

Mobile IP 핸드오프를 위한 효율적인 TCP 방식

(An Efficient TCP Mechanism for Mobile IP Handoffs)

권재우[†] 박희동^{**} 조유제^{***}

(Jae-Woo Kwon) (Hee-Dong Park) (You-Ze Cho)

요약 기존 TCP가 이동 환경에 그대로 적용될 경우, 빈번하게 발생하는 핸드오프에 의한 패킷 손실에 대해 불필요한 폭주 제어 알고리즘을 수행함으로써 TCP의 성능 저하를 초래한다. 본 논문에서는 이동 호스트의 핸드오프 발생 사실을 최대한 빨리 감지하기 위한 TCP-MD 방식과 핸드오프 시 등록 과정동안 송신측에서 전송을 중지하여 패킷 손실을 최소화하는 TCP-R 방식을 제안한다. 제안된 방식은 종단간 TCP 연결을 유지하며 이로 인해 기존망과 호환이 가능하도록 한다. 또한, 유선 구간에서는 기존 TCP의 수정 없이 사용하고, 핸드오프 발생 사실을 알리기 위해 별도의 메시지 정의 없이, 단지 이동 호스트 내 TCP에서 Mobile IP 핸드오프 시 등록 관련 메시지를 이용함으로써 TCP 성능을 향상시킬 수 있고 구현이 단순하다는 장점이 있다. 다양한 환경에 대한 시뮬레이션을 수행하여 각 제안된 방식들을 적용할 경우 TCP의 성능을 크게 향상시킬 수 있음을 보였다.

키워드 : Mobile IP, 핸드오프, 폭주제어, TCP-MD, TCP-R

Abstract When using TCP over a mobile network, TCP responds to a handoff by invoking a congestion control algorithm, thereby resulting in a degraded end-to-end performance in a mobile network. In this paper, two schemes are proposed, TCP-MD and TCP-R. TCP-MD can detect the movement of a mobile host early on, whereas TCP-R can force the source to freeze data transmission during registration. The proposed schemes maintain end-to-end TCP semantics, making it possible to fully interoperate with the existing infrastructure. Only a small change is required in the mobile host, plus the implementation is simple because some Mobile IP messages are used to notify the handoff, eliminating the need for any additional messages. Simulations confirmed that the proposed schemes give an excellent performance under various environments.

Key words : Mobile IP, handoffs, congestion control, TCP-MD, TCP-R

1. 서론

인터넷 상에서 단말의 이동성 보장을 위해 Mobile IP 프로토콜이 제안되고 있으며, 상위 TCP 프로토콜이 무선 및 이동 환경에 그대로 적용이 가능한지에 대한 연구가 활발히 진행 중에 있다. TCP는 현재 인터넷에서 사용되고 있는 전송 프로토콜로서 종단간 신뢰성 있는 데이터의 전송 및 바이트 스트림의 연결성 서비스를 제공

한다. TCP는 기본적으로 오류율이 낮은 유선 링크로 구성된 통신망에서 잘 적용할 수 있도록 설계되었으므로, 데이터 전송 시에 발생하는 모든 패킷 손실은 통신망의 폭주에 의해 발생한 것으로 간주하여 동적으로 윈도우의 크기를 조절하는 폭주 제어를 수행한다[1]. 따라서 유무선 통합망에서 기존 TCP를 적용한다면 무선 링크의 오류 및 단말의 이동에 의한 패킷 손실까지도 통신망의 폭주로 간주하기 때문에 상당한 성능 저하를 초래할 수 있다.

무선 및 이동 환경에서 TCP 성능 개선을 위해 지금까지 제안된 방식에 대한 문제점에 대해 살펴보면 다음과 같다. 연결 분리 방식은 종단간 TCP 연결이 파괴되는 단점이 있으며 이동성 지원 관점에서 핸드오프 동안 이전 기지국에서 유무선 링크 간의 연결에 관한 상태 정보를 새로운 기지국으로 전달해야 하므로 핸드오프의 지연이 상태 정보의 크기에 비례하여 증가하게 된다[2].

· 본 연구는 한국과학재단의 목적기초연구(R01-1999-00239) 지원으로 수행되었음

† 비회원 : 삼성전자 연구원

kwonjaewoo@orgio.net

** 비회원 : 포항1대학 컴퓨터응용학과 교수

hdpark@pohang.ac.kr

*** 종신회원 : 경북대학교 전자전기컴퓨터학부 교수

yzcho@ee.knu.ac.kr

논문접수 : 2001년 4월 25일

심사완료 : 2002년 5월 31일

기지국에서의 재전송 방식은 이동성 지원보다는 무선 링크의 오류에 대한 보상에 초점을 맞추고 있다[3]. 종단간 핸드오프 지원 방식으로 Caceres가 제안한 빠른 재전송 방식은 핸드오프 발생 시 짧은 연결 중단 시간을 가진다고 가정하고 있기 때문에 긴 연결 중단 시간을 가진다면 성능 향상을 기대할 수가 없다[4]. 이를 개선하기 위해 핸드오프 후 폭주 윈도우가 축소되지 않도록 TCP의 persist 모드를 이용하는 방법이 새롭게 제안되었으나 연결 분리 방식에 기반하고 있다는 단점을 가진다[5].

