

에러 보고를 통한 무선 TCP의 성능 향상

준회원 김 경 희*, 정회원 김 낙 명*

An Enhanced Wireless TCP protocol based on Explicit Error Notification

Kyung-Hee Kim* and Nak-Myeong Kim* *Regular Members*

요 약

인터넷상에서 광범위하게 사용되고 있는 TCP는 통신망에서 패킷 유실이 생길 경우 이물 망 혼잡에 의한 유실이라고 보고 혼잡제어를 통해 혼잡을 해결하고자 한다. TCP는 상대적으로 매우 안정적인 유선링크를 염두에 두고 개발된 전송규약이기 때문에 높은 BER 특성을 가지는 무선링크에 적용시 성능의 하락이 두드러지게 된다. 본 논문에서는 유무선이 통합된 망에서, 특히 전송오류율이 높은 환경에서의 TCP 성능저하를 극복하기 위한 방안으로 Explicit Error Notification(EEN) 알고리즘을 제안하였다. EEN 알고리즘은 무선 링크에서 에러가 발생할 경우 수신측에서 TCP 세그먼트의 시퀀스 넘버와 송수신 주소를 디코딩한다. 이를 통해 송신 TCP에게 무선링크에서 에러로 인한 패킷 손실이 발생했음을 알려주는 NACK를 보내 패킷의 손실이 유선의 혼잡에 의한 것이 아님을 알게 하여 무선링크에서 TCP 프로토콜의 성능을 향상시켰다. 제안한 EEN 알고리즘의 성능을 분석한 결과 에러를 많이 정정할수록 성능향상의 폭이 클 수 있었다.

ABSTRACT

When a packet loss occurs in a communication network operating a TCP protocol, the TCP protocol regards it that the loss has resulted from network congestion. Then the TCP protocol performs congestion control. When it is applied to the wireless network having quite a high BER characteristics, the performance of TCP protocol is degraded very much. In this paper, we propose an Explicit Error Notification(EEN) algorithm to improve the performance of the wireless TCP. When a packet loss occurs in the wireless network, the TCP receiver decodes the TCP segment sequence number and the address of the TCP sender and receiver, and then informs the TCP sender of the error in wireless network by sending a NACK. It is to distinguish packets in error from losses of network congestion. In this paper, the performance of the proposed EEN algorithm is analyzed and simulated. In fact, as more errors are corrected, the proposed algorithm shows a larger improvements in performance.

1. 서 론

최근 PCS폰, 셀룰러폰 등 휴대 전화기의 급속한 보급 및 발달, 노트북 컴퓨터의 보급, Personal Digital Assistant(PDA) 실용화 등이 이어지면서, 무

선으로 인터넷에 접속하고자 하는 욕구가 증대되고 있다. 무선 네트워크는 실내 infrared 네트워크와 빌딩 범위의 무선 랜, 캠퍼스 지역 패킷 무선 통신망, 대도시의 셀룰러 무선 네트워크 등을 포함한다. 또한 4세대 이동통신을 위한 기간망 구조로는 All-IP

* 이화여자대학교 정보통신학과 이동통신 연구실(guesswho@mm.ewha.ac.kr)

논문번호 : 010041-0320, 접수일자 : 2001년 3월 20일

※ 이 연구는 산업자원부가 주관하는 차세대신기술사업인 차세대 무선통신용 트랜시버시스템 개발의 연구비 지원에 의한 것이며 이에 감사드립니다.

에 근거한 통신망이 가장 적합한 것으로 떠오르고 있다. 다만, 현재 인터넷 상에서 광범위하게 쓰이고 있는 전송규약인 Transmission Control Protocol (TCP)는 무선환경에 비해 상대적으로 매우 안정적인 유선 환경을 염두에 두고 개발된 프로토콜이기 때문에 무선 환경의 특성을 반영하는 것은 매우 중요하다²⁾.

무선망은 경로손실, 페이딩, 노이즈, 간섭 등의 이유로 높은 BER을 갖는다⁹⁾. 무선 채널은 10^{-2} ~ 10^{-4} 의 BER을 가지며¹³⁾, 이러한 높은 BER은 패킷의 크기에 따라 다르지만 일반적으로 패킷 손실이 혼잡에 기인하는 유선환경에 비해 상대적으로 매우 높은 Packet Error Rate(PER)을 초래하게 된다. 따라서 패킷전송의 실패를 가져와 TCP 세그먼트가 제대로 전송이 되지 않거나 ACK가 제대로 전송되지 않는 경우를 유발한다.

또, 이동환경에서는 전파에 대한 장애물이나 음영 지역, 특히 셀룰러 네트워크 환경을 가정해 볼 때 셀 간의 이동인 핸드오프에 의해 순간적으로 연결이 끊어지는 문제가 생길 수 있다¹⁰⁾. 이러한 연결 끊김은 패킷 손실 가능성을 높게 된다.

