

논문 2008-45TC-9-5

혼잡 및 무선 구간 손실의 차별적 처리를 통한 WTCP 성능 개선

(Performance Improvement of WTCP by Differentiated Handling of
Congestion and Random Loss)

조 남 진*, 이 성 창**

(Namjin Cho and Sungchang Lee)

요 약

유선망을 가정하여 설계된 기존의 TCP를 무선망이 혼재된 망에서 그대로 사용하면 무선구간의 랜덤 손실도 유선구간의 혼잡 손실과 동일하게 간주하게 된다. 이러한 잘못된 판단과 조치는 대역폭을 낭비하여 전체 네트워크의 처리량(throughput)을 저하하게 된다. 이러한 문제를 하기 위한 노력들이 많이 있었다. 본 논문에서는 유선망 혼잡과 무선구간 손실을 판별할 개선된 방법을 사용하여 망 상황 진단의 정확도를 높이고, 판단된 각 상황에 대해서 망 처리량을 높일 수 있는 개선된 제어 방법을 제안하였다. 또한, 시뮬레이션을 통해 제안한 알고리즘의 성능을 TCP Westwood, TCP Venow와 비교하여 평가하였다.

Abstract

The traditional TCP was designed assuming wired networks. Thus, if it is used networks consisting of both wired and wireless networks, all packet losses including random losses in wireless links are regarded as network congestion losses. Misclassification of packet losses causes unnecessary reduction of transmission rate, and results in waste of bandwidth. In this paper, we present WTCP(wireless TCP) congestion control algorithm that differentiates the random losses more accurately, and adopts improved congestion control which results in better network throughput. To evaluate the performance of proposed scheme, we compared the proposed algorithm with TCP Westwood and TCP Venow via simulations.

Keywords : wireless TCP, congestion loss, random loss, available bandwidth, congestion loss rate

I. 서 론

인터넷과 무선망의 눈부신 발전은 인류의 발전에 많은 기여를 하였으며, 그 사용 범위가 점점 확대 되면서 다양한 분야에 활용되고 있다. 인터넷 사용에 있어 가장 핵심적인 프로토콜로 자리 잡은 TCP는 기존의 유선망에 한정되어 사용되었지만 무선망의 급격한 사용 증가로 인하여 유무선 망에 확대 적용하게 되었다.

무선망은 유선망에 비해 Fading, Noise, Interference, Handoff 등에 의한 높은 BER(Bit Error Rate)과 낮은 대역폭, 동적인 무선 선로의 환경변화 등으로 인하여 많은 패킷이 손실된다.^[1] 일반적으로 무선망은 기존의 유선망과 연동하여 복합적인 구조를 갖게 된다. 그러나 기존의 유선망 TCP는 이러한 무선망의 특성을 고려하지 않고 설계되어 패킷의 손실이 발생하면 혼잡에 의한 손실이라고 판단하고 혼잡 제어 메커니즘을 호출하여 패킷의 전송률을 줄인다. 유선망에서는 패킷 손실의 90% 이상이 중간 노드인 Router의 Buffer Overflow로 인한 혼잡이 원인이 되었으나, 무선망에서 발생하는 패킷의 손실은 높은 BER로 인하여 발생하게 된다. 따라

* 학생회원, ** 정회원 한국항공대학교
(Korea Aerospace University)

※ 이 논문은 2005년도 한국항공대학교 교비지원 연구비에 의하여 지원된 연구의 결과임.

접수일자: 2008년8월13일, 수정완료일: 2008년9월17일

서 혼잡에 의한 패킷 손실이 아닌 경우라면 패킷의 전송률을 감소하여 재전송을 하는 것은 대역폭을 낭비하여 네트워크 성능을 저하시키게 된다. 그러므로 패킷 손실의 원인을 혼잡에 의한 손실과 무선망에서의 임의적인 손실로 구분하여 전송률을 조절해야 한다.

무선망에 적합한 TCP로서 패킷 손실의 구분을 위한 여러 가지 알고리즘이 제안되었다. 연결 분할 접근 방식의 I-TCP(Indirect TCP)^[2], 지역 회복 기법인 Snoop TCP^[3~4]와 종단 간 연결 방식의 End-to-End 기법이 있다. End-to-End 기법은 TCP 자체적으로 해결하는 방식이 아닌 Network Layer 레벨에서 도움을 받아 해결하는 방법으로, 대표적으로 종단 간의 연결성을 지향하면서 송신 단의 프로토콜 적용만으로 효과를 가지는 TCP Westwood^[5~7], TCP Veno^[8] 등이 있다.

본 논문에서는 TCP Veno의 문제점을 지적하고 네트워크의 혼잡 상태를 판단하여 손실의 유형을 구분하는 알고리즘을 제안한다. 네트워크의 혼잡이 심할 경우, TCP Veno는 무선 상황에 의한 임의적인 손실임에도 불구하고 패킷 손실의 원인이 혼잡에 의한 것이라고 잘못 판단한다. 따라서 TCP Veno의 문제점을 실험적으로 분석하고 해결하기 위하여 RTT(Round Trip Time)을 이용한 대역폭 측정 후, 이를 개선한 알고리즘을 적용한다. 혼잡에 의한 패킷 손실이 아닐 경우, Congestion Loss Rate 파라미터를 이용하여 확실한 무선 손실을 구분하고, 적절한 전송률 조절 방법을 통하여 유무선 혼합 망 환경에 적합한 TCP를 제안한다.