본 논문에서는 빠른 재전송 방식 및 TCP persist 모드를 응용하였으며, 종단간 TCP 연결을 유지하면서 유선 구간에서는 기존 TCP를 수정 없이 사용하고, 단지 이동 호스트 내 TCP에서 Mobile IP 핸드오프 시 등록 관련 메시지 처리 과정을 추가하여 TCP 성능을 향상시키는 방안을 제안하였다. 이를 위해 본 논문에서는 이동 호스트의 핸드오프 발생 사실을 최대한 빨리 감지하기 위한 알고리즘인 TCP-MD 방식과 핸드오프 발생 사실을 통보 받으면 송신측에서 전송을 중지하여 패킷 손실을 최소화하는 알고리즘인 TCP-R 방식을 제안하고 있다. 또한 각 제안된 방식을 따로 적용하기보다는 각 방식들을 통합(TCP-MD&R)함으로써 여러 다른 환경에 대해 큰 성능 향상을 기대할 수 있다. 본 방식의 성능 평가를 위해 VINT 프로젝트로 개발된 통신망 시뮬레이터인 NS-2를 사용하여 제안된 방식과 기존 방식 간의 성능을 비교 분석한다.

본 논문의 구성은 다음과 같다. 서론에 이어 2장에서는 이동 환경에서 TCP의 적용 시 문제점에 대해 살펴보고, 유무선망에서 TCP 성능을 향상시키기 위해 지금까지 제안된 방식들의 문제점에 대해 살펴본다. 3장에서는 본 논문에서 제안하는 TCP-MD 방식과 TCP-R 방식에 대해 설명한다. 그리고, 4장에서는 시뮬레이션을 통해 제안된 알고리즘의 성능을 분석한다. 마지막으로 5장에서는 결론을 맺는다.

2. 기존 관련 연구 고찰

Mobile IP에서 이동 호스트(mobile host: MH)는 두 개의 IP 주소를 가진다[6]. 한 주소는 홈 주소로 일반 IP 주소와 마찬가지로 고정된 주소이며, 다른 한 주소는 COA(Care-Of Address)로 새로운 연결 지점마다 변경되는 주소이다. Mobile IP를 지원하기 위해 각 지역 네트워크에는 이동성을 지원하는 이동성 에이전트(mobility agent)가 있고, 이동 호스트의 현재 위치를 나타내는 COA를 이용한 터널링을 통해 패킷 전달이 이루어진다.

이동성 에이전트는 홈 에이전트(home agent: HA)의 역할과 외부 에이전트(foreign agent: FA)의 역할로 나누어지며, 등록 과정을 통해 이동 호스트에 대한 바인딩을 관리하고 홈 주소와 COA사이의 주소 변환 기능을 수행한다. 또한 주기적으로 에이전트 광고 메시지(Agent Advertisement)를 방송하여 이동 호스트에게 현재 연결된 네트워크가 홈 네트워크인지 외부 네트워크인지 구별할 수 있게 한다.

2.1 Mobile IP에서의 핸드오프 과정

이동 호스트가 FA간의 셀 사이로 핸드오프 시 이동을 감지하고 등록 과정을 수행하는 메커니즘이 요구된다.

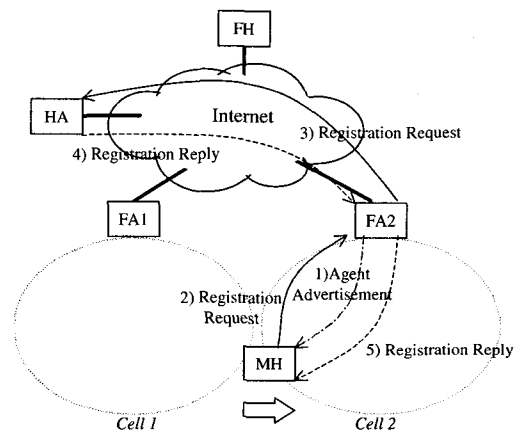


그림 1 Mobile IP에서의 핸드오프 절차

그림 1에서는 Mobile IP 핸드오프 과정을 나타내고 있다. FA는 주기적으로 Agent Advertisement 메시지를 방송함으로써 자기의 존재를 셀 내의 모든 이동 호스트에게 알리고, 이동 호스트는 일정 시간동안 현재의 FA로부터 Agent Advertisement 메시지를 수신하지 못한 상태에서 새로운 FA의 Agent Advertisement를 수신한 경우에 새로운 셀로의 핸드오프를 시작한다. 만약 이동 호스트가 Agent Advertisement 메시지를 받지 못한다면 Agent Solicitation 메시지를 전송함으로써 해당하는 이동 에이전트로부터 Agent Advertisement 메시지를 받을 수 있게 된다. 새로운 Agent Advertisement 메시지를 수신하면 등록 과정을 수행하기 위해서 이동 호스트는 새로운 FA(FA2)에게로 자신의 IP주소와 자신의 HA 주소 및 COA를 포함하고 있는 Registration Request 메시지를 전송한다. 새로운 FA는 Registration Request 메시지를 HA로 포워딩하고, HA는 해당 이동 호스트에게로 Registration Reply를 전송하여 이동 호

스트 패킷의 라우팅을 시작한다.

2.2 이동 환경에서 TCP 적용 시 문제점

핸드오프가 발생했을 때 이동 호스트는 이동을 감지하고 등록 과정을 수행해야 하므로 이 과정동안 패킷을 송수신할 수 없는 연결 중단 시간이 발생하게 된다. 특히 현재 구현된 Mobile IP에서는 이러한 연결 중단 시간이 수 초에 이르기 때문에 송신측 TCP에서는 타임아웃이 exponential backoff 방식으로 크게 증가하므로 이러한 긴 타임아웃 시간동안 송신측 TCP는 패킷 전송을 중지하게 된다[7]. 이 경우 송신측 TCP에서는 폭주와 무관하게 핸드오프에 의한 패킷 손실까지도 폭주 제어 알고리즘을 수행하게 되므로 상당한 성능 저하를 초래한다. 특히 타임아웃이 여러 차례 발생할 경우 폭주 윈도우 관련 파라미터 값이 너무 작게 되어 비록 연결이 다시 이루어져도 정상 상태로 다시 접어드는데 긴 시간이 요구된다. 또한 현재 인터넷의 사실상 표준인 TCP Reno 방식은 여러 개의 패킷 손실에 적절히 대처하지 못하는 문제점을 가지고 있다[8].