무선 데이터 통신 환경에서 기존의 TCP를 적용하는 경우 전송 효율의 저하가 발생된다. 이를 위해 제안된 알고리즘은 크게 두 부류로 나눌 수 있다. 첫 번째 접근은 TCP 송신기가 무선 구간이 있음을 알지 못하도록 하는 것이다. 대표적인 연결분할 방식으로 I-TCP 프로토콜이 있다²⁾. 무선링크의 높은 BER을 극복하기 위하여 유선링크와 무선링크 각각에 TCP 연결이 설정되고 이 두 연결을 기지국에서 상호 연결하여 주는 구조를 가지고 있다. 따라서 무선링크의 에러에 상관없이 TCP 송신자는 유선링크가 끝나는 곳까지만 TCP 패킷이 전달되면 ACK를 받게 된다. 그러나 I-TCP 사용의 결과로 TCP ACK가 TCP 송신자와 수신자 종단간으로 사용되지 못하고 대신에 유선링크와 무선 링크 각기 다른 ACK를 사용함으로써 TCP 종단간 의미가 유지되지 못하는 단점이 생기게 된다. Berkeley Snoop module¹¹⁾도 이 범주에 속한다¹¹⁾.

두 번째 접근은 수신기가 네트워크에서 무선구간의 존재를 알고 혼잡으로 인한 손실과 무선 링크로 인한 손실을 구분할 수 있게 하는 것이다. 이 경우에는 무선 링크에서 데이터 손실이 생겨도 혼잡알고리즘을 동작시키지 않는다. 핸드오프로 인한 손실을 극복하고자 하는 알고리즘으로는 빠른 재전송, Mobile TCP 등이 있는데, 이는 핸드오프 문제에만

관련된 것으로 무선 링크의 에러특성에 대한 극복은 하지 못한다는 단점이 있다. 무선 손실로부터 혼잡 손실을 구분해 내는 알고리즘도 [12]에서 제안되었다.

그러나 무선 손실로부터 혼잡 손실을 구분해 내는 알고리즘들은 아이디어만을 기술할 뿐, 구체적인 구현에 대해서는 언급하고 있지 않았다. 본 논문에서는 무선에서 에러가 나면 수신측에서 그것을 감지하여 송신측 TCP로 유선 네트워크상의 혼잡이 아닌 무선에서 에러가 났다는 표시의 NACK를 보내는 방법을 제안한다. 이를 위해 TCP 세그먼트의 시퀀스 숫자와 송수신 주소에 Error Correcting Code(ECC)를 적용하였다. 그리고 이를 Tahoe 알고리즘에 적용하여 성능을 평가하여 그 우수함을 보였다.

본 논문의 구성은 다음과 같다. II장에서는 TCP의 혼잡 제어 메커니즘과 그와 관련된 기존의 알고리즘에 대해 살펴본다. III장에서는 무선 통신 환경에서 TCP 성능 향상을 위해 본 논문에서 제안한 EEN 알고리즘에 대해 기술한다. IV장에서는 제안하는 EEN 알고리즘을 적용한 경우와 적용하지 않은 경우의 성능을 비교 평가한 후, 마지막으로 V장에서는 본 논문의 결론을 맺는다.

II. TCP의 혼잡제어 알고리즘 및 관련 알고리즘

1. 혼잡제어 알고리즘

TCP의 최초의 규격은 RFC793⁵⁾이지만, 과거 수 년동안 Tahoe, Reno, Vegas와 같은 여러 가지 변형들이 발표되었다⁹⁾. 본 논문에서는 Tahoe와 Reno를 중심으로 기술하도록 하겠다.

수신기로부터 바이트 k의 누적 ACK는 k보다 작은 바이트에 대해서 성공적으로 도착했으며, 바이트 k에서 시작되는 세그먼트는 아직 도착하지 않았음을 의미한다. TCP 송신기는 네트워크에서 데이터를 잃어버렸다는 것을 알았을 때 잃어버린 세그먼트를 재전송하여 이를 복구한다. TCP 수신기는 정해진 타임아웃 구간 안에 세그먼트를 위한 positive cumulative ACK를 받지 않은 잃어버린 데이터를 재전송한다.

세그먼트를 잃어버린 후, Retransmit Timeout (RTO) 전에 발견하여 Fast Retransmit하기 위하여 cumulative ACKs의 사본을 만들어 이를 수신 TCP로 전송한다. 사본 ACK를 사용하여 수신기는 세그먼트가 잃어버린 것을 알고, 이를 재전송한다. 네트

워크에서 순서가 바뀌어 도착할 수 있으므로 첫 번째 ACK에 대해서 재전송하지 않고 3번째 ACK까지 기다린다.

TCP는 혼잡을 처리하기 위하여 윈도우 기반의 알고리즘을 사용한다. 여기서 윈도우는 네트워크에서 ACK되지 않고 보내질 수 있는 바이트 수의 추정이다. TCP 송신기는 전송 윈도우가 수신기가 공시한 윈도우 크기를 넘지 않도록 하여 flow control을 한다. 혼잡윈도우를 이용하여 전송되는 데이터의 양을 송신기는 규제하는데 이를 혼잡제어라 한다. 연결이 시작될 때, slow start가 수행된다. 여기서 혼잡 윈도우는 1 세그먼트로 초기화되고 새로운 ACK를 받을 때마다 최대 세그먼트 크기만큼 윈도우를 증가시킨다. 미리 정해진 임계치(sssthresh)에 도달하면 연결은 congestion avoidance 단계로 넘어간다. 즉, 각 stage에서 slow start는 혼잡윈도우가 sssthresh에 도달하기 전까지 수행된다. 혼잡 윈도우는 전송 윈도우가 성공적으로 전송되면 1 세그먼트씩 증가한다. 패킷 손실이 있으면 송신기는 혼잡 윈도우의 크기는 초기화된다. 타임아웃이 일어나면 혼잡윈도우는 1 세그먼트로 정해지고 연결은 다시 slow start를 시작하고 sssthresh는 혼잡윈도우 값의 반으로 정해진다.

