

# 이동 컴퓨팅 환경에서 핸드오프를 고려한 TCP 성능 개선 방안

박종문, 송수석

연세대학교 컴퓨터과학과

## A Strategy for Improving TCP Performance Considering Handoff in Mobile Computing Environments

J.M. Park, J.S. Song

Dept. of Computer Science, Yonsei University

### 요 약

오늘날 통신 환경에서 널리 사용되고 있는 TCP는 유선망과 고정된 호스트로 이루어진 전통적인 네트워크에 적합하며 유선망의 특성을 적용한 것이기 때문에, 무선 구간의 제한된 대역폭, 높은 지연, 산발적인 비트 에러, 일시적인 연결 두절 및 핸드오프 등과 같은 특징을 가진 특성상 유선망과는 상이한 무선망에 이를 그대로 적용하게 되면 오히려 불필요한 메커니즘의 호출로 인해 End-to-End Throughput의 성능 저하를 가져온다. 이러한 무선망에서 발생하는 핸드오프로 인해 TCP 성능이 저하되는 것을 방지하기 위한 방안으로 핸드오프 과정 동안 무선 구간에서 발생하는 패킷 손실로 인한 고정 호스트 재전송 타이머의 타임아웃과 불필요한 혼잡제어 절차의 수행에 따른 성능감소를 미연에 방지하기 위한 방안으로 핸드오프가 발생했을 때 핸드오프를 가장 먼저 알게되는 이동 호스트가 ICMP(Internet Control Message Protocol) 형태의 EHN(Explicit Handoff Notification) 패킷을 고정 호스트에 전송하여 새로운 기지국과 고정 호스트에 핸드오프의 시작을 명시적으로 알림으로써 핸드오프 과정 동안에 발생하는 고정 호스트 재전송 타이머의 타임아웃과 혼잡제어 절차를 강제로 일어나지 않게 하여 기지국에서의 부하를 감소하게 할 수 있는 새로운 방식의 EHN 알고리즘을 제안한다.

#### 1. 서 론

최근 무선 통신의 대중화가 증가함에 따라 무선망이 미래의 인터넷 정보 사회에서 중요한 역할을 담당하게 될 것으로 기대된다. 무선망의 출현으로 기존의 인터넷 환경은 유/무선이 통합된 단일 망으로 변화하고 있다. 이러한 인터넷 통합 망에서 우선적으로 직면하는 문제는 역시 단말기의 이동성이며 이에 따라 어드레싱과 라우팅 기능을 보장하여 단말기의 이동성에 대응할 수 있는 Mobile IP에 대한 연구가 최근 깊이 이루어져 왔다. 그러나, 이러한 네트워크 계층에 대한 문제 이외에 End-to-End 신뢰성 있는 연결을 보장하기 위한 TCP(Transmission Control Protocol)의 유효성에 대한 검증은 해결해야 할 과제로 남아 있다. 오늘날 통신 환경에서 널리 사용되고 있는 TCP는 유선망과 고정된 호스트로 이루어진 전통적인 네트워크에 적합하며 유선망의 특성을 적용한 것이기 때문에, 무선 구간의 제한된 대역폭, 높은 지연, 산발적인 비트 에러, 일시적인 연결 두절 및 핸드오프 등과 같은 특징을 가진 특성상 유선망과는 상이한 무선망에 이를 그대로 적용하게 되면 오히려 불필요한 메커니즘의 호출로 인해 End-to-End Throughput의 성능 저하를 가져온다[4]. 이런 Throughput의 성능 저하는 TCP가 유선망의 낮은 비트 에러율 때문에 통신상에서 발생하는 패킷 손실을 기본적으로 혼잡에 의한 것으로 생각하여 혼잡 제어 메커니즘으로 패킷 손실을 처리하기 때문에 발생한다. TCP 송신자는 패킷 손실을 발견하면 먼저 전송 윈도우 크기를 줄이고 손실된 패킷을 재 전송한다. 또한 혼잡 제어나 회피 메커니즘을 초기화[2]하고 재전송을 위한 타이머의 값을 증가시킨다. 하지만 무선망에서의 패킷 손실은 대부분 혼잡에 의해서가 아