2.3 무선망을 위한 기존 TCP 관련 연구 고찰

현재 많은 연구 결과들이 유무선 복합망 환경에서 TCP 성능을 저하시키는 원인을 분석하여 이에 적절히 대응할 수 있는 방법들을 제시하고 있다. 제안된 방법들의 공통적인 특징은 가능한 한 연결 경로에 있는 무선 링크에서의 오류가 유선망에 영향을 미치지 않도록 하여 유선 링크에서의 오류에 의해서만 폭주 제어 알고리즘이 호출되도록 하는 것이다.

I-TCP와 같은 연결 분리 방식은 유무선 구간을 구분하여 각각 독립된 연결을 가지도록 함으로써 각 링크를 효율적으로 이용할 수 있기 위한 방안이다. 연결을 무선 링크 연결과 유선 링크 연결로 나누는 방식의 이점은 이동성 및 무선 링크의 불안정과 관련된 문제들이 무선 링크 내에서만 처리가 된다는 점이다. 그러나 이동성 지원 관점에서 핸드오프 동안 이전 FA에서 유무선 링크 간의 연결에 관한 상태 정보를 새로운 FA로 전달해야 하므로 핸드오프의 지연이 상태 정보의 크기에 비례하여 증가하게 된다. 또한, 모든 패킷이 기지국에서 무선측의 연결과 유선측의 연결을 거쳐야 하므로 두 배의 오버헤드를 초래한다.

Snoop 방식과 같은 FA에서의 재전송 방식은 기존의 TCP를 수정하지 않고 종단간 TCP 연결을 유지하기 위해 단지 FA에서 데이터 버퍼링과 지역 재전송을 수행한다. 이 방식의 기본 목적은 기존의 TCP에 대한 수정 없이 성능 향상을 추구한다는 것이다. 그러나 이 방식은 이동성 지원보다는 무선 링크의 오류에 대한 보상

에 초점을 맞추고 있다. 따라서 현재 이동성 지원을 위해 가장 적합한 방식은 종단간 핸드오프 지원 방식이라 할 수 있다.

현재 이러한 방식으로 Caceres가 제안한 빠른 재전송(Fast Retransmit) 방식이 있는데 TCP의 폭주 제어 알고리즘의 빠른 재전송 방식과 거의 유사한 기능을 수행하도록 제안되었다. 그러나 이 방식은 핸드오프 발생 시 짧은 연결 중단 시간을 가진다고 가정하고 있다. 즉, 짧은 연결 중단 시간동안 한 두개의 패킷이 손실된다면 제안된 빠른 재전송 방식에 의해 잘 동작할 수 있지만, 긴 연결 중단 시간을 가진다면 송신단은 벌써 타임아웃이 발생해 폭주 윈도우 크기를 계산하기 위한 파라미터 값(폭주 윈도우 크기, 슬로 스타트 임계값)이 줄어든 상태이므로, 이러한 상황에서 빠른 재전송 알고리즘을 실행할지라도 성능 향상을 기대할 수가 없다. 이를 개선하기 위해 핸드오프 후 폭주 윈도우가 축소되지 않도록 TCP의 persist 모드를 이용하는 M-TCP 방식이 제안되었다. 그러나 이 방식은 종단간 연결을 중간 노드에서 분리하므로 앞서 살펴본 연결 분리 방식의 문제점들을 가지게 된다.

3. 제안된 알고리즘

Mobile IP 핸드오프 시 패킷 손실이 발생하는 구간은 이동 감지 과정과 등록 과정이며, 이러한 손실 발생을 최소화하기 위해 본 논문에서는 이동 호스트의 핸드오프 발생 사실을 최대한 빨리 감지하기 위한 알고리즘인 TCP-MD(Movement Detection) 방식과 핸드오프를 감지하게 되면 등록 과정동안 송신측으로 전송 중지를 요청하게 하여 패킷 손실을 최소화하는 알고리즘인 TCP-R(Registration) 방식을 제안한다.

3.1 TCP-MD

Mobile IP에서는 망 내의 부하를 고려해서 이동 감지를 위해 사용하는 Agent Advertisement 메시지의 방송 주기를 최소 1초로 권고하고 있다. 그러나 방송 주기의 granularity가 크기 때문에 이동 호스트의 이동을 감지하는데 걸리는 시간이 너무 길어질 수 밖에 없다. 제안된 TCP-MD 방식은 이러한 이동 감지 시간을 줄이기 위해 상위 TCP 계층에서 일정 시간동안 패킷의 수신 없이 있을 경우, 핸드오프 발생 가능성을 Mobile IP로 알려주는 메커니즘을 추가하여 가능한 한 빨리 이동을 감지하도록 하고 있다.

그림 2에서는 제안된 이동 감지 방식인 TCP-MD를 이용한 Mobile IP 동작 절차를 나타내고 있으며, 고정 호스트에서 이동 호스트로의 패킷 전송만을 가정할 때

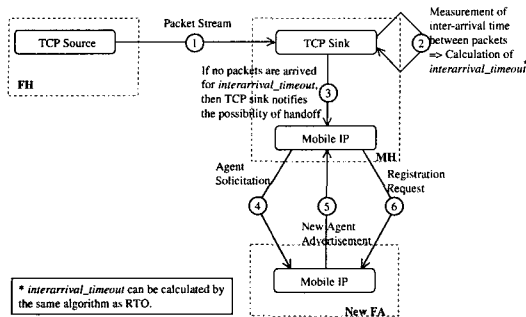


그림 2 TCP-MD를 이용한 Mobile IP 동작 절차

이동 호스트에서 이동을 감지하기 위한 동작 절차를 설명하면 다음과 같다.

