 알려주기 때문에 그 시간 이후에 데이터를 전송하게 된다. 그림 13은 상향식 핸드오버가 발생했을 때의 비교이다. 역시 앞서의 결과와 같이 더 빠르게 데이터를 전송하는 것을 확인할 수 있다. 두 실험에서 핸드오버 이후에 FDL기법은 데이터를 기존의 방법보다 먼저 수신하는 것을 확인할 수 있다. 본 논문은 핸드오버 과정에서 발생하는 첫 번째 타임아웃 문제 해결을 통해 더 나은 성능을 보이는 것을 확인할 수 있다.

그림 14는 하향식 핸드오버 상황에서 시간에 따른 혼잡윈도우의 크기를 나타내고 있다. 앞서의 처리량의 경우처럼 핸드오버가 시작되는 지점보다 앞서서 망을 Freeze 한 경우에는 ACK 패킷을 받지 못하기 때문에, 지속적인 CWND 하락을 겪게 된다. 새로운 망으로 바뀐 후 빠르게 적응하는 것을 볼 수 있다. 혼

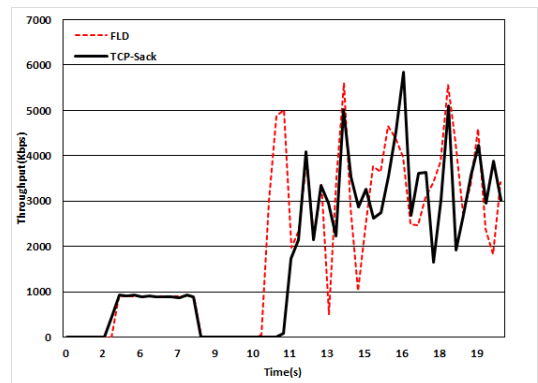


그림 12. 하향식 핸드오버 상황에서 에러가 발생할 때의 처리량
Fig. 12. Throughput in Downward handover when errors are occurred

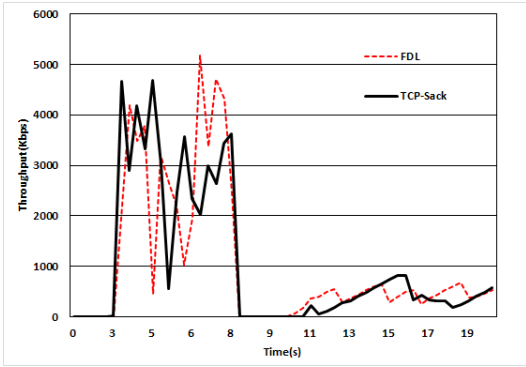


그림 13. 상향식 핸드오버 상황에서 에러가 발생할 때의 처리량
Fig. 13. Throughput in Upward handover when errors occurred

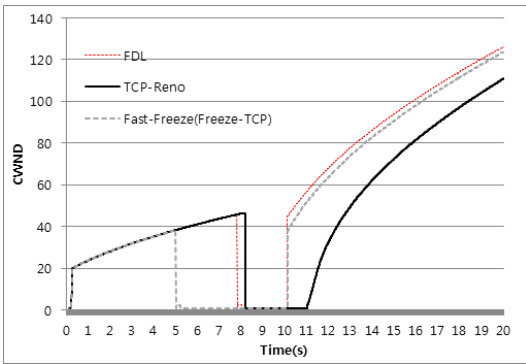


그림 14. 하향식 핸드오버 상황에서 CWND
Fig. 14. CWND in Downward handover

잡원도우의 크기를 통해 타임아웃이 언제 일어났는지 확인할 수 있다. (그림 14)에서 TCP-Reno의 성능을 보면 핸드오버가 끝나는 10초 이후의 상황에서도 계속적으로 CWND값을 1로 유지한다. 제안하는 기법은 FDL기법은 핸드오버 구간에서 CWND를 0으로 유지하고 있다. 하지만 핸드오버 이후 망이 재개되는 시점에서 타임아웃에서 Slow-start를 수행하게 된다. 이를 통해 빠르게 대역폭에 적응하는 것을 확인할 수 있다.

그림 15는 상향식 핸드오버 상황에서의 CWND 크기를 시간순서대로 나열한 결과이다. 상향식 핸드오버 상황에서도 하향식 핸드오버의 상황과 비슷한 결과가 나오는 것을 확인할 수 있다. 두 실험에서 보듯이 핸드오버 이후에 제안한 기법인 FDL기법은 기존의 기법보다 더 빠르게 전송을 시작하는 것을 확인할 수 있다. 실험을 통해 확인할 수 있는 것은 기존의 방법보다 핸드오버 이후에 빠르게 데이터를 전송할 수 있다는 것이다.