3개의 중복 ACK를 받으면 혼잡윈도우를 세그먼트 1로 초기화하지 않고 sssthresh의 반값으로 setting 하는 것을 Fast Recovery라고 하며, Fast Retransmit 알고리즘과 함께 TCP-reno에서 쓰인다.

2. 관련 알고리즘

무선 구간의 에러를 유선 구간의 혼잡과 구별하고자 하는 노력은 [12]등에서와 같이 있어왔다. 본 논문에서 제안하는 알고리즘과 비슷한 알고리즘에 대해 알아보기로 하자. 첫 번째로 Explicit Congestion Notification(ECN)^[3]을 들 수 있겠다. 게이트웨이가 혼잡할 때 패킷 헤더에 네트워크의 혼잡을 나타내는 비트를 정해주고 이를 전송하는 것이다. 비트를 정할 때 Random Early Detection (RED) gateway^[6]는 최근 시간 윈도우에 대한 평균 큐 길이의 변화추이에 의해 초기의 혼잡을 알아낸다. 큐 길이의 평균이 임계치를 넘으면 게이트웨이는 패킷을 랜덤하게 선택하고 확률적으로 IP 패킷 헤더에 ECN 비트를 표시한다. TCP를 위해 이는 ACK에 언허서 보내지게 된다. 수신기가 ECN과 함께 ACK를 받으면 혼잡윈도우를 반으로 줄이게 된다. ECN 알고리즘을 좀더 발전시켜 Multilevel ECN 알고리즘이 [4]에서 제안되기도 하였다.

두 번째로 Explicit Loss Notification(ELN)을 들 수 있다. 버클리 snoop 프로토콜^[11]에서 이를 사용할 수 있다. ELN은 네트워크의 혼잡과 관계없는 패킷 손실을 송신기에게 알려주는 것이다. 기지국에서 동작하는 snoop agent는 데이터 세그먼트를 받으면 시퀀스에서 홀, 짝 잃어버린 구간을 기억해 놓는다. 중복 ACK가 수신기로부터 도착했을 때 snoop agent는 홀의 목록을 조사한다. 데이터 송신기로 ACK를 전달하기 전에 ACK가 리스트에 있는 세그먼트와 일치한다면 ACK에 ELN bit를 정한다. 송신기가 ELN 정보가 있는 ACK를 받으면 다음 세그먼트를 재전송하고 혼잡제어 알고리즘을 동작시키지 않는다.

세 번째로 Error Notification Ack(ENA)^[7]를 들 수 있다. 본 논문에서 제안하는 EEN과 가장 비슷하고 비슷한 시기에 아이디어^[8]가 제안되었다. ENA 알고리즘은 유선 구간의 혼잡과 무선 구간의 에러를 구분하여 무선 구간의 에러가 발생할 경우 ACK의 option format을 변형한다. Expected number와 손실된 세그먼트와 제대로 수신된 세그먼트의 정보를 포함하고 있는 3개의 Block과 세그먼트가 도착했을 때 에러인가 아닌가를 표현해주는 Bit와 에러인 세그먼트 번호를 갖고 있도록 구성한다. 이 ACK의 정보를 받으면 혼잡제어 알고리즘을 동작시키지 않는다.

무선구간의 에러를 알려주기 TCP 송신자에게 알리기 위해서는 TCP 세그먼트 주소와 TCP 송수신자의 주소를 알아야 하는데 이를 알기 어려우므로, ECN알고리즘은 무선구간의 에러를 보고하는 대신에 네트워크의 혼잡을 알려주는 알고리즘이다. snoop에서 사용되는 ELN 알고리즘은 네트워크의 상황에 따라 패킷의 순서가 일정하지 않음을 간과하였다. 또, ENA 알고리즘은 TCP 세그먼트의 시퀀스 숫자와 TCP 송수신자의 주소를 어떻게 알아내는데에 대한 언급이 없다는 단점이 있다. 만약 패킷에 손실이 발생할 경우, TCP header에서 세그먼트 시퀀스 숫자와 송수신 주소가 가장 많은 부분을 차지하므로 에러가 발생할 가능성이 높다. 그럼 4를 보면 이를 알 수 있다. 그러므로 에러가 발생한 패킷에서 시퀀스 숫자와 송수신 주소를 아는 것은 어렵다는 점을 생각하지 않았다. 이를 위해 본 논문에서는 ECC를 TCP 헤더에 적용하는 방법을 제안하였다.