- ① 연결이 이루어진 상태에서 송신측으로부터 패킷을 계속 수신하고 있다.
- ② 수신측 TCP에서 수신되는 패킷들 간의 도착 간격을 측정하여 다음 패킷이 정상적으로 도착해야 할 최대 예상 도달 시간인 *interarrival_timeout*을 계산한다. 이때, *interarrival_timeout*값 계산 알고리즘은 재전송 타이머의 타임아웃(RTO) 계산 알고리즘을 그대로 적용한다.
- ③ 만약 계산된 *interarrival_timeout* 동안 패킷의 수신 없이 있다면, Mobile IP로 핸드오프의 발생 가능성을 알리고 타이머를 리셋시킨다.

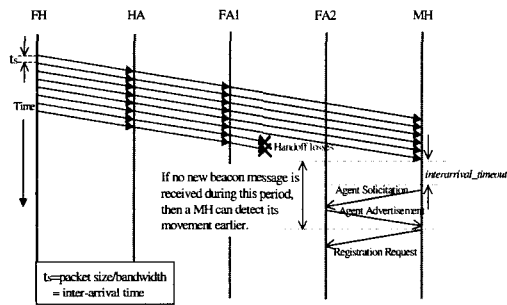


그림 3 TCP-MD 방식의 타이밍 다이어그램

그림 3에서는 TCP-MD 방식에 대한 타이밍 다이어그램을 나타내고 있다. TCP-MD 방식은 수신측에서 수신되는 패킷의 도착 간격을 측정하여 다음 패킷의 최대 예상 도달 시간인 *interarrival_timeout*값을 계산하는 과정을 추가하고 있으며, 보다 빨리 핸드오프 발생 가능성을 감지함으로써 송신측에서 더 이상의 타임아웃 발생을 최소화할 수 있다. *interarrival_timeout* 계산 알

고리즘은 아래와 같이 기존 TCP에서의 재전송 타임아웃(RTO)의 계산 알고리즘과 동일하게 적용하고 있다[9].

$$diff = current_time - average$$

$$average = average + \alpha * diff$$

$$deviation = deviation + \beta * (|diff| - deviation)$$

$$interarrival_timeout = average + 4 * deviation$$

여기에서 *current_time*은 현재 수신된 패킷의 도착 간격 시간이며, *average*는 이전 패킷들의 도착 간격에 대한 평균값을 나타낸다. 또한 *diff*는 현재 패킷의 도착 간격과 이전 패킷들의 평균값과의 차이를 나타내며, *deviation*은 이에 대한 편차값을 나타낸다. 이 때, IETF의 RFC의 권고값대로 α 는 1/8, β 는 1/4로 설정한다. 다음 패킷의 도착 간격에 대한 예측치인 *interarrival_timeout*은 수신되는 패킷들의 도착 간격의 평균값과 편차값에 의해 결정되며, 편차값의 반영에 의해 망의 상황에 따라 동적으로 타임아웃 값이 적절히 변화할 수 있다. 즉, 망이 폭주 상황에 가까워질수록 편차값이 커지므로 보다 큰 타임아웃 값을 가지게 되며, 이는 핸드오프 외의 영향에 의해 타임아웃이 발생할 가능성이 작을 수 있다.

TCP-MD에서는 *interarrival_timeout* 동안 패킷의 수신 없이 있을 경우 핸드오프가 발생한 것으로 간주함으로써 기존의 방법에 비해 신속한 핸드오프 처리가 가능한데, 이는 다음과 같은 가정을 기반으로 하고 있다.

- 첫째, 송신측에서는 대량의 데이터를 전송하고 있다.
- 둘째, 신뢰성 있는 링크 계층 프로토콜을 사용함으로써 무선링크 상의 오류는 무시될 수 있다.

만약, 무선링크 오류가 어느 정도 존재할 경우, 불필요한 Agent Solicitation 메시지를 전송할 수도 있으나, 이는 MH와 FA간의 무선링크 내에 국한되는 것으로 전체적인 TCP 성능에는 크게 영향을 끼치지 않는다. 다만, 불필요한 Agent solicitation 메시지 전송을 최소화하기 위해 별도의 타이머를 MH의 Mobile IP 계층에 도입으로써 이를 제한할 수 있다. 즉, MH의 TCP-MD가 Mobile IP 계층에 핸드오프 가능성을 알리더라도 타이머에 의해 너무 빈번히 Agent solicitation 메시지가 전송되는 것을 막을 수 있다.

한편, MH가 데이터를 송신할 경우에는 TCP-MD에 의한 빠른 이동 감지 과정이 필요 없다. 왜냐 하면, MH가 링크 레벨에서의 핸드오프 완료 후, 설정된 전송링크를 통해 바로 데이터를 전송할 수 있기 때문이다. 이것은 MH가 송신자일 경우에는 수신할 때와는 달리 HA에 자신의 위치를 등록할 필요 없이 직접 FH로 데이터를 전송할 수 있기 때문이다.

TCP-MD의 구현과 관련하여 TCP-MD와 Mobile IP 계층간에 정의된 제어정보 및 인터페이스 구성은 그림 4에서와 같다. 즉, TCP-MD의 구현은 기존의 TCP 계층에 Handoff decision/notifying 모듈을 추가한 후, Mobile IP 계층의 Movement detection 모듈에 전달할 제어정보를 정의해 줌으로써 가능하다. 새로이 추가되는 Handoff decision/notifying 모듈은 기존 TCP의 RTO (Retransmission timeout) 계산 모듈과 거의 동일한 것으로서, Input processing 모듈로부터 입력받은 정보를 바탕으로 *interarrival_timeout* 값을 계산하고 핸드오프 발생 여부를 판단하게 된다. 만약 핸드오프 가능성이 있을 경우 Mobile IP 계층의 Movement detection 모듈에 제어정보 즉, *handoff_possible*를 전송한다. 그 이외의 동작은 기존의 TCP 및 Mobile IP와 동일하다.