니라 핸드오프 시의 disconnection이나 높은 비트 에러율에 의해서 발생하게 된다. 그러므로 앞에서 언급한 TCP의 패킷 손실 회복 메커니즘을 그대로 무선망에 적용시키면 오히려 불필요한 성능 저하를 가져오게 된다. 그리고, 이러한 유선 망에서의 TCP 프로토콜을 이동 호스트의 이동성으로 인한 핸드오프 특징을 가진 무선 환경에 그대로 적용될 경우, 이러한 동작은 유선 구간의 대역폭 이용 면에서 불필요한 감소를 초래하며 처리율의 저하 및 지연을 증가시켜 전체 망의 성능을 저하시키게 된다. 본 논문에서는 무선망 상에서의 핸드오프가 있을 때 TCP 수행을 향상시키기 위해 제안된 기존의 연구들에 대해서 살펴본다. 그리고 무선망에서 발생하는 핸드오프로 인한 TCP 성능이 저하되는 것을 방지하기 위한 방안으로 새로운 알고리즘을 제시한다.

#### 2. 핸드오프를 고려한 기존의 연구

네트워크가 유/무선망을 통합한 단일 망으로 구성되었을 때, 무선 구간에서 이동 호스트의 핸드오프로 인한 TCP의 성능 저하를 방지하기 위해 현재 제안되고 있는 프로토콜들을 간략하게 살펴보면 다음과 같다. Fast Retransmission 알고리즘은 고정 호스트 즉, 송신노드가 무선망의 존재 여부를 알고 있는 End-to-End 프로토콜로써 throughput과 지연에서 이동 호스트의 이동에 따른 그 영향을 측정하고, 성능 손실의 원인이 되는 요소들을 식별하여 핸드오프가 완료된 이후에 네트워크 레벨에서 통신이 재개될 때 통신이 잠시 멈추는 시간을 감소시키는 빠른 재전송을 수행한다. 즉 핸드오프 과정이 끝난 이후에 이동 호스트가 마지막으로 수신한 패킷에 해당하는 3개의 중복 Ack 또는, 특정한 TCP

Ack 패킷을 고정 호스트로 전송하여 재전송 타이머를 기다리지 않고 손실되었다고 생각되는 세그먼트에 대해 빠른 재전송 절차를 수행하게 함으로써 핸드오프 시간을 줄이는 방법이다[3]. Fast Retransmission은 핸드오프를 알리는 신호가 도착할 때에만 빠른 재전송을 호출하기 때문에 기존 고정 호스트상의 소프트웨어에는 단지 최소한의 변화만이 요구되고, 기반이 되는 네트워크나 중간 라우터로부터 특별한 지원에 의존하지 않는 장점을 가진 반면에, 무선망에서의 패킷 손실의 주된 원인중 하나인 핸드오프 시 disconnection으로 인한 패킷 손실 해결책만을 제시하며 무선망 자체의 높은 비트 에러율에 의한 패킷 손실은 간과하고 있고, 핸드오프 이후에 이전의 완전한 대역폭에 도달하기 위해서는 몇 번의 RTT(Round Trip Time)가 경과해야만 하고, 송신자가 언제나 빠른 재전송을 시작하기 전에 전송 윈도우 크기를 줄임으로써 초당 전송률을 감소시키는 단점이 있다.