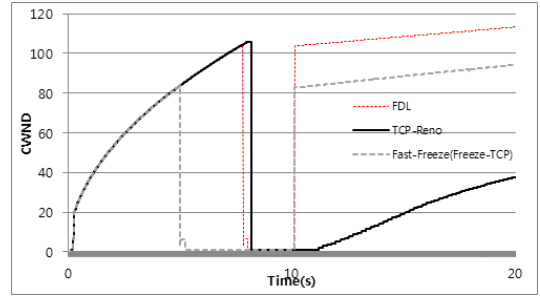


그림 15. 하향식 핸드오버 상황에서 CWND
Fig. 15. CWND in Upward handover

핸드오버 종료 시점의 예측에 실패했을 경우를 고려하기 위하여, 추가적인 시뮬레이션을 진행하였다. 잘못된 연산 등 외부 요인을 통해 종료 시점의 예측에 실패했을 때는 크게 두 가지 상황으로 나누어 볼 수 있다. 첫 번째 상황은 종료시점이 실제 망이 재개되는 시점보다 일찍 예측이 되는 경우이다. 이 경우에는 패킷을 전송하더라도 이동 단말에서는 패킷을 수신할 수 없기 때문에 타임아웃이 발생하게 되고 기존의 TCP와 같이 동작하게 된다. 두 번째 경우는 종료시점의 예측이 망이 재개된 이후보다 늦게 예측이 이루어지는 경우이다. 이에 따른 영향을 살펴보기 위하여 추가적인 시뮬레이션을 진행하였다. 실험 방법은, 100% 종료 시점에서 의도적인 지연을 추가하여, 망 재개 후 바로 데이터를 전송하는 것이 아니라 지연시간동안 대기 했다가 데이터를 전송하는 방식으로 진행하였다. 실험은 상향식 핸드오버 상황에서만 실험을 진행하였고, 9~20초간의 결과를 그림 16에 나타내었다.

그림 16에서 확인 할 수 있듯이, 잘못 계산되어진 종료시점은 100%완벽히 종료 시점을 예측한 기법보다 늦게 데이터를 전송하는 것을 확인할 수 있다. 이는 망을 재개하는 시점이 잘못 적용되어 발생하는 경

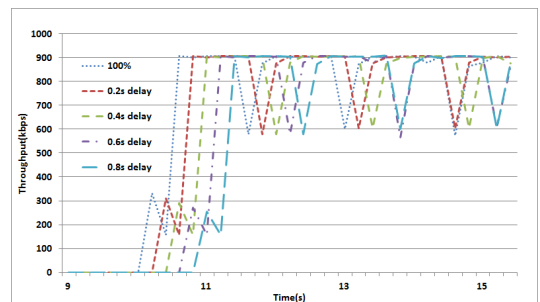


그림 16. 상향식 핸드오버 상황에서 종료시점 예측이 실패했을 경우의 처리량
Fig. 16. Throughput in Upward handover when the prediction of endpoint is failed

우이다. 그렇기 때문에 종료시점을 정밀하게 측정할 수 있는 연구가 지속적으로 진행되어야 한다.

그림 17은 그림 10의 하향식 핸드오버 상황에서 각각의 기법들이 가지는 타임아웃의 횟수와 핸드오버 종료 이후 새로운 망에서 데이터 전송 시작 시간을 나타낸 그림이다. 타임아웃의 발생은 핸드오버의 시작에서부터 핸드오버가 종료되는 시점사이에서만 발생하였다. 제안한 기법인 FDL과 기존의 Fast-Freeze 상황에서는 핸드오버 과정에서 발생하는 타임아웃은 핸드오버 종료 이후의 새로운 망에서의 전송률에는 영향을 미치지 않는다. 왜냐하면 새로운 망이 재개되는 순간 종료시점을 알고 있는 송신측은 데이터 전송을 바로 재개하기 때문이다. 그렇기 때문에 그림 17에 나타난 FDL 및 Fast-Freeze의 타임아웃은 의미를 가지지 않는다. 반대로, 기존의 기법인 TCP-Reno는 타임아웃 발생횟수가 FDL과 Fast-Freeze에 비해서 적더라고 하더라도, 타임아웃에 의해 새로운 망에서의 전송 시작 시간의 영향을 받게 된다. 이는 핸드오버가 끝나더라도 TCP는 핸드오버가 끝났는지, 아닌지를 알 수 없기 때문에 망이 재개된 이후에도 아직 타임아웃에 걸려 있는 상태이기 때문이다. 그렇기 때문에 새로운 망에서의 데이터 전송 시작 시간은 제안하는 기법보다 늦어지게 된다.