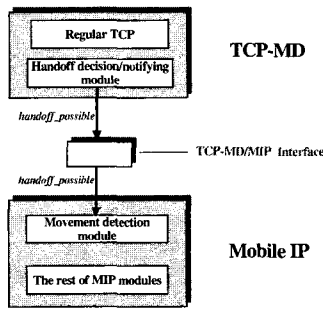


그림 4 TCP-MD와 Mobile IP 계층간 상호작용

3.2 TCP-R

Mobile IP에서는 핸드오프 시 이동 호스트가 FA의 Agent Advertisement 메시지를 수신하게 되면 새로운 서브 네트워크로의 이동을 확인하게 된다. 이와 같이 새로운 FA로의 이동을 감지하면 이동 호스트는 등록 과정을 수행하게 되는데, 제안된 TCP-R 방식은 ACK 메시지의 advertised window를 이용하여 이동 호스트가 새로운 FA와의 등록 과정이 완료될 때까지 송신측이 전송을 중지하도록 하고, 등록이 완료되면 이전 윈도우 크기로 전송을 재개할 수 있는 방식이다. 이 때 TCP 폭주 제어 관련 파라미터 값이 줄어드는 것을 방지함으로써 TCP 성능 향상을 가져올 수 있다.

그림 5에서는 핸드오프 발생 후 등록 과정 시 제안된 방식의 동작 절차를 나타내고 있다. 등록 절차를 설명하면 다음과 같다.

- ① 핸드오프 시 이동 호스트가 새로운 FA의 Agent Advertisement 메시지를 수신한다.
- ②, ②' 이동을 감지하면 새로운 FA로 Registration

Request 메시지를 전송한다. 이 때, MH 내 Mobile IP는 상위 TCP로 이러한 사실을 알려주면서, ACK 내의 advertised window 크기(*win_size*)를 0으로 설정하여 상대방 고정 호스트로 전송하면 상대방 고정 호스트는 TCP persist 모드로 들어가며 전송을 중지하게 된다.

- ③, ③' FA와 HA 간에 Registration Request & Reply 메시지가 교환된 후 이동 호스트가 새로운 FA로부터 Registration Reply 메시지를 수신하면 윈도우 크기를 이전 값(*pre_value*)으로 설정해서 상대방 고정 호스트가 전송을 재개하게 한다.

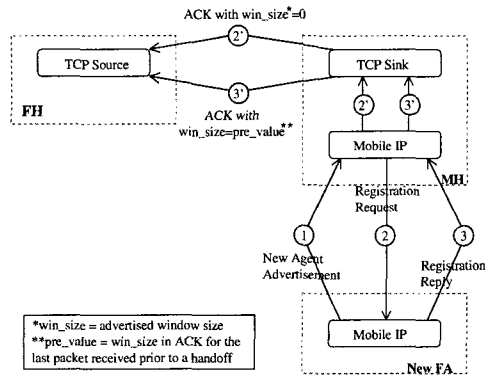


그림 5 등록 과정 시 제안된 TCP-R 방식의 동작 절차

그림 6에서는 Mobile IP 핸드오프 시 등록 메시지 교환 절차 및 TCP-R 방식의 동작 절차를 타이밍 다이어그램으로 나타내고 있다

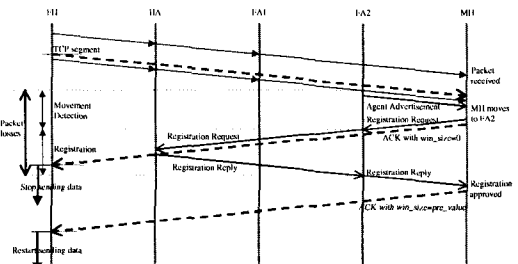


그림 6 핸드오프 시 TCP-R 방식의 타이밍 다이어그램

한편, MH가 데이터를 송신할 경우에 TCP-R은 더욱 간단히 동작할 수 있다. 왜냐하면, MH가 Mobile IP 계층에서 이동을 감지한 후, 자신의 TCP 계층에 바로 이 사실을 알려 TCP 계층이 신속히 data 전송을 중지할 수 있기 때문이다.

TCP-R의 구현과 관련하여 TCP-R과 Mobile IP 계층간에 정의된 제어정보 및 인터페이스 구성은 그림 7에서와 같다. 즉, TCP-R의 구현은 기존의 TCP 계층에 존재하는 Window size control 모듈만을 수정한 후, Mobile IP 계층으로부터 전달받을 제어정보를 정의해 줌으로써 가능하다. 수정된 Window size control 모듈은 Mobile IP의 Movement detection 모듈 및 Registration management 모듈로부터 각각 start_persist 및 stop_persist 제어정보를 수신할 수 있도록 설계되어진다. 즉, Window size control 모듈은 새로운 FA를 발견한 Mobile IP 계층의 Movement detection 모듈로부터 start_persist 제어정보를 전달받을 경우, 윈도우 크기를 0으로 하여 송신 TCP로 하여금 persist 모드로 들어가게 만든다. 또한 위치 등록 과정을 마친 Registration management 모듈로부터 stop_persist 제어정보를 전달받을 경우에는 윈도우 크기를 이전의 값으로 복원하여 송신 TCP에게 전달함으로써 persist 모드로부터 빠져 나오게 한다. 그 이외의 동작은 기존의 TCP 및 Mobile IP와 동일하다.