Ⅲ. EEN 알고리즘 및 구현방법

1. 송수신으로 에러보고

무선 링크 상에서 TCP 성능 향상을 위하여 본 논문에서는, 무선에서 에러가 나면 수신측 TCP에서 그것을 감지하고 바로 에러를 보고하는 NACK을 송신측 TCP로 보내는 방법을 제시한다. EEN 알고리즘은 ECN 알고리즘과 구현 배경은 같지만 혼잡을 알려주는 것이 아니라 무선구간의 에러를 알려주는 것이 다르다. 송신측 TCP로 무선 에러가 난 패킷의 일련 번호가 적힌 NACK을 보냄으로써 다음 전송 시에 그 패킷을 먼저 재전송 되도록 조절한다. 이렇게 함으로써 송신측 TCP는 무선 환경에 의해 손실된 패킷에 대한 ACK가 안 오더라도 타임아웃까지 시간을 기다리지 않고 바로 해당 패킷을 재전송한다. 타임아웃이 발생하면 윈도우의 크기가 줄어 전송률이 매우 낮아지게 되므로 이를 방지하기 위함이다.

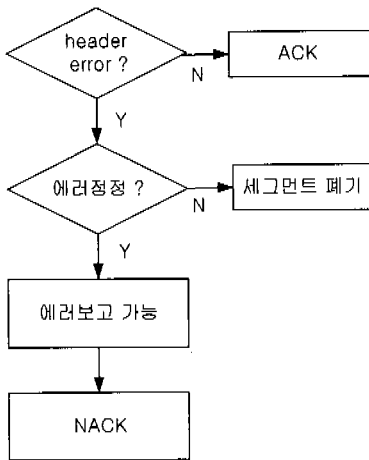


그림 1. EEN 수신측 알고리즘

그림 1.에서 EEN 알고리즘의 flow chart를 보았다. 시뮬레이션 결과에서도 보이듯이, 이를 매우 높은 성능 향상을 보인다. EEN 알고리즘의 단점으로는 에러를 정정하였으나, 시퀀스 숫자 혹은 송수신 TCP의 주소의 에러를 완벽하게 고치지 못하여 오동작을 할 수 있다는 단점이 있다. 그러나 이는 성능 향상에 비해 그 영향이 크지 않다. 그림 2.와 3.에서 볼 수 있듯이 기존의 경우에는 무선에서 에러가 발생할 경우에 송신측 TCP는 타임아웃동안 손실된 패킷에 대한 ACK를 기다리고, 해당 패킷이 오지 않을 경우 이것을 네트워크상의 과부하로 알고 윈도우 수를 줄이게 된다. 제안된 방법에서는

NACK을 송신측 TCP로 보내기 때문에 타임아웃동안 기다리지 않고, 윈도우 수도 줄이지 않는다.

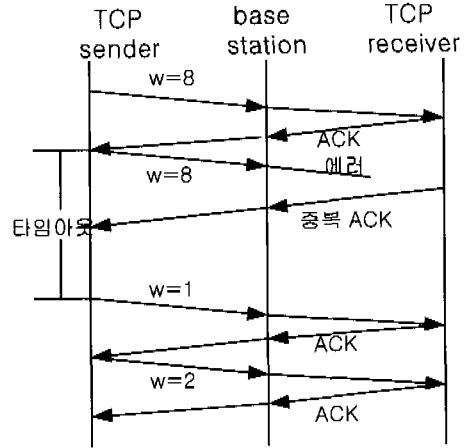


그림 2. 무선 손실로 인한 혼잡윈도우 감소

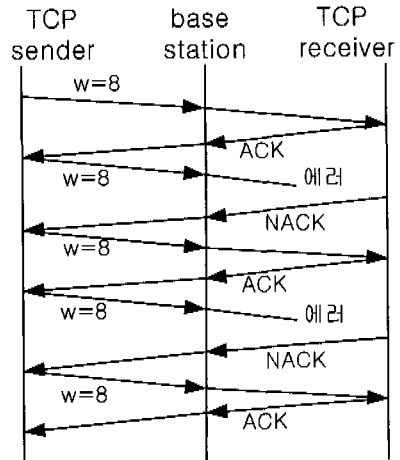


그림 3. 무선 손실 발생시 제안하는 EEN 알고리즘 동작

에러를 보고하기 위해서는 세그먼트의 시퀀스 번호와 송신과 수신 TCP의 주소를 알아야만 한다. TCP 헤더에 에러가 발생하면 이 세그먼트를 버리지 않고, TCP 헤더의 option 부분에 세그먼트의 시퀀스 번호와 송신과 수신 TCP를 알아내기 위해 ECC를 적용하는 방법을 제안한다.

2. Error Correcting Codes

TCP 프로토콜을 무선구간에 적용하면 무선구간의 에러 또는 이동으로 인한 핸드오프에 의해 패킷 손실이 일어나게 된다. 무선 구간은 높은 BER을 가지므로 수신기에서 패킷을 받았더라도 에러가 검

출되면 패킷을 버리게 된다. 또한 핸드오프시 발생하는 패킷 손실의 경우 수신기에서 패킷을 받지 못하고 무선 구간에서 잃어버리게 된다. 핸드오프로 인해 패킷을 잃어버리는 경우는 어쩔 수 없으나, 무선구간의 높은 BER로 인해 에러가 난 세그먼트를 받게 될 경우 이를 이용해 성능 개선을 하지는 것이 본 논문의 아이디어이다.