	FDL	Fast-Freeze	TCP-Reno
The number of Time-out	11	25	4
Restart Time Of Transmission	10.1396s	10.139s	11.069s

그림 17. 타임아웃 횟수와 새로운 망의 데이터 전송 시작 시간
 Fig. 17. The number of timeout and the start time of data transmission in new networks

V. 결 론

이기중 무선망 환경에서 핸드오버 문제는 여러 계층을 종합적으로 고려해 해결해야 한다. 하지만 현재의 연구는 대부분 하위 계층인 물리 및 Link 계층을 고려하고 있다. 하지만 상위계층인 전송계층에서도 핸드오버로 인해 발생하는 패킷 손실 및 전송률 저하에 민감하기 때문에 이를 고려하여야 한다. 그 중 대표적인 연구로 Freeze-TCP가 있다. 하지만 Freeze-TCP는 언제 망을 Freeze 해야 할지 알지 못한다. 그래서 본 논문에서는 Freeze의 적정시간을 구하고 이를 Freeze DeadLine 이라고 명명하였다. 또한 기존의 방법보다

좋은 성능을 보이는지 이론적으로 기술하였고, 이를 NS-2를 통해 상향식 핸드오버와 하향식 핸드오버의 상황 모두 고려하여 실험을 진행하였다. 또한 CWND의 비교를 통해 기존의 방법보다 더 나은 성능을 보이는 것을 확인할 수 있었다. 향후 연구과제로는 정확한 FDL을 설정하기 위해 핸드오버 종료시점을 정밀하게 예측할 수 있는 기법을 연구할 예정이다. 그리고 실제 환경에서 제안한 기법을 실행할 때 발생하는 문제점들을 확인할 예정이다.

References

- [1] IEEE, "IEEE standard for local and metropolitan area network-Part 21: Media Independent Handover Services," IEEE std 802.21-2008, pp. 1-90, 2009.
- [2] C. Lim and J. Jang, "A new scheme to improve overall TCP throughput with vertical handover between 3G cellular packet networks and wireless LANs," *EURASIP J. Wirel. Commun. Netw. archive*, vol. 2009, Jan. 2009.
- [3] G. Tim, M. James, DS. Phatak, and G. Vipul, Mar. "Freeze - TCP: A true end-to-end TCP enhancement mechanism for mobile environments," in *Proc. IEEE INFOCOM 2000*, pp. 1537-1545, Israel, 2000.
- [4] Rutagemwa H, Pack S, Shen X, Mark J, "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.
- [5] J. Fitzpatrick, S. Murphy, M. Atiquzzaman, and J. Murphy, "Using cross-layer metrics to improve the performance of end-to-end handover mechanisms," *J. Comput. Commun.*, vol. 32, no. 15, pp. 1600-1612, Sept. 2009.
- [6] H. Jeong, S Choi, "Transmission rate control in heterogeneous wireless networks using multiple connections," *J. KICS*, pp. 993-1003, 2012.
- [7] G.M.T. Da Costa and H.R. Sirisena, "Freeze TCP with time stamps for fast packet loss recovery after disconnections," *J. Comput. Commun.*, vol. 26, no. 15, pp. 1792-1799, 2003.

- [8] H. H. Lee and J. H. Choi, "Enhanced freeze-TCP for Vertical Hand-over," *Conf. IEEK*, vol. 2010, no. 6, pp. 887-889, May 2010.
- [9] N. Wang, Y. Chen, C. Cheng, Y. Chiang, and Y. Wang, "Improving TCP performance with fast adaptive congestion control during soft vertical handoff," *Telecommun. Syst.*, vol. 52, no. 1, pp. 97-104, Jun. 2011.
- [10] E. Ko, D. An, I. Yeom, and Yoon, "Congestion control for sudden bandwidth changes in TCP," *Int. J. Commun. Syst.*, vol. 25, no. 12, pp. 1550-1567, Dec. 2011.
- [11] The Network Simulator NS-2,
- [12] D. Li, K. Sleurs, E. Lil, and A. Capelle, "Fast link adaptation for TFRC after a handover," in *Proc. IEEE Int. Symp. Personal, Indoor and Mobile Radio Commun.*, pp. 1-5, Cannes, Sept. 2008.
- [13] S. Ko and K. Chung, "Rate adaptation scheme for video streaming service in vertical handover environments," *J. KIISE : Inf. Netw.*, vol. 38, no. 6, pp. 439-449, 2011.
- [14] V. Tiwari, S. Kansal, and A. Gaiwak, "Performance evaluation of TCP variants using media independent handover in heterogeneous network," in *Proc. Int. Conf. Comput. and Commun. Technol.*, pp. 367-370, Sept. 2010.
- [15] "IEEE Standard for Local and metropolitan area networks – Part 21: Media Independent Handover Services," Jan. 2009.

정 현 진 (Hyeon-Jin Jeong)



2012년: 인천대학교 경영 및 컴퓨터공학과 졸업 (학사)
 2014년: 인천대학교 대학원 컴퓨터공학과 졸업 (석사)
 2014년~현재: 인천대학교 컴퓨터공학과 박사과정
 <관심분야> 무선 인터넷 프로토콜, 라우팅 프로토콜

최 승 식 (Seung-Sik Choi)



1988년: 연세대학교 전자공학과 졸업 (학사)
 1990년: KAIST 대학원 전기 및 전자공학과 졸업 (석사)
 2002년: KAIST 대학원 전기 및 전자공학과 졸업 (박사)
 1990년~2004년: KT서비스 개발연구소 선임연구원
 2004년~현재: 인천대학교 컴퓨터공학과 부교수
 <관심분야> 무선 엑세스제어, 무선자원관리, 무선 인터넷 프로토콜