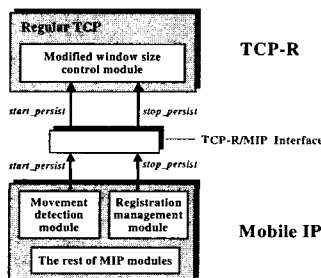


그림 7 TCP-R과 Mobile IP 계층간 상호작용

4. 시뮬레이션을 통한 성능 분석

본 논문에서는 현재 TCP의 사실상 표준인 TCP Reno방식과 네트워크 레벨에서의 이동성 지원을 위한 Mobile IP를 고려하여 시뮬레이션을 수행했다. 핸드오프의 영향만을 고려하기 위해 무선 구간에서의 오류는 신뢰성 있는 링크 계층 프로토콜을 사용하여 무시할 수 있다고 가정한다. 또한 데이터 전송 방향은 고정 호스트에서 이동 호스트로의 FTP에 의한 대량 데이터 전송만을 고려하고 있다. 이를 검증하기 위해 NS-2를 사용하여 시뮬레이션을 수행하였다[10].

4.1 시뮬레이션 모델

그림 8에서는 본 논문에서 고려된 망 구성을 나타낸다. FA 간의 핸드오프를 고려하였으며 각 FA 간의 연

결은 유선 LAN환경을 고려한다. 여기서 FA의 셀 반경은 약 250m이며 무선 링크는 무선 LAN 환경을 고려하고 있다. 성능 분석을 위해 고정 호스트(FH)와 중간 노드(FN) 간의 유선링크

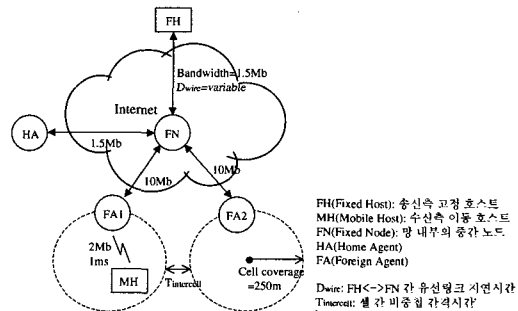


그림 8 시뮬레이션 망 구성

지연시간인 Dwire와 각 FA가 커버하는 셀 간의 중첩되지 않는 시간 간격, 즉 이동 호스트(MH)가 이동시 새로운 외부 에이전트(FA)로부터 beacon 메시지를 수신할 수 있는 시간인 Tintercell 값의 변화에 따라 각 방식들의 성능을 비교한다. 유선망에서는 폭주에 의한 패킷 손실이 없다고 가정하며, 무선 링크 구간에는 신뢰성 있는 링크 계층 프로토콜을 사용하여 비트 오류에 의한 패킷 손실이 없다고 가정한다.

표 1에서는 시뮬레이션에 사용된 각 파라미터 값들을 정의하고 있다.

표 1 시뮬레이션 파라미터

파라미터	값
TCP 패킷 크기	1000 bytes
TCP timer resolution	100 ms
무선링크 대역폭	2 Mbps
무선링크 지연시간	1 ms
LAN 대역폭 (FN<->FA1,2)	10 Mbps
WAN 대역폭 (FN<->FH,HA)	1.5 Mbps
유선링크 지연시간	Variable

4.2 성능 분석

성능 분석은 TCP-MD 방식과 TCP-R 방식으로 나누어 수행하였다. TCP의 성능을 평가하기 위해 수율(throughput)을 비교하였다.

4.2.1 TCP-MD

그림 9에서는 기존 TCP와 제안된 TCP-MD 방식간의 수율을 비교하고 있다. 여기서 송신측인 고정 호스

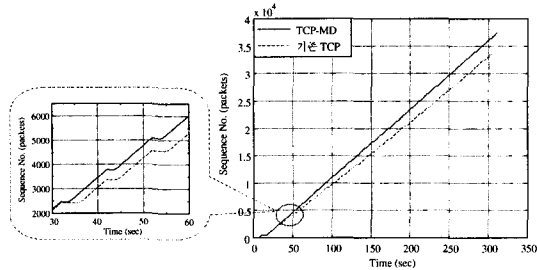


그림 9 기존 TCP와 TCP-MD 간의 수율 비교(Dwire=20ms, Tintercell=0sec)

트와 중간 노드간의 유선링크 지연시간(Dwire)은 20ms이며, 각 FA가 커버하는 셀 간의 비중첩 간격(Tintercell)은 0초로 설정했다. 기존 TCP를 적용했을 경우 송신측 TCP에서는 여러 차례의 타임아웃이 발생하므로 핸드오프 후 폭주 윈도우 크기가 복원되기까지 많은 시간이 요구된다. 제안된 TCP-MD 방식의 경우 이동 감지 시간을 줄일 수 있으므로 여러차례의 타임아웃 발생을 최소화할 수 있다.

표 2 Tintercell과 Dwire값의 변화에 따른 TCP와 TCP-MD 간의 수율 비교

(a) 셀 간 비중첩 간격(Tintercell) : 0 sec

방식 \ Dwire	20ms	50ms	100ms
기존 TCP	71.89 %	62.83 %	51.65 %
TCP-MD	78.20 %	67.66 %	53.36 %

(b) 유선링크 지연시간(Dwire) : 50 ms

방식 \ Tintercell	0 sec	1 sec
기존 TCP	62.83 %	44.40 %
TCP-MD	67.66 %	46.86 %

표 2에서는 유선링크 지연시간의 변화에 따른 기존 TCP와 TCP-MD 방식 간의 수율을 비교하고 있다. 성능 평가를 위해 수신된 패킷의 순서번호를 보고 핸드오프가 발생하지 않는 경우를 기준으로 각 결과를 백분율로 나타내었다. 이 때, 310초의 시뮬레이션 시간동안 매 10초마다 30번의 핸드오프가 발생하도록 하였다. TCP-MD 방식의 경우 이동 감지 시간을 줄일 수 있으므로 송신측 TCP에서의 여러 차례의 타임아웃 발생을 최소화할 수 있다. 따라서 슬로 스타트 임계값을 높게 유지할 수 있으므로 빠른 시간 안에 정상 상태에 접어들 수 있다. 표 2에서 유선링크 지연시간(Dwire) 및 셀 간 비중첩 간격(Tintercell)이 커질수록 제안된 방식의 성능

항상 이득이 작아짐을 확인할 수 있다.