본 논문의 구성은 다음과 같다. II장에서는 기존 유선망에서의 TCP에 대한 간략한 소개와 무선망에 적합하게 제안된 여러 연구들을 소개한다. III장에서는 TCP Veno의 문제점을 분석하고 그 문제점을 보완할 수 있는 방법을 제안한다. IV장에서는 시뮬레이션을 통하여 제안한 알고리즘의 성능을 다른 알고리즘과의 비교를 통하여 분석한다. 마지막으로 V장에서는 본 논문의 결론 및 향후 연구 방향을 제시한다.

II. 기존 TCP 알고리즘

1. 유선망의 TCP 혼잡제어 기법

TCP의 혼잡제어 기법은 패킷 손실이 발생한 경우, 효과적으로 대처하기 위한 전송률 조절 방법으로 네 가지 알고리즘을 사용한다. 즉, Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery를 결합시켜 혼잡 상태를 제어한다.^[9]

Tahoe TCP는 최초의 TCP에 망의 혼잡 완화를 효율적으로 하기 위하여, Slow Start, Congestion Avoidance, Fast Retransmit의 세 가지 알고리즘을 추가한 것으로, 송신 노드가 3개의 중복된 ACK를 수신하면 Fast Retransmit 알고리즘에 의하여 타임아웃을 기다리지 않고 패킷을 재전송한다.^[10]

Reno TCP는 Tahoe TCP를 기반으로 Fast Recovery 이후에 전송률을 현저히 낮추어 망의 대역폭을 낭비하는 것을 막는다.^[11]

이 외에도 NewReno TCP^[12~13], TCP-SACK^[14], TCP Vegas^[15] 등의 여러 가지 버전들의 TCP가 있다.

2. 무선 환경에 적합한 TCP 혼잡제어 기법

가. Split Connection 기법

I-TCP(Indirect TCP)는 Split Connection 기법의 하나로 그림 1과 같이 종단 간 연결을 유선망과 무선망의 두 부분으로 나누고 그 중간 지점의 Base Station에서 연결을 관리한다. 고정 호스트에서 이동 호스트로 전송한 데이터를 Base Station에서 Buffering하여 그 데이터에 대한 ACK를 즉시 고정 호스트로 전송하고, Buffering한 데이터를 다시 이동 호스트로 전송한다. 무선망에서 패킷 손실이 일어나면, 고정 호스트에서 패킷을 재전송 하지 않고 Base Station에서 중복 ACK나 타임아웃을 감지하여 재전송을 수행함으로써 무선망에서의 패킷 손실이 송신 단에 영향을 끼치지 않게 할 수 있다.^[2]

I-TCP는 기존 TCP의 구조를 변경하지 않는 장점이 있으나 TCP의 가장 큰 특징인 종단 간 의미 구조가 깨진다는 단점이 있다. 즉, 수신 단에서 실제로 데이터를 받지 못했음에도 불구하고 송신 단은 ACK를 받을 수 있다는 것이다. 또한, I-TCP가 적용된 수신단과 기지국의 전송 계층(Transport Layer)에 별도의 소프트웨어가

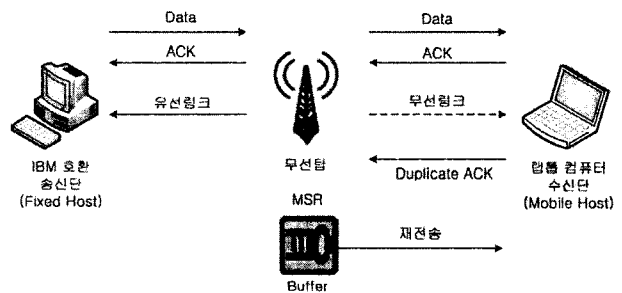


그림 1. I-TCP의 구조
Fig. 1. Architecture of I-TCP.

필요하고, TCP의 성능이 기지국에 가용한 버퍼 크기에 따라 다르게 된다. 이러한 Split Connection 기법의 또 다른 대표적인 프로토콜에는 M-TCP^[16]가 있다.

나. Local Recovery 기법

Local Recovery 기법의 가장 대표적인 프로토콜은 Snoop TCP가 있다. 유선망과 무선망의 접속점에 위치한 Base Station에는 Snoop 모듈이 존재한다. Snoop 모듈은 이 모듈을 통과하는 모든 패킷을 검사하여 기본적으로 데이터와 ACK는 그대로 보내고 수신 단으로부터 아직 ACK를 받지 않은 패킷들만 버퍼에 저장한다. 패킷이 모듈을 통과할 때, 중복 ACK나 타임아웃이 발생하면 패킷 손실로 판단하여 그 패킷에 대한 중복 ACK는 폐기하고, 버퍼에 저장되어 있는 패킷을 재전송한다 [3-4].

그림 2는 Snoop TCP의 구조를 도시한 것이다. I-TCP와는 다르게 Snoop 프로토콜에서 Base Station은 데이터와 ACK를 Relay하는 역할만을 담당한다. 또한 무선망에서의 패킷 손실을 감추기 위해서 중복 ACK를 Snooping하는 역할과 무선망의 에러로 인한 패킷 손실을 재전송하는 역할을 한다. 따라서 불필요한 혼잡 제어를 방지하게 되고 종단 간 연결 개념을 유지하며 기존 응용 프로그램과의 호환성을 제공하기 때문에 응용 프로그램의 수정 없이도 TCP 연결의 성능을 향상시킬 수 있다.

그러나 기지국에서 지역적인 재전송을 통해 무선망의 패킷 손실을 복구하고 있는 동안 유선망에 접속되어 있는 송신 단은 ACK 패킷을 받지 못하기 때문에 유선망에서의 가용 대역폭 활용이 저하될 수 있다. 또한 지역적 재전송에 의한 패킷 손실 복구 작업이 오랫동안 지연되는 경우에는 무선망에서 발생하는 패킷 손실을 송신 단에게 숨길 수 없는 경우가 발생하게 된다.