무선 구간의 에러를 송신측으로 알리기 위해서는 송신측 주소와, 수신측 주소, 그리고 TCP 세그먼트 번호를 알면 가능하다는 사실에 근거하여 이를 위해 ECC(error correcting codes)를 사용한다. 이를 통해 BER을 10^{-7} 까지 줄일 수 있다. 이 점에 있어서 ENA 알고리즘과 분명히 다르다. ECC는 전송되는 데이터 스트림에 redundancy를 넣어 전송하는 동안 발생하는 에러를 검출 및 수정하는 코드를 말하는 것으로 1948년 Shannon에 의해 수학적으로 증명이 되었으며 이후 1960년에 single error correcting이 가능한 Hamming Code가 처음으로 발표되었다. 1960년을 전후해서 BCH code, RS code로 발전하였다. 본 논문에서는 송신측 주소와 수신측 주소, 그리고 TCP 세그먼트 번호를 BCH 코드로 코딩하여 보내는 방법을 제안한다.

3. TCP header 포맷 변형

TCP header에서 check sum은 TCP 헤더 부분에 에러가 났는지 알았는지만을 check하게 된다. 그래서 에러가 났다고 판단하면 그 세그먼트를 버리게 된다. 그러나 우리가 제안하는 모델을 위해 TCP 세그먼트에 error correction field for source and destination port 부분을 코드의 redundancy 부분을 위하여 첨가한다.(그림 4.) 에러가 났을 때 에러를 보고하기 위하여 송신측과 수신측의 주소만이라도 살리기 위해 더해진 부분에 의해 에러를 정정하게 된다. 에러를 고치게 되어 송신측과 수신측의 주소를 알게 되면 에러를 알릴 수 있게 되고, 에러를 고치지 못하게 되면 TCP는 전형적인 혼잡 제어 알고리즘을 수행하게 된다.

현재 IP 주소의 부족으로 인하여 IPv4에서 IPv6로의 움직임이 있다. IPv4에서 IPv6로의 변형은 header부분의 큰 변화를 일으킬 것이기 때문에 본 논문에서 제안된 TCP 세그먼트 헤더의 변형은 그리 큰 문제가 되지 않는다. error를 report 하기 위해서는 IP 주소도 필요한데, IPv6의 Extension Header 중에서 Destination Options Header의 option 부분에 에러 정정 코드의 redundancy를 넣

Source port		
Destination port		
Sequence number		
Ack. number		
HLEN	Reserved	Code bits
Window		
Checksum		
Urgent pointer		
Error correction		
Options(if only)		padding
User data		

그림 4. 변형된 TCP header 포맷

을 수도 있다.

IV. EEN의 성능 분석

1. 시뮬레이션 모델

시뮬레이션 네트워크 상에서 유선 네트워크는 일종의 백본망 역할을 하고, 무선 네트워크는 서브망 역할로 서로 연결된다. 그리고 무선 서브망과 유선 백본 사이의 접속점으로서 동작하는 기지국을 포함한다. 네트워크 상에 무선 단말기가 위치하고, 그 단말기는 자신이 존재하는 셀에 있는 기지국에 의해 관리된다. 기지국은 고정 호스트와 유선으로 연결되어 있어, 무선 단말기와 고정 호스트는 기지국을 통하여 통신을 할 수 있다. 기지국에서 고정 호스트로 가는 경로는 여러 갈래가 존재하며, 그 경로 상에 라우터 역시 여러 개 존재한다. 이것을 시뮬레이션을 위해 모델링을 하면, 무선 링크로 된 부분과 유선 링크로 된 부분으로 나눌 수 있다.(그림5.) 이 모델에서는 전송되는 패킷의 크기를 1000 바이트로, 윈도우 크기를 8로 가정한다. 그리고 채널 특성은 [14]를 따랐다.

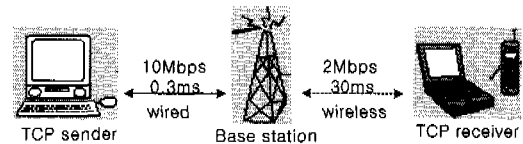


그림 5. 시뮬레이션 모델링

제안된 방법에서 무선 구간의 에러가 난 대부분의 패킷은 근원지 주소와 목적지 주소를 에러 정정

능력에 따라 디코딩하여 NACK을 보내지며 나머지는 에러에 의해서, 혹은 다른 이유에 의해서 근원지 주소와 목적지 주소를 정확히 알지 못해 NACK을 보낼 수 없다. 그리고 시뮬레이션에서는 타임아웃 시간을 0.5s로, round-trip time delay를 30ms로 가정했다. 시뮬레이션은 코딩을 하지 않았을 경우와 했을 경우로 나누어서 해 보았다.