본 논문에서 비교하는 이동 감지 방식인 ECS(Eager Cell Switching) 방식의 경우 평균 이동 감지 시간이 0.5초로 짧기 때문에 FA 간 중첩되지 않는 구간이 커질수록 성능 향상에 한계를 가지게 된다. 즉, 셀 간 비중첩 간격이 1초일 때, 1초 후에 새로운 beacon 메시지를 수신할 수 있으므로 TCP-MD 방식에 의한 빠른 이동 감지에 한계가 있을 수 밖에 없다. 또한 유선링크 지연 시간이 커질수록 더욱 많은 패킷 손실을 초래하므로 제안된 방식에 의한 성능 향상 이득이 작아지게 된다. 이러한 한계를 극복하기 위해 등록 과정에서 새로운 알고리즘인 TCP-R 방식을 제안하고 있다.

4.2.2 TCP-R

각 FA가 커버하는 셀 간의 중첩되지 않는 시간 간격과 송신측 고정 호스트와 중간 노드 간의 유선링크 지연시간의 변화에 따라 기존 TCP와 Caceres가 제안한 빠른 재전송 방식, 그리고 TCP-R 방식과의 성능을 비교한다.

표 3에서는 셀 간의 비중첩 간격과 유선링크 지연시간을 달리하여 기존 TCP와 제안된 TCP-R 방식과의 수율을 비교하고 있다. 성능 평가 기준으로 패킷의 순서번호를 보고 수율을 계산하여 핸드오프가 발생하지 않는 경우를 기준으로 각 결과를 백분율로 나타내었다.

표 3 Tintercell과 Dwire값의 변화에 따른 TCP와 TCP-R 간의 수율 비교

(a) 셀 간 비중첩 간격(Tintercell) : 0 sec

방식 \ Dwire	20ms	50ms	100ms
기존 TCP	71.89 %	62.83 %	51.65 %
TCP-R	80.91 %	72.16 %	60.76 %

(b) 유선링크 지연시간(Dwire) : 50 ms

방식 \ Dwire	0 sec	1 sec
기존 TCP	62.83 %	44.40 %
TCP-R	72.16 %	57.43 %

제안된 방식에서는 핸드오프 발생 시 송신측 고정 호스트가 전송을 중지하고 등록 과정이 완료되면 전송을 재개한다. 이 때 TCP 폭주 제어 관련 파라미터 값이 줄어드는 것을 방지함으로써 TCP 성능 향상을 가져올 수 있다. 표 3에서 알 수 있듯, TCP-R 방식은 유선링크 지연시간 및 셀 간 비중첩 간격에 관계없이 높은 성능 향상 이득을 가진다.

그림 10에서는 셀 간의 비중첩 시간 간격이 0초일 때

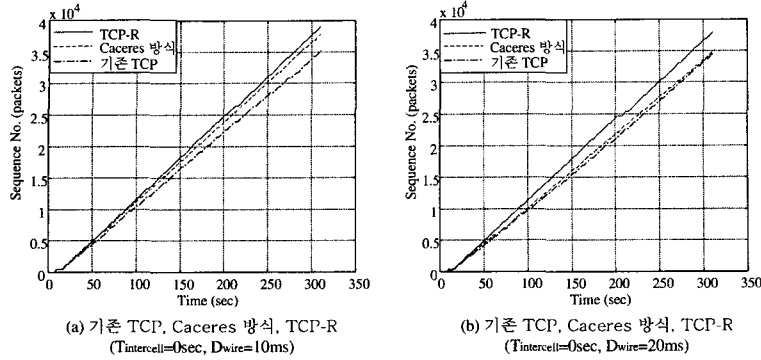


그림 10 Dwire 변화에 따른 TCP, Caceres 방식, TCP-R 간의 수율 비교(Tintercell=0sec)

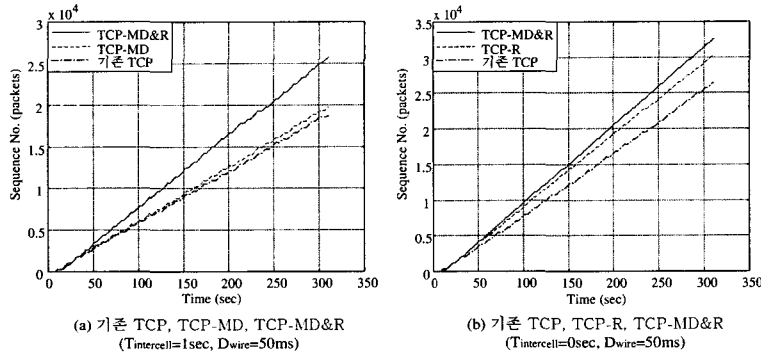


그림 11 각 제안된 방식과 통합된 방식 간의 성능 비교

유선링크 지연시간의 변화에 따른 수율을 비교하고 있다. 유선링크 지연시간이 커짐에 따라 Caceres가 제안한 빠른 재전송 방식의 성능이 현저히 저하됨을 확인할 수 있는데, 그 이유는 Caceres 방식의 경우 핸드오프 시 패킷 손실이 작은 경우만을 가정하고 있기 때문이다. 이에 반해 제안된 방식은 지연시간의 변화에 관계없이 TCP 성능이 향상되었음을 확인할 수 있다.

4.2.3 TCP-MD&R

TCP-MD 방식은 유선링크 지연시간 및 셀 간의 비중첩 간격이 커짐에 따라 성능 향상 이득이 작아진다. 이러한 한계를 해결하기 위해 등록 과정에서 TCP-R 방식을 제안하고 있다.