다. End-to-End Mechanism

End-to-End 기법은 기존의 유선망에 사용하던 TCP를 수정하여 종단 간의 연결을 유지하면서 무선망의 특성에 의한 임의적인 무선 손실과 망의 혼잡에 의한 손실을 구별하는 효과적인 동작 방법이다. 이 방법은 라우터나 Base Station에 별도의 모듈이 필요하지 않기 때문에 기존 TCP들의 장점을 모두 수용할 수 있다. 또한 비교적 정확한 예측으로 패킷 손실을 구분하여 불필요한 혼잡 제어 알고리즘이 발생하는 것을 방지한다.

이러한 End-to-End Mechanism과 관련된 대표적인

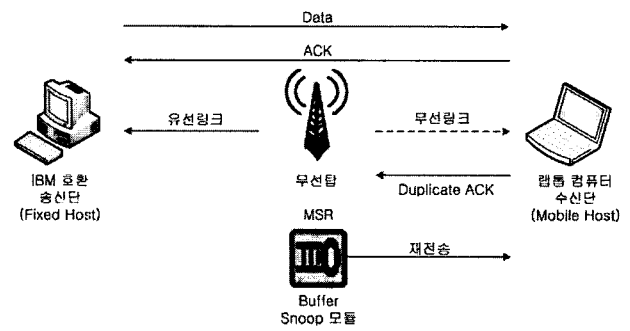


그림 2. Snoop TCP의 구조
Fig. 2. Architecture of Snoop TCP.

알고리즘으로 TCP Westwood와 TCP Veno가 있다.

(1) TCP Westwood

종단 간 연결을 유지하며 무선망의 특성으로 인한 손실을 효과적으로 재전송하는 혼잡제어 알고리즘을 가지는 TCP Westwood는 샘플링과 Exponential Filtering을 통해 송신 윈도우의 크기를 측정하는 방식이다^[5]. 빠른 재전송과 효과적인 혼잡 회피를 통해 기존의 TCP보다 더 효율적인 동작이 가능하며 무선 상의 손실에 대한 조정에 있어서도 매우 효과적이다.

TCP Westwood 알고리즘은 자신이 보낸 패킷의 ACK 수신률과 Exponential Filtering을 이용하여 가용 대역폭을 측정한다. 중복 ACK나 타임아웃 발생 시 측정된 가용 대역폭으로 전송률을 결정한다. 가용대역폭의 측정 방법은 다음과 같다.

$$S_BWE_k = \frac{d_k}{(t_k - t_{k-1})} \tag{1}$$

식 (1)에서 d_k 는 송신한 데이터, t_k 는 송신한 데이터에 대한 ACK를 받은 시간, t_{k-1} 는 이전 ACK를 받은 시간을 나타낸다. S_BWE_k 는 ACK를 받을 때마다 계산되고 Low-Pass Filter를 통해 가용 대역폭이 계산된다.

$$BWE_k = \frac{\frac{2\tau}{t_k - t_{k-1}} - 1}{\frac{2\tau}{t_k - t_{k-1}} + 1} BWE_{k-1} + \frac{S_BWE_k + S_BWE_{k-1}}{\frac{2\tau}{t_k - t_{k-1}} + 1} \tag{2}$$

식 (2)에서 BWE_k 는 시간 t_k 에서의 가용대역폭, $\frac{1}{\tau}$ 는 필터의 cut-off frequency를 나타낸다^[5].

TCP Westwood의 장점은 하위 레이어의 도움과 TCP 세그먼트 헤더에서 사용하고 있는 옵션의 조정이

필요 없고 기존의 TCP보다 무선망의 에러에 대하여 링크 사용률을 더 높일 수 있다는 것이다. 또한 다른 TCP와의 동작에서 높은 친화성과 안정성을 보인다.

하지만 임의적인 손실이 많이 발생할 때 불완전하게 실행되고 송신 단이 무선 구간에 있을 경우에 대한 고려가 미흡하다는 단점이 있다.

(2) TCP VenO

TCP VenO는 Reno TCP와 TCP Vegas의 혼잡제어 매커니즘을 적절하게 결합하여 무선망에서 보다 좋은 성능을 발휘하는 End-to-End 기법중의 하나이다. VenO는 망의 혼잡 레벨을 파악하기 위하여 Vegas와 유사한 매커니즘을 사용한다. 하지만 Proactive 혼잡제어를 사용하여 패킷 손실을 미연에 방지하는 Vegas와는 다르게 VenO는 패킷 손실이 발생하였을 경우 패킷 손실이 혼잡에 의한 손실인지 무선망의 특성에 의한 임의적인 손실인지를 파악한다.