PROBE 알고리즘은 TCP 성능에서 핸드오프의 영향을 감소하기 위해서 TCP로부터 이동 호스트의 이동을 숨기는 Split-connection 프로토콜로써, 핸드오프가 완료된 이후에 이동 호스트가 고정 호스트로 3개의 Negative Ack를 전송하여 해당 패킷에 대한 빠른 재전송이 발생하게 하는 방법이며, BUFFER+FREEZE 알고리즘은 이동 호스트의 mobility에 의한 패킷 손실과 네트워크 혼잡 때문의 패킷 손실사이를 구별하기 위해서 고정 호스트의 TCP가 이동 호스트의 Mobile을 알게 만드는 End-to-End 프로토콜로써, 핸드오프 과정 동안에 기지국이 Ack 패킷 내에 수신자의 버퍼 상태를 나타내는 advertized window의 값을 "0"으로 설정한 후 고정 호스트로 전송하여 고정 호스트를 persist mode로 동작하게 하였으며 핸드오프가 종료된 이후에 새로운 Ack를 수신하면 정상적인 동작으로 전환하는 방법이다[1]. 이 알고리즘에서는, 기지국이 항상 Ack 패킷을 저장하여 두었다가 전송을 해야하며 기지국에서 계층의 투명성이 보장되지 않는다는 문제가 있다.

Snoop protocol은 핸드오프 시 발생하는 데이터 손실을 줄이기 위해서 새로운 방식의 라우팅 프로토콜을 제시한 Split-Connection 방식이다. 이 알고리즘은 핸드오프 과정에서 발생하는 지연을 없앨 수 있고, 비트 에러율이 높은 경우에는 상당한 성능 향상을 나타내고 있지만, 핸드오프가 진행되는 상태에 관한 많은 정보를 기지국에서 가지고 있어야 하고, 멀티캐스트 라우팅을 수행함으로써 불필요한 데이터 전송과 버퍼링으로 기지국에 상당히 큰 부하를 주는 단점을 가지고 있다.

3. EHN(Explicit Handoff Notification) 알고리즘 제안

본 논문에서는 이러한 무선망에서 발생하는 핸드오프로 인해 TCP 성능이 저하되는 것을 방지하기 위한 방안으로 핸드오프 과정 동안 무선 구간에서 발생하는 패킷 손실로 인한 고정 호스트(FH : Fixed Host) 재전송 타이머의 타임아웃과 불필요한 혼잡제어 절차의 수행에 따른 성능 감소를 미연에 방지하기 위한 방안으로 EHN 알고리즘을 제안한다.

EHN 알고리즘은 핸드오프가 발생했을 때 핸드오프를 가장 먼저 알게 되는 이동 호스트(MH : Mobile Host)가 ICMP(Internet Control Message Protocol) 형태의 EHN(Explicit Handoff Notification) 패킷을 고정 호스트에 전송하여 새로운 기지국과 고정 호스트에 핸드오프의 시작을 명시적으로 알림으로써 단지 해당 기지국에서만 상태정보를 유지하게 하여 기지국에서의 부하를 감소하게 할 수 있고, 핸드오프 과정 동안에 발생하는 고정 호스트 재전송 타이머의 타임아웃과 혼잡제어 절차의 혼잡 제어 절차를 미연에 방지함으로써 핸드오프 이전의 전송 윈도우 크기를 핸드오프 이후에도 사용하게 하여 망 성능의 저하를 미연에 방지할 수 있다. 이렇게 함으로써 핸드오프 이전에 사용하던 대역폭을 핸드오프 이후에도 그대로 다시 사용할 수 있으며 이러한 수행으로 인해 망 혼잡과 핸드오프를 쉽게 식별하여 대처할 수 있을 것이다.

EHN 패킷에는 핸드오프 과정 동안 무선 구간에서 발생하는 패킷 손실로 인한 고정 호스트 재전송 타이머의 타임아웃과 불필요한 혼잡제어 수행에 따른 성능 감소를 미연에 방지하기 위한 방안으로 송신노드인 고정 호스트를 persist mode로 만들어 방지하기 위한 송신국이 어떠한 패킷도 전송을 하지 못하게 수신기의 버퍼 크기를 의미하는 광고된 윈도우의 크기를 "0"으로 설정하는 내용과 이동 호스트에서 핸드오프 이후의 새로운 기지국에 대한 주소가 포함된다.