2. 시뮬레이션 결과

무선 무선 TCP-tahoe와 유선 TCP-tahoe에 대해 시뮬레이션을 해보았다. 그림 6.에서처럼 무선 구간에서는 유선 구간에 비해 심각한 성능 저하가 일어나게 된다. 1000개의 세그먼트를 보낸 경우, 유선 링크에서는 윈도우 크기도 거의 8에서 유지되고 걸리는 시간도 매우 짧다. 그러나 무선 링크에서는 전송 윈도우 수가 거의 2에서 유지되고 있음을 볼 수 있다. 이것은 RTT당 두 개 정도의 패킷밖에 못 보내는 것을 의미하며 그만큼 전송 속도가 늦어지고 있다는 것을 뜻한다. 같은 시간 동안 유선의 TCP는 1000개의 세그먼트를 다 보내는데, 무선의 TCP는 약 200개밖에 못 보내고 있다. 제안된 방법의 경우에는 거의 최대 윈도우 수에서 유지되고 있다.

중간 중간에 가끔씩 윈도우 수가 주는 것은 기지국에서 근원지 주소와 목적지 주소를 정확히 알지 못해서 에러가 난 경우이다. 이 경우에는 무선에 의한 에러가 아니므로 TCP는 그것을 혼잡에 의한 에러로 가정하고 윈도우 수를 줄일 것이다. 만약 기지

국에서 100%근원지 주소와 목적지 주소를 안다면 제안된 방법의 경우에는 처음 시작을 제외하고는 끝까지 윈도우 크기를 최대로 유지할 것이다.

그림 7.을 살펴보면, 1000바이트의 세그먼트 1000개를 보냈을 경우 EEN 알고리즘을 적용하지 않았을 경우보다 EEN 알고리즘을 적용한 경우가 매우 높은 성능 향상을 보였다. 또한 에러 정정 개수에 따라 보내기 위해 필요한 시간이 짧아짐을 알 수 있다.

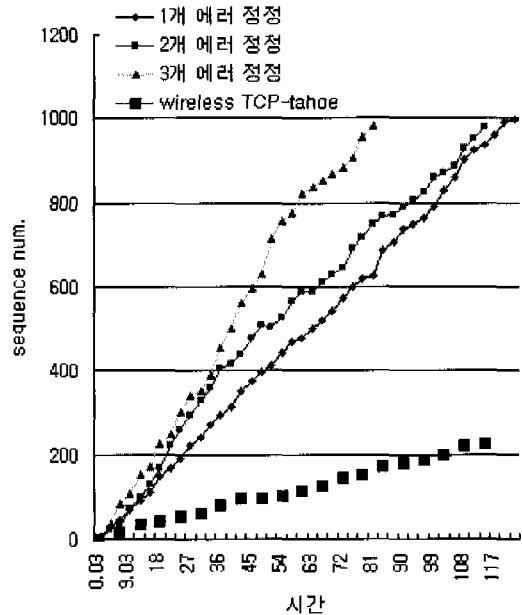


그림 7. TCP-tahoe에서 에러 정정 개수에 따른 성능 비교

1개의 에러를 정정한 것과 2개의 에러를 정정한 것 사이에는 성능 차이가 많지 않지만, 2개의 에러를 정정한 것과 3개의 에러를 정정한 것 사이에는 큰 성능 차이가 있음을 알 수 있다. 윈도우의 크기에 있어서도 에러 정정 개수에 따라 변하고 있음을 알 수 있다. 적은 수의 에러를 정정했을 때 평균적인 혼잡윈도우의 크기는 약 3~4이고, 더 많은 에러를 정정함에 따라 최대 혼잡윈도우크기까지 도달하는 횟수가 잦아지는 것을 그림 8~11.을 통해 볼 수 있다. TCP-tahoe로 시뮬레이션 했으므로 에러가 나면 윈도우 크기가 1까지 떨어지게 된다.

TCP-reno에서는 빠른 재전송이 있는데, EEN을 사용하면 빠른 재전송과 빠른 회복은 필요하지 않게 된다. 즉 다시 말하면 빠른 재전송과 빠른 회복을 EEN이 더 빠르게 대신한다고 말할 수 있다.