Vegas에서는 Expected와 Actual Rate를 다음과 같이 측정한다.

$$Expected = cwnd / BaseRTT \quad (3)$$

$$Actual = cwnd / RTT \quad (4)$$

$$Diff = Expected - Actual \quad (5)$$

BaseRTT는 측정된 RTT값 중 가장 작은 값이며, RTT는 현재의 Round Trip Time을 나타낸다. cwnd는 현재 Congestion Window Size이다. 식 (3)에서 Expected는 TCP가 설정되고 난 이후 망의 상황이 가장 좋을 때의 처리량이며, 식 (4)의 Actual은 현재의 처리량이다. 식 (5)에서 Diff는 망의 상황이 가장 좋았을 때와 현재의 망 상태를 비교하는 것이다. RTT가 BaseRTT보다 크다는 것은 병목링크에 패킷이 축적되어 있다는 의미이며 VenO는 링크상의 라우터 큐의 backlog인 N을 식 (6)과 같이 구한다.

$$N = Actual * (RTT - BaseRTT) = Diff * BaseRTT \quad (6)$$

VenO 알고리즘의 핵심 아이디어는 N값을 측정하는 목적이 혼잡 회피를 위한 선형 처리가 아닌 현재의 연결 상태가 혼잡인지 아닌지를 가리키는 것이다. 패킷 손실이 없을 경우 윈도우 사이즈를 늘리는 Reno의 기본 아이디어는 VenO에도 적용된다. 하지만 패킷 손실 감지 시 $N < \beta$ 이면, 혼잡에 의한 것이 아닌 임의적인 손실로 간주하여 Reno와 다른 제어 기법을 사용한다.

VenO에서 β 의 값은 3으로 하는 것이 가장 좋다는 것을 실험적으로 입증하였다.

VenO는 손실이 일어나면 패킷 손실의 원인을 구별한 후, 그에 맞는 전송률을 조절함으로써 성능을 향상시켰다. 하지만 RTT를 사용하여 네트워크 상태를 측정하였기 때문에 Router의 Buffer에 쌓인 패킷과 ACK의 양까지 고려되어 네트워크의 정확한 상태를 판단을 할 수 없으며 이에 따라 전송률을 조절하는 문제점이 있다.

III. 개선된 TCP 알고리즘

1. 망의 상황 판단 분석

VenO 알고리즘은 RTT를 사용하여 네트워크의 상태를 파악한다. 하지만 네트워크의 혼잡으로 인한 패킷의 손실은 병목링크의 Buffer가 Overflow 될 때 발생하기 때문에 패킷 손실의 원인을 정확히 판단할 수 없다. 따라서 본 논문은 네트워크의 혼잡이 심할 경우 VenO의 문제점을 분석하기 위하여 VenO 알고리즘에 의해 임의적인 손실을 잘못 판단하는 오류율을 측정하였다.

그림 3은 시간에 따른 VenO의 N값을 나타낸다. Real random loss는 무선 상황에 따른 임의적인 패킷 손실이 일어난 것으로, VenO에서 패킷 손실의 원인을 판단하는 β 와 비교할 때 잘못된 판단을 하는 오류율이 약 65%로 나타났다. 즉, 임의적인 패킷 손실임에도 불구하고 혼잡에 의한 패킷 손실로 잘못 판단하여 대역폭의 낭비와 동작의 손실을 가져온다.

네트워크의 혼잡이 심할 경우 VenO의 잘못된 판단으로 인한 대역폭 낭비와 Westwood의 가용대역폭의 분석을 위하여 cwnd를 측정하였다. 그림 4는 그림 3과 동일한 망 상황에서 Westwood와 VenO의 cwnd를 나타낸다. 잘못된 망의 상황 판단으로 인하여 fluctuation이 심한 VenO에 비해 Westwood는 전체적으로 유연하게 전송률을 조절한다.

본 논문은 망의 혼잡 상황 시 가용대역폭을 계산하고 이를 VenO에 적용하는 알고리즘을 제안한다.

네트워크의 혼잡이 심할 경우 VenO의 잘못된 판단으로 인한 대역폭 낭비와 Westwood의 가용대역폭의 분석을 위하여 cwnd를 측정하였다. 그림 4는 그림 3과 동일한 망 상황에서 Westwood와 VenO의 cwnd를 나타낸다. 잘못된 망의 상황 판단으로 인하여 fluctuation이 심한 VenO에 비해 Westwood는 전체적으로 유연하게 전송률을 조절한다.

본 논문은 망의 혼잡 상황 시 가용대역폭을 계산하고

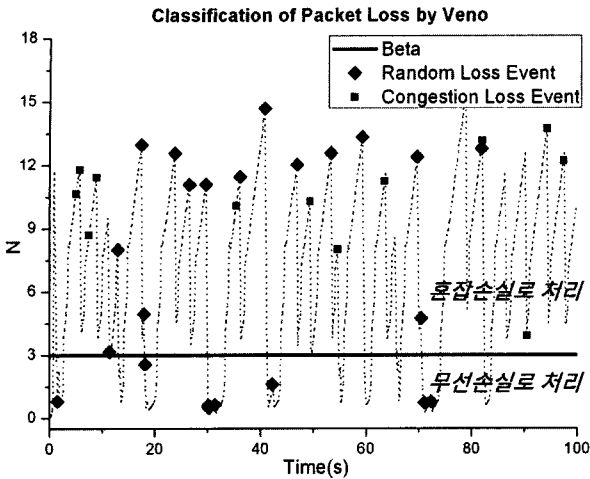


그림 3. Veno의 무선 및 혼잡 손실 판별
Fig. 3. Classification of packet loss by Veno.

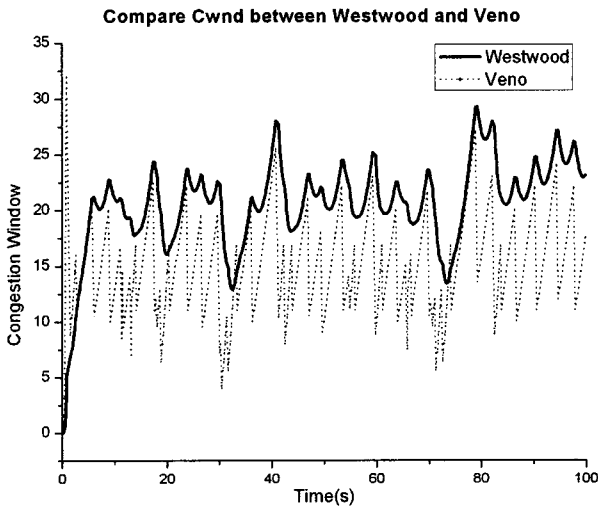


그림 4. Westwood와 Veno의 cwnd 비교
Fig. 4. Compare cwnd between Westwood and Veno.