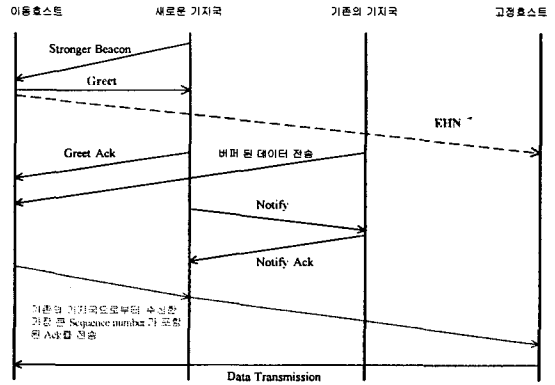


그림 1 : EHN 알고리즘에서 핸드오프 메시지 교환 절차

핸드오프 과정에서 EHN 패킷의 교환 절차는 [그림 1]과 같다. 무선망의 핸드오프 과정에서 이동 호스트가 새로운 beacon을 수신하게 되어 핸드오프의 시작을 알게되면 새로운 기지국에는 기존의 기지국과 자신의 IP 주소이외에 고정 호스트의 IP 주소를 포함하는 Greet 패킷을 보내고, 고정 호스트에는 이동 호스트의 광고된 윈도우를 "0"으로 설정하는 필드와 이동 호스트의 새로운 기지국의 주소를 포함하는 EHN 패킷을 전송한다. 새로운 기지국은 이에 대한 Greet Ack을 이동 호스트로 전송한다. 핸드오프가 완료되면, 이동 호스트는 기존의 기지국에서 수신한 데이터 중 가장 큰 Sequence number의 정보를 고정 호스트에 보내 핸드오프가 완료된 후에 고정 호스트가 그 Sequence number이후부터 데이터를 전송할 수 있도록 한다.

고정 호스트가 EHN 패킷을 수신하면 현재의 시점에서 송신 윈도우의 값을 "0"으로 바꾸어 스스로 Persist mode로 바꾼 후에 핸드오프의 종료로 알리는 이동 호스트의 Ack를 기다린다. Persist mode에서는, 고정 호스트에 있는 모든 상태와 재전송 타이머는 멈추게 되고, Persist timer를 이용하여 수신노드에게 패킷을 보낼 수 있는 상태로 윈도우가 증가했는지를 물어보는 Probe 패킷을 주기적으로 보내게 되는데, 이 Probe 패킷에 대한 처리는 새로운 기지국이 실시한다. 그리고, 핸드오프 이후에 이동 호스트로부터 핸드오프의 종료로 알리는 새로운 Ack가 새로운 기지국을 통해서 전송되면, 고정 호스트는 핸드오프 이전의 advertized window 값으로 설정함으로써 Persist mode에서 나오게 되어 핸드오프 이전에 사용하던 윈도우 크기를 그대로 사용함으로써 TCP 성능을 최대한으로 유지할 수 있다. EHN 패킷은 핸드오프가 이루어지는 동안에만 한번 발생되므로 TCP의 연결동안에 귀한 경로의 트래픽에 큰 영향을 주지 않는다.