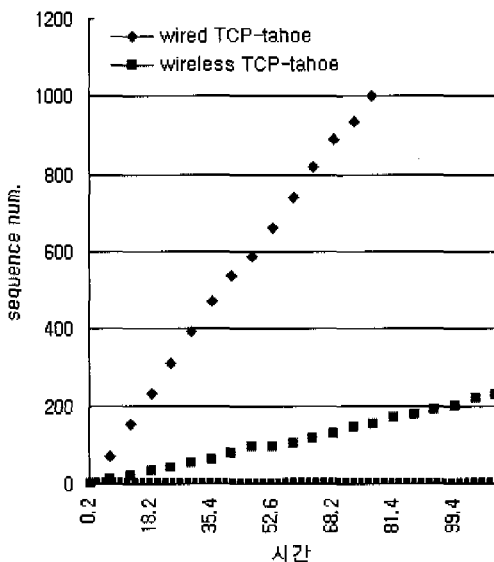


그림 6. TCP-tahoe 유선과 무선망에서의 성능 비교

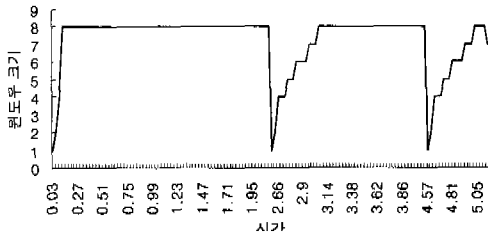


그림 8. 유선망에서 혼잡윈도우 크기 변화

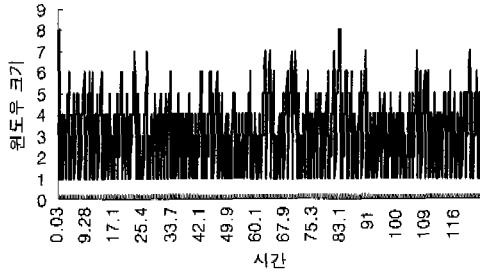


그림 9. 제안된 EEN 알고리즘에서 혼잡윈도우 크기 변화 : 1개 에러 정정

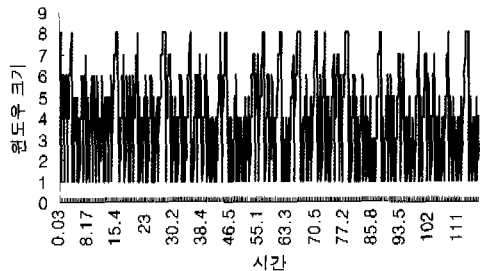


그림 10. 제안된 EEN 알고리즘에서 혼잡윈도우 크기 변화 : 2개 에러 정정

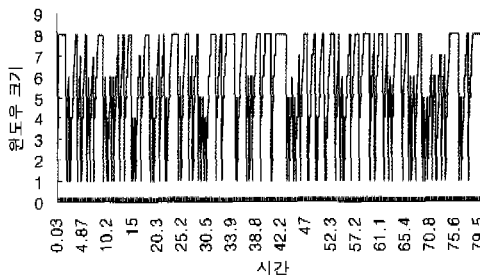


그림 11. 제안된 EEN 알고리즘에서 혼잡윈도우 크기 변화 : 3개 에러 정정

표 1.에서는 유선 TCP-tahoe, 무선 TCP-tahoe에서 에러 정정 능력에 변화를 준 것, 무선 TCP-reno와 무선 TCP-tahoe를 시뮬레이션하여 평균 윈도우 크기와 총 시간, 평균 throughput을 정리하여 나타

내었다. 평균 throughput은 식(1)과 같이 계산하였고 무선 reno는 성능 비교를 위해 넣었다. 표를 보면 무선 reno보다 무선 tahoe에 EEN을 적용하였을 때 성능이 좋아지는 것을 알 수 있다. 평균 윈도우 크기와 시간은 반비례하게 되며 성능 비교는 throughput 평균에서 더 확실하게 나타나게 된다.

표 1. 모의 실험 결과

	평균 window 크기(pkt)	총시간(s)	throughput 평균
wired tahoe	7.23	5.14	197
에러 3개 정정	5.59	81.21	98.94
에러 2개 정정	4.06	116.15	51.01
에러 1개 정정	3.27	123.62	35.24
wireless reno	2.09	198.56	9.15
wireless tahoe	1.75	205.91	7.47

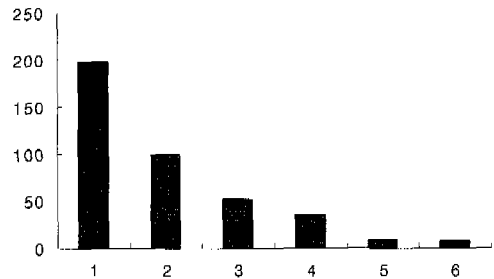


그림 12. 평균 throughput 비교

$$throughput_{\text{평균}} = \frac{\text{사이클당보낸패킷수}}{\text{한사이클시간}} \quad (1)$$

throughput 평균을 구하기 위하여 시뮬레이션을 통해 얻은 한 사이클당 시간은 wired TCP-tahoe의 경우 36.7ms, 에러 정정 능력이 1 일 경우 56.6ms, 2 일 경우 79.6ms, 3일 경우 92.8ms, wireless TCP-reno 경우 228ms, wireless TCP-tahoe 경우 234ms 이다.

그림 12.에서 1은 유선 tahoe, 2는 3개의 에러를 정정한 경우, 3은 2개의 에러를 정정한 경우, 4는 1개의 에러를 정정한 경우이며, 5는 무선 reno이고, 6

은 무선 tahoe 이다. 3개의 에러를 정정하면 유선과 throughput에 있어서 어느 정도 차이가 생기게 되는데, 이는 에러 정정 능력을 더 키움에 따라 그 차이를 더욱 줄일 수 있다. EEN의 에러 정정 능력이 약하더라도 무선 reno보다 성능이 좋은 것을 볼 수 있다. 에러를 하나 혹은 둘을 정정할 때의 성능 차이보다 에러를 둘 혹은 셋을 정정할 때의 성능 차이가 더 나는데 이는 데이터의 burst 특성 때문이다.