이를 Veno에 적용하는 알고리즘을 제안한다.

2. Congestion Loss Rate

네트워크의 혼잡이 심하지 않다면, 패킷 손실의 원인은 대부분 무선 상황에 의한 임의적인 손실이다. Veno는 임의적인 손실 판단 시 cwnd를 1/5로 줄이고, 임의적인 손실이 빈번하게 일어날 경우 이는 대역폭을 낭비하는 결과를 초래한다. 따라서 다음과 같은 congestion loss rate을 사용하여 확실한 무선 손실일 경우 전송률을 그대로 유지한다¹⁷⁾.

$$con_T_i = \frac{C_i}{(T_i - T_{i-1})} \quad (7)$$

식 (7)에서 C_i 는 혼잡에 의한 패킷 손실의 수, T_i 는

무선 손실이 발생했을 때의 시간을 나타낸다. 즉, 임의적인 패킷 손실 사이에 발생한 혼잡 손실의 비율을 계산하여 현재 일어난 무선 손실을 판단한다. 하지만 무선 손실 사이의 시간 간격이 크면 망의 상황이 변화할 수 있고, 송신 단에서 전송하는 패킷의 양이 항상 변화하므로 정확한 망의 상태를 반영할 수 없다. 따라서 일정한 시간 간격 내에서 전송하는 패킷의 양에 대한 혼잡 손실의 비율을 고려해야 한다.

본 논문은 식 (7)의 congestion loss rate를 다음과 같이 수정하여 무선 손실을 판단하는 알고리즘을 제안한다.

$$con_LR_i = \frac{C_i}{cwnd_i}, (T_i = BaseRTT * 5) \quad (8)$$

식 (8)에서 T_i 는 일정한 시간 간격으로 나타나며, C_i 는 T_i 동안 일어난 혼잡에 의한 손실 횟수, $cwnd_i$ 는 T_i 동안 보낸 패킷의 양을 나타낸다.

3. 망의 상황 판단에 따른 패킷 손실의 처리

가. 망의 혼잡 상태 판단 시

패킷 손실 감지 시, 라우터 큐의 backlog 값인 N을 통하여 현재 망의 상태가 혼잡하다고 판단되면 측정된 가용 대역폭을 사용하여 ssthresh (slow start

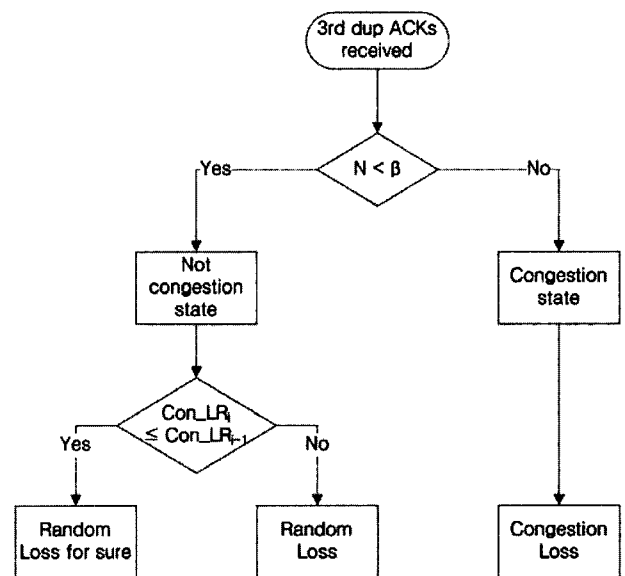


그림 5. 제안 알고리즘의 흐름도
Fig. 5. Flow chart of proposed algorithm.

```

if (3rd duplicate ACKs) then
    // 3rd duplicate ACK received
    if (  $N \geq \beta$  ) then
        // consider state to be congestion state
        ssthresh = ( BWE*BaseRTT ) / size
    else if (  $N < \beta$  ) then
        // consider state to be not congestion state
        if (  $con\_LR_i \leq con\_LR_{i-1}$  )
            // consider loss to be random loss for sure
            Keep All Parameter
        else
            // consider loss to be random loss not for
            sure
            ssthresh = cwnd * 4/5
        end if
    end if
end if

if ( Timeout ) then // a Timeout expires
    ssthresh = ( BWE*BaseRTT ) / size
    if ( ssthresh < 2 )
        ssthresh = 2
    end if
    cwnd = ssthresh
end if
    
```

그림 6. 제안 알고리즘의 의사 코드
Fig. 6. Pseudocode of proposed algorithm.

threshold)를 결정한다. 가용 대역폭은 망의 혼잡 상태만을 반영하여 계산되고 이를 전송률에 적용함으로써 대역폭의 낭비를 줄이게 된다.

나. 망의 비 혼잡 상태 판단 시

패킷 손실 감지 시, 현재 망의 상태가 비 혼잡 상태라고 판단이 되면 congestion loss rate를 사용하여 확실한 무선 손실을 구분한다. 확실한 무선 손실일 경우 전송률을 감소할 필요가 없기 때문에 전송률을 그대로 유지한다.