4. 제안한 알고리즘 성능 분석

4.1 제안된 알고리즘 성능 평가 방법

본 논문에서는 제안된 알고리즘의 성능을 평가하고 무선 링크들을 시뮬

레이트 하기 위해 확장된 Netsim이라는 시뮬레이터를 기반으로 하여 결과를 도출해 낸다. TCP가 무한한 데이터를 생성한다고 가정하기 위해 전송할 데이터에 시뮬레이터에서 제공하는 최대값(140000000)을 취한다. 대용량의 데이터가 고정 호스트에서 이동 호스트로 전송되어진다. 패킷은 무선 링크에서 손실을 입기 쉬운데, 이러한 손실들은 정상상태를 의미하는 good 상태와 핸드오프 상황을 의미하는 bad 상태의 Two-state markov model을 사용할 때 결정되어진다. 이때 패킷 여러들과 상태-전이 값들이 랜덤 분포를 따르지 않고 일정하다고 가정되어지는 간단한 모델을 사용한다. 시뮬레이션은 60초 동안 4초기간의 핸드오프가 4번 발생하게 하였다. 10초 동안 정상적인 망 상태를 의미하는 good 상태에 있고, 그 다음에 핸드오프를 의미하는 bad 상태에 들어간다. bad 상태에는 4초 동안 남아있고 그 다음에 good 상태로 재 진입한다. 이 사이클이 연결의 기간동안 계속된다. 그리고 무선 구간에서 핸드오프의 영향을 알아보기 위해서 유선 네트워크에서는 혼잡으로 인한 손실이 없다고 가정한다. 비교하는 알고리즘의 성능을 보여주는 파라미터는 시간과 TCP 패킷의 수 그리고 혼잡윈도우의 크기이다. 비교 대상은 빠른 재전송 및 회복을 지원하는 기존의 Tahoe TCP 버전의 알고리즘과 무선망에서 핸드오프를 고려해서 Tahoe TCP의 알고리즘을 수정한 EHN 알고리즘이다.

4.2 제안된 알고리즘의 비교분석

4.2.1 Congestion Window 비교

[그림 2]는 핸드오프가 없는 상황에서 기존의 Tahoe TCP와 핸드오프가 있는 상황에서 Tahoe TCP 그리고 제안된 EHN 알고리즘에서 혼잡 윈도우 크기를 비교하였다. 핸드오프가 없는 상황에서 Tahoe TCP는 cwnd가 계속해서 증가함을 보이고 있으나, 핸드오프가 있는 상황에서 Tahoe TCP는 핸드오프가 발생될 때마다 송신자(고정 호스트)는 이미 보내진 패킷의 확인응답을 기다리는 동안 타임아웃 될 것이고 반복된 Slow Start 절차를 수행하여 cwnd 크기가 증가하지 못함을 보이고 있다. 반면에, 제안된 EHN 알고리즘에서는 송신자가 persist mode가 되어 재전송 타이머가 멈추게 되기 때문에 타임아웃이 일어나지 않게 된다. 즉, 핸드오프 기간동안 소스에서 발생하는 타임아웃의 가능성을 완전히 제거한다. 그리고, 핸드오프가 발생되었을 때 혼잡 제어 절차를 수행하지 않음으로, cwnd의 크기는 이전 상태를 그대로 유지하여 핸드오프 이후의 데이터 전송에 대비할 수 있다. 핸드오프가 있을 때, Tahoe TCP는 그림에서 보면 핸드오프가 발생하는 10초 ~ 14초의 구간에서 13초 지점에서, 24초 ~ 28초의 구간에서 26초 지점에서, 38초 ~ 42초의 구간에서 40초 지점에서, 52초 ~ 56초의 구간에서 54초 지점에서 Slow Start 절차가 수행되어서 cwnd의 크기가 "1"에서 다시 증가하는 것을 알 수 있다.