V. 결론

현재 인터넷에서 표준으로 사용중인 TCP는 신뢰성 높은, 고정 호스트와 유선 링크 기반의 네트워크를 고려해서 구현된 프로토콜이다. 따라서 전송 중에 발생하는 데이터의 손실을 네트워크에 부하로 인한 혼잡으로 오인하여 송신측 TCP에서 전송원두의 크기를 줄이게 된다. 즉 전송률을 줄여 혼잡을 피하는 알고리즘을 수행한다. 그런데 무선 링크는 높은 BER 특성을 가지기 때문에, 혹은 사용자가 이동을 해서 핸드오프가 발생하기 때문에 데이터가 손실된다. 이러한 무선 링크가 통신 네트워크 상에 존재할 경우 기존 TCP의 알고리즘은 TCP의 종단 간에 심각한 성능 저하를 초래한다. 따라서, 본 논문에서는 무선 통신 환경에서 에러가 발생할 경우, 기지국에서 무선에 의한 에러를 정확히 인식하고 에러가 난 패킷의 근원지 주소와 목적지 주소를 알게 된다면 TCP 수신자는 TCP 송신자에게 이를 알리는 EEN 알고리즘을 제안했다. 네트워크의 혼잡과 무선의 에러를 구분하고 혼잡 제어 메커니즘에 의한 동작을 수행을 막은 상태에서 손실된 패킷을 재전송 하여 네트워크의 성능을 향상시켰으며, 이를 위해 TCP 헤더에 에러정정코드를 적용하였다. 시뮬레이션 결과 EEN 알고리즘을 적용하여 성능이 향상됨을 보였고, 에러를 많이 정정할수록 TCP 프로토콜의 평균 throughput이 증가되어 나타났다.

참고 문헌

[1] H. Balakrishnan, S. Seshan and R.H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks", *ACM Wireless Networks*, vol.1, No. 4, pp. 469-481, December, 1995.

[2] A. Bakre and B.R.Badrinath, "I-TCP: Indirect TCP for mobile hosts", *Proceedings of the*

15th International Conference on distributed Computing Systems, pp.136-143. June 1995

[3] S. Floyd, "TCP and Explicit Congestion Notification", *Computer Communications Review*, vol.24 No.5, pp.10-23. October 1994.

[4] Arjan Duresi, Mukundan Sridharan, Chunlei Liu, Mukul Goyal, Raj Jain "Multilevel Early Congestion Notification", *Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics SCI, ABR over the Internet*, Orlando, FL, pp. 12-17, July, 2001

[5] J. B. Postel. "Transmission Control Protocol", *Information Sciences Institute, Marina del Rey, CA, RFC-793*, September 1981

[6] Floyd S. and Jacobson V., "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Transactions on networking*, Vol.1, No.4, pp.397-413, August 1993

[7] 이정규, 김상희, "TCP를 사용하는 위성링크에서의 성능 향상을 위한 ENA 알고리즘", *한국통신학회논문지*, v.25, n.8A, pp.1177-1185, 2000

[8] 유영선, 김낙명, 김경희, "이동통신환경에서의 TCP 성능향상에 관한 연구", *한국통신학회 하계 종합학술발표회(下)*, pp.1006-1009, 2000. 7 월

[9] H.Balakrishnan, "Challenges to Reliable Data Transport over Heterogeneous Wireless Network", PhD Thesis, *University of California at Berkeley*, 1998

[10] K. Brown and S. Singh, "M-TCP: TCP for Mobile Cellular Networks", *Computer Communication Review*, pp.19-23, October 1997

[11] Nachiket Deshpande, "TCP Extensions for wireless networks", <http://www.cis.ohio-state.edu/~deshpand>

[12] Vaidya N, Biaz S, "Discriminating Congestion Losses from Wireless Losses using Inter-Arrival Times at the Receiver", Technical Report 98-014, *Texas A&M University*, June 1998

[13] M.Naghshineh, M.Schwartz and A.S.Acampora, *Issues in Wireless Access Broadband Networks, Wireless Information Networks, Architecture, Resource Management*

and Mobile Data, edited by J.M.Holtzman,
Kluwer Academic Publishers, 1996

- [14] G. T. Nguyen, R. H. Katz, B. D. Noble, and
M. Satynarayanan, "A trace-based approach
for modeling wireless channel behavior",
Winter Simulation Conference, December,
1996.

김 경 회(Kyung-hee Kim)

준회원



2000년 2월 : 이화여자대학교
전자 공학과 졸업
2000년 3월~현재 : 이화여자
대학교 정보통신학과
석사과정

<주관심 분야> CDMA 이동 통신 시스템, 위성 통
신, 채널 코딩

김 낙 명(Nak-myeong Kim)

정회원



1980년 2월 : 서울대학교 전자
공학과 졸업
1982년 2월 : KAIST 전기 및
전자공학과 석사
1990년 2월 : 미국 Cornell
University 전기공학과
박사

1990년~1996년 : LG정보통신(주) 책임연구원
1996년~현재 : 이화여자대학교 공과대학 정보통신학
과 부교수

<주관심 분야> 디지털 이동 통신, 위성 통신, 통신
망 이론