IV. 시뮬레이션 및 분석

1. 시뮬레이션 환경

제안한 TCP의 성능평가를 위하여 NS-2(Network Simulator 2) ver-ns2.26^[18]를 사용하였다. 시뮬레이션

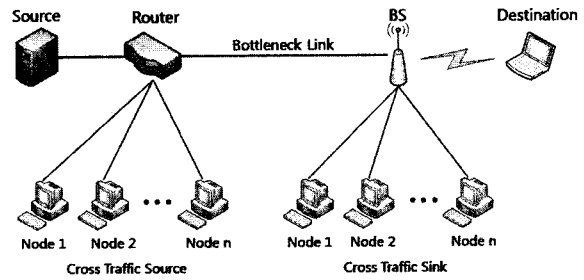


그림 7. 시뮬레이션을 위한 네트워크 구성
Fig. 7. Topology of network for simulation.

표 1. 시뮬레이션을 위한 파라미터
Table 1. Parameters for simulation.

Parameters		Value
Bandwidth / Delay	Wired	10Mb/1ms
	Bottleneck Link	10Mb/50ms
	Wireless	2Mb/0.01ms
Number of Flows (n)		6, 12, 18
Queueing Policy		DropTail
Type		FTP
Period		100 seconds

토폴로지 구성은 그림 7과 같으며 Cross Traffic 노드는 UDP Flow를 이용하여 혼잡상황을 발생시켰다. Base Station과 Destination 노드 사이에 무선링크의 Error Model을 적용하여 무선망에서의 전송오류에 따른 TCP 성능을 실험하였다. Error Model은 Multi State Error Model로서 오류가 일어나는 상태가 오류율에 따라 여러 가지가 존재하는 모델이다. 각각의 상태는 상태 지속 시간과 상태가 변할 확률에 따라 상태 전이를 하게 된다.

2. 시뮬레이션 결과 및 분석

가. 제안한 알고리즘의 확실한 무선손실 구분

Veno는 단지 N 과 β 에 의하여 무선손실과 혼잡손실을 구분하기 때문에 확실한 무선손실을 판단하지 못한다. 또한 무선손실의 판단 후에 1/5의 전송률을 감소하여 불필요한 대역폭의 낭비를 가져온다. Enhanced Veno^[17]는 무선손실 발생 간격 내에서 발생하는 혼잡손실을 측정하여 확실한 무선손실을 구분하여 전송률을 유지한다. 확실한 무선손실의 판정율을 그림 8에 나타내었다. Enhanced Veno는 무선손실의 시간간격이 일정하지 않을 경우 정확한 망의 상황을 판단하지 못한다. 제안한 알고리즘은 Traffic의 양이 증가함에 따라 확실

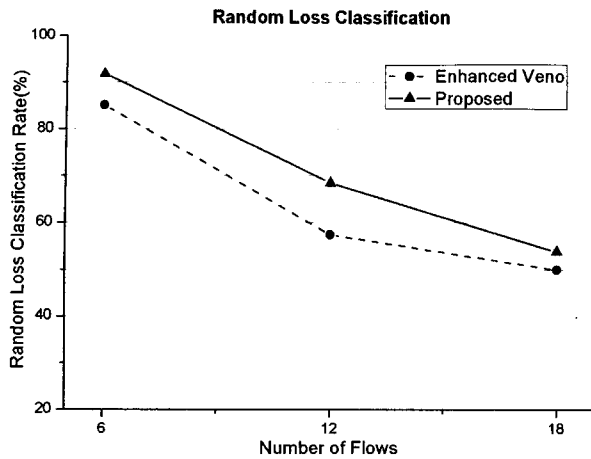


그림 8. 무선 손실 판별을 비교
Fig. 8. Compare classification rate of random loss.

한 무선손실의 판정율이 감소하였지만, 기존의 알고리즘에 비해 비교적 높은 판정율을 보여준다. 확실한 무선손실의 판정 중 시뮬레이션에서는 오판이 없었다.

나. 제안한 알고리즘의 처리량 비교

그림 9는 에러율을 0.1%로 고정하고 병목링크의 대역폭을 조절하면서 제안한 TCP와 TCP Westwood, TCP Veno의 처리량을 비교한 것이다. 대역폭이 5Mbps 미만일 경우 무선망에 적합하게 제안된 Veno와의 처리량 차이가 거의 없으나, 대역폭이 증가할수록 제안한 TCP와의 처리량 차이가 커지는 것을 알 수 있다. 이것은 병목 링크의 대역폭이 증가할수록 혼잡에 의한 패킷 손실은 감소하고 무선 에러에 의한 임의적인

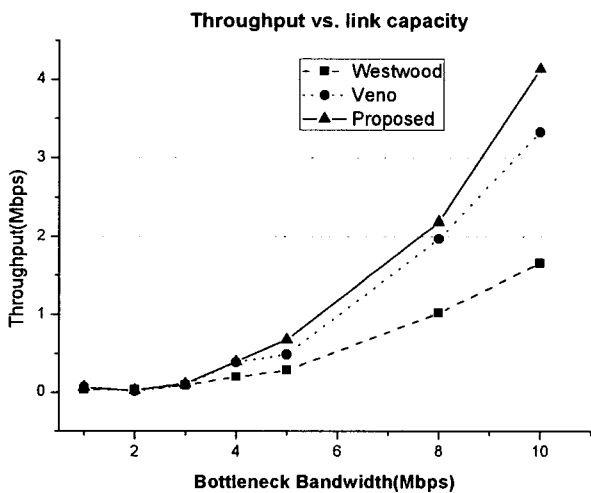


그림 9. 대역폭에 따른 TCP 처리량, 에러율 0.1%
Fig. 9. Throughput versus Link capacity, error rate 0.1%.