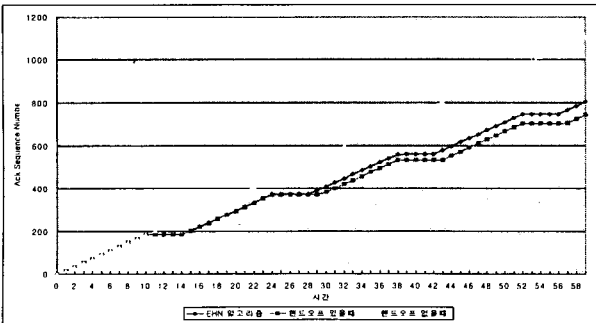


그림 2 : Congestion Window 비교

4.2.2 Throughput 비교

[그림 3]은 고정 호스트가 기지국을 통해 전송한 패킷에 대해 이동 호스트가 고정 호스트로 전송한 Ack의 수를 보여주고 있다. 핸드오프가 있을 때, Tahoe TCP는 타임아웃의 발생으로 인해 초당 Ack의 수가 cwnd 감소의 영향으로 간헐적으로 느리게 증가하나 제안된 EHN 알고리즘에서는 핸드오프의 발생 시에만 전송이 중단되고 핸드오프가 완료된 이후에는 고정 호스트에서 혼잡 제어 절차를 호출하지 않고 이전의 cwnd를 그대로 사용함으로써 핸드오프 이전에 사용하던 대역폭으로 데이터를 전송하고 있음을 알 수 있다. Tahoe TCP의 Throughput은 12.383 이고, 제안된 EHN의 Throughput은 13.383으로 EHN 알고리즘이 Tahoe TCP의 성능에 비해 약간 좋다는 것을 알 수 있으나 핸드오프 기간과 핸드오프의 횟수가 증가함에 따라 그 성능의 차이는 더 증가함을 예상할 수 있다.

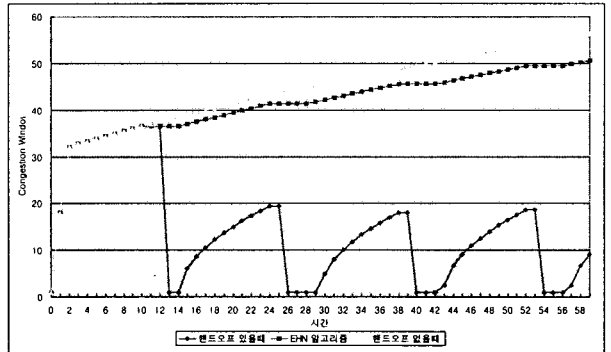


그림 3 : Throughput 비교

5. 결론

본 논문에서는 무선 구간에서 이동 호스트의 핸드오프로 인한 TCP의 성능 저하를 방지하기 위해서 무선망에서 핸드오프가 발생할 때, 핸드오프를 가장먼저 알게되는 이동 호스트가 ICMP 형태의 EHN 패킷을 고정 호스트에 전송하여 명시적으로 새로운 기지국과 고정 호스트에 알림으로써 핸드오프에 좀 더 적극적으로 대처할 수 있게 하여 핸드오프 과정 동안에 무선구간에서 발생하는 패킷 손실로 인한 고정 호스트의 재전송 타이머의 타임아웃과 불필요한 혼잡제어 수행에 따른 성능감소를 미연에 방지할 수 있는 EHN 알고리즘을 제안하였다. 이 알고리즘은 핸드오프 이전에 사용하던 대역폭을 핸드오프 이후에도 그대로 다시 사용할 수 있었고, 이러한 수행으로 인해 망 혼잡과 핸드오프를 쉽게 식별하여 대처할 수 있었다. 향후 과제로는 시뮬레이션을 통한 다양한 분석내용의 검증과 실제 네트워크에서 동작시킬 때 나타나는 효과를 알아보기 위해 실제 유/무선망이 통합된 망에 적용해 보는 것이고, 무선망과 ATM 망과 같은 초고속 네트워크가 연동 될 경우에 있어서 나타나는 성능의 효과를 분석하고 적용하는 것이다.

[참고문헌]

- [1] Aldar C. F. Chan, Danny H. K. Tsang, and Sanjay Gupta, "Impacts of Handoff on TCP Performance in Mobile Wireless Computing". <http://www.ee.ust.hk/~ustatm/conference.html>.
- [2] V. Jacobson. Congestion avoidance and control. In Proceedings of the ACM SIGCOMM '88, pages 314-329, August 1998.
- [3] R. Caceres and L. Iftode, "Improving the performance of reliable transport protocols in mobile computing environments," IEEE Journal on Selected Areas in Communications, vol. 13, no. 5, June. 1995.
- [4] 김효선, 유 혁, "무선망에서의 개선된 TCP 프로토콜에 대한 고찰," 한국 정보 과학회 논문집, 제25권 제2호, 1998.