표 4에서는 각 제안된 방식들을 통합해서 적용했을 경우, 셀 간 비중첩 간격의 변화에 따른 수율을 비교하고 있다.

그림 11에서는 각 제안된 방식들을 통합한 방식과 각 제안된 방식간의 성능을 비교하고 있다. 그림 11(a)에서는 셀 간 비중첩 간격을 1초로 설정하여 TCP-MD 방

표 4 Tintercell 변화에 따른 제안된 방식들의 수율 비교(Dwire=50ms)

방식 \ Tintercell	0 sec	1 sec
기존 TCP	62.83 %	44.40 %
TCP-MD	67.66 %	46.86 %
TCP-R	72.16 %	57.43 %
TCP-MD&R	77.53 %	61.36 %

식의 경우 셀 간 비중첩 간격이 클수록 성능 향상 이득이 작아짐을 보이고 있는데 TCP-R 방식을 추가함으로써 이득을 크게 할 수 있다. 그림 11(b)에서는 셀 간 비중첩 간격을 0초로 설정하여 TCP-MD방식의 경우 보다 빨리 핸드오프 발생을 감지할 수 있으므로 TCP-R 방식과 통합함으로써 성능 향상 이득이 커질 수 있다.

지금까지 살펴본 바와 같이 각 제안된 방식들을 통합함으로써 여러 다른 환경에 대해 성능 향상 이득을 크게 할 수 있다.

5. 결론

본 논문에서는 핸드오프 발생 시 패킷 손실을 최소화하기 위해 이동 감지 과정 및 등록 과정에서 새로운 알고리즘을 제안하였다. 우선 이동 감지 과정에서 제안된 TCP-MD 방식의 경우, 가능한 한 빨리 이동 사실을 감지하여 손실을 최소화함으로써 핸드오프 후 TCP 폭주 윈도우가 가능한 한 빨리 복원될 수 있도록 한다. 등록 과정에서 제안된 TCP-R 방식의 경우, 이동 호스트에서 이동을 감지한 후 등록 과정을 거치는 동안 송신측 TCP의 패킷 전송을 중지시킴으로써 더 이상의 패킷 손실을 방지한다. 또한 폭주 윈도우 관련 파라미터 값을 그대로 유지할 수 있으므로 그에 따른 성능 향상을 가져올 수 있다. 무엇보다 제안된 방식들은 Mobile IP에서 사용되고 있는 등록 관련 메시지를 이용하므로 구현이 간단하다.

본 논문에서는 각 제안된 방식들에 대해 유선링크 지연시간 및 셀 간 비중첩 간격의 변화에 따른 성능 비교를 수행하였으며, TCP-MD 방식의 경우 두 파라미터 값이 작을수록 큰 성능 향상 이득을 가질 수 있었다. TCP-R 방식은 Caceres가 제안한 빠른 재전송 방식과 성능 비교를 수행하였으며, Caceres가 제안한 방식의 경우 유선링크의 지연시간이 커짐에 따라 성능 저하를 가져온 반면 제안된 방식은 지연시간의 변화에 관계없이 TCP 성능이 향상됨을 확인할 수 있었다. 특히 유선링크 지연시간 및 셀 간 비중첩 간격에 상관없이 성능 향상 이득을 크게 할 수 있었다. 무엇보다 이동 감지 과정과 등록 과정에서 제안된 방식들을 따로 적용하는 것보다 각 제안된 방식들을 통합함으로써 여러 다른 환경에서도 성능 향상 이득을 크게 할 수 있었다.

참고 문헌

- [1] M. Allman, V. Paxson, W. Stevens, "TCP Congestion Control," *RFC 2581*, April 1999.
- [2] Ajay Bakre, B. R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts," *Proc. 15th Intl Conf. on Distributed Computing Systems(ICDCS)*, pp. 136-143, May 1995.
- [3] Hari Balakrishnan, V. N. 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, pp. 756-769, Dec. 1997.
- [4] R. Caceres, L. Iftode, "Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments," *IEEE Journal on Selected Areas in Communications*, Vol. 13(5), pp. 850-857, June 1995.
- [5] Kevin Brown, Suresh Singh, "M-TCP: TCP for Mobile Cellular Networks," *ACM SIGCOMM Computer Communication Review*, Vol. 27, No. 5, October 1997.
- [6] Charles Perkins, "IP Mobility Support," *RFC 2002*, October 1996.
- [7] N. A. Fikouras, K. El Malki, S. R. Cvetkovic, C. Smythe, "Performance of TCP and UDP during Mobile IP Handoffs in Single-Agent Subnetworks," *IEEE Wireless Communications and Networking Conference(WCNC)*, Vol. 3, pp. 1258-1262, 1999.
- [8] Fall, K., and Floyd, S., "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP," *Computer Communication Review*, Vol. 26, No. 3, pp. 5-21, July 1996.
- [9] W. Richard Stevens, *TCP/IP Illustrated, Volume 1*, Addison-Wesley, 1994.
- [10] NS-2 Homepage, <http://www.isi.edu/nsnam/ns/>.



권재우

1999년 2월 경북대학교 전자공학과 졸업(공학사). 2001년 2월 경북대학교 전자공학과 졸업(공학석사). 2001년 3월 ~ 현재 삼성전자 연구원. 관심분야는 이동컴퓨팅, 차세대 이동통신망, 차세대 인터넷 프로토콜



박희동

1993년 2월 경북대학교 전자공학과 졸업(공학사). 1998년 2월 경북대학교 전자공학과 졸업(공학석사). 2001년 2월 경북대학교 전자공학과 박사과정 수료. 1998년 2월 ~ 현재 포항1대학 컴퓨터응용학과 전임강사. 관심분야는 이동컴퓨팅, 차세대 이동통신망, 차세대 인터넷 프로토콜

조유제

정보과학회논문지 : 정보통신
제 29 권 제 4 호 참조