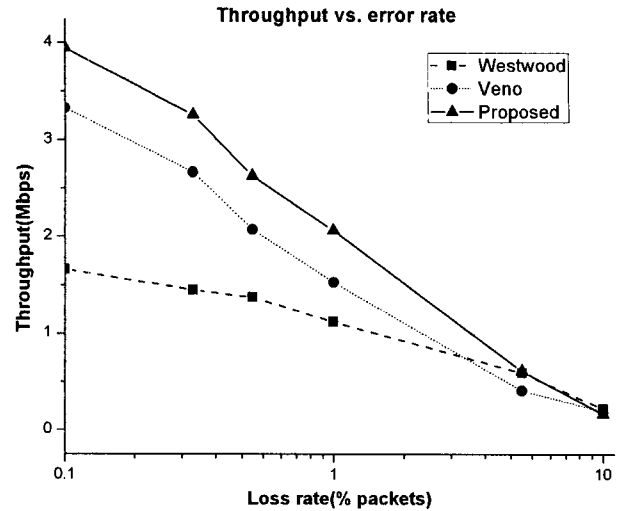


그림 10. 패킷 에러율에 따른 TCP처리량, 대역폭 10Mbps
Fig. 10. Throughput vs. error rate, Link capacity 10Mbps.

손실이 증가하여 무선 손실에 대한 대처 방법에 따라 처리량이 결정되기 때문이다.

임의적인 손실이라고 판단하면 1/5의 전송률을 감소하는 Veno에 비해 제안한 TCP는 확실한 무선 손실 판단 시 전송률을 그대로 유지하기 때문에 대역폭이 증가함에 따라 처리량이 높아진다.

그림 10은 무선 링크에서의 에러율의 변화에 따른 TCP의 처리량을 나타낸다. 병목 링크의 대역폭은 10Mbps로 고정하였다. 시뮬레이션 결과를 통하여 모든 구간에서 제안한 TCP의 처리량이 높은 것을 알 수 있다. 이것은 제안한 TCP가 망의 상황을 더 정확하게 판단하여 전송률을 조절하기 때문이다. 즉, 에러율이 낮은 망의 상황에서는 측정된 가용대역폭으로 전송률을 조절하고 에러율이 높은 망의 상황에서는 congestion loss rate 알고리즘을 적용하여 불필요한 전송률 감소를 하지 않는다. 모든 TCP들의 처리량은 에러율이 1% 이상 일 때 급격히 감소하며 이 결과로 무선망의 특성이 가지는 높은 BER은 TCP 성능에 큰 영향을 가져온다는 것을 알 수 있다.

다. 제안한 알고리즘의 성능 비교

그림 11과 그림 12는 Veno와 제안한 TCP의 시간에 따른 congestion window를 측정된 것이다. Cross traffic 노드의 수 n은 6, 대역폭은 10Mbps, 에러율은 0.05%로 고정하였다. 제안한 TCP는 Veno에 비해 cwnd의 변화가 적고 전체적으로 높은 값에 머물러 있다. Veno는 패킷 손실을 감지할 때마다 일정 배수만큼

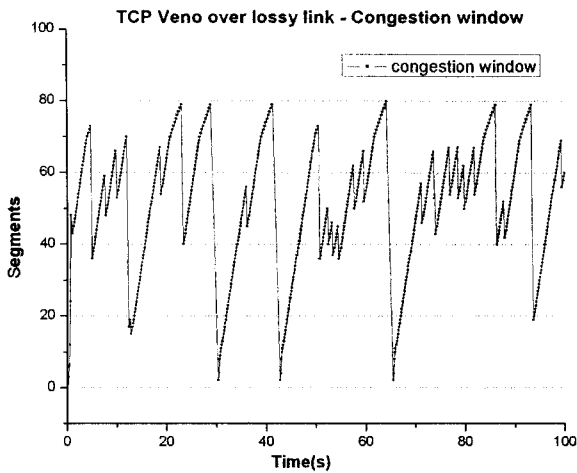


그림 11. TCP VenO의 congestion window
 Fig. 11. Congestion window of TCP VenO.

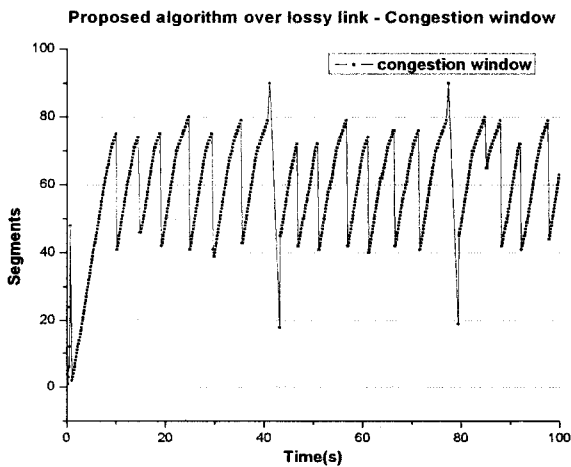


그림 12. 제안한 TCP의 congestion window
 Fig. 12. Congestion window of proposed TCP.

cwnd를 줄이지만, 제안한 TCP는 가용 대역폭을 측정하고 확실한 무선 손실을 구분하여 망의 상황 판단에 따라 동작하기 때문에 안정된 전송률을 보여준다. 또한 timeout 발생 시 전송률을 현저히 낮추는 VenO와 다르게 제안한 TCP는 측정된 가용대역폭만큼 전송률을 신속히 복구한다.

V. 결 론

본 논문에서는 유무선 혼합 망 환경에서 망의 상황을 판단하여 패킷 손실의 원인을 식별함으로써 확실한 무선 손실의 구분으로 전송률을 과도하게 줄이는 현상을 방지하고 가용대역폭의 측정을 통해 처리량을 높이는 알고리즘을 제안하였다. 성능 분석을 위하여 NS-2를 이용한 시뮬레이션을 통해 TCP VenO, TCP Westwood

와 성능을 비교하였다.

제안한 알고리즘은 망의 상황을 판단하여 패킷 손실의 원인이 혼잡에 의한 것이라고 식별되면 가용 대역폭을 측정하여 전송률을 조절한다. 무선 링크의 어려움이 낮은 망의 상황에서 기존의 WTCP에 비해 높은 처리량을 보여주었는데, 이는 혼잡에 의한 패킷 손실로 인해 전송률을 과도하게 줄이지 않고 사용가능한 대역폭을 측정함으로써 전체적으로 안정된 망의 상태를 유지하기 때문이다.

대역폭이 높은 망의 상황에서는 혼잡에 의한 패킷 손실은 감소하고 무선 상황에 따른 패킷 손실이 증가하게 되어 무선 손실에 대한 대처 방법에 따라 처리량이 결정된다. 제안한 알고리즘은 혼잡이 심하지 않은 망의 상황에서 발생한 패킷 손실에 대해서는 확실한 무선 손실을 구분하여 전송률을 줄이지 않고 유지하기 때문에 대역폭의 증가에 따라 처리량이 증가하였다.

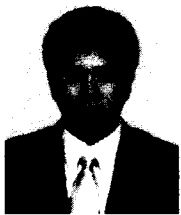
또한, TCP 계산량에 있어서 큰 오버헤드 없이 가용 대역폭을 측정하고 확실한 무선 손실을 구분하여, 고정된 비율로 전송률을 감소하는 VenO에 비해 효율적으로 대역폭을 사용하고 전체적으로 안정된 전송률을 보여주었다.

참 고 문 헌

- [1] W. Stallings, "Wireless Communication & Networks", Prentice Hall, 2004.
- [2] Bakre, A. Badrinath, "I-TCP : indirect TCP for mobile hosts Distributed Computing Systems", Proceedings of the 15th International Conference on, pp.136-143, 1995.
- [3] A. Chockalingan, M. Zorzi, and V. Tralli, "Wireless TCP performance with link layer FEC/ARQ", in Proc. Of ICC, pp.1212-1216, 1999.
- [4] C. F. Chiasserini and M. Meo, "Improving TCP over wireless through adaptive link layer setting", In IEEE GLOBECOM, Symposium on Internet Performance(IPS 2001), San Antonio, TX, November 2001.
- [5] Saverio Mascolo, Claudio Casetti, Mario Gerla, M. Y. Sanadidi, Ren Wang, "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", ACM SIGMOBILE, 2001.
- [6] M. Gerla, M. Y. Sanadidi, R. Wang, A. Zanella, C. Casetti, S. Mascolo, "TCP Westwood: Congestion Window Control Using Bandwidth Estimation", In Proceedings of IEEE Globecom 2001, pp 1698-1702. San Antonio, Texas, USA,

- November 2001.
- [7] Claudio Casetti, Mario Gerla, Saverio Mascolo, M.Y. Sansadidi, and Ren Wang, "TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks" In Wireless Networks Journal 8, pp.467-479, 2002.
- [8] Cheng Peng Fu, Soung C. Liew, "TCP Veno: TCP Enhancement for Transmission Over Wireless Access Networks", IEEE JOURNAL, Feb. 2003.
- [9] W. Stevense, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithm", RFC 2001, Jan. 1997, URL: <http://www.ietf.org/rfc/rfc3041.txt/>
- [10] V. Jacobson, "Congestion Avoidance and Control.", in Proc. SIGCOMM '88, pp.314-329 Stanford, CA,
- [11] V. Jacobson, "Modified TCP Congestion Avoidance Algorithm.", Apr. 1990.
- [12] M. Allman and V. Paxson, "TCP Congestion Control.", RFC 2581, Apr. 1999.
- [13] S. Floyd and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm.", RFC 2582, Apr. 1999.
- [14] M. Mathis, J. Mahdavi et al. "TCP Selective Acknowledgement Options.", RFC 2018, Oct. 1996.
- [15] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance.", in Proc. SIGCOMM '94, pp.24~35, London, U.K., Oct. 1994.
- [16] K. Brown and S. Singh M~ TCP : TCP for Cellular Networks ACM SIGCOMM Computer Communication Review", Volume 27 Issue 5, October 1997.
- [17] Bin Zhou, Cheng Peng Fu, "An Enhancement of TCP Veno over Light~Load Wireless Networks", IEEE COMMUNICATIONS LETTERS, Volume 10, NO. 6, June 2006.
- [18] ns~2 network simulator (ver 2.6). LBL, URL : <http://www.isi.edu/nsnam/ns>

 저 자 소 개



조 남 진(학생회원)
 2008년 한국항공대학교 정보통신
 공학과 졸업.(공학사)
 2008년 3월~현재 한국항공대학교
 정보통신공학과 석사과정.
 <주관심분야 : TCP/IP, Digital
 Home Network, Wireless Sensor
 Network>



이 성 창(정회원)
 1976년~1983년 경북대학교
 전자공학과 졸업.(공학사)
 1983년~1985년 한국과학기술원
 전기 및 전자공학과 석사
 졸업.(공학석사)
 1985년~1987년 한국과학기술원
 시스템공학센터
 1987년~1991년 Texas A&M University 졸업.
 (공학박사)
 1992년~1993년 한국전자통신연구원
 1993년~현재 한국항공대학교 항공전자 및
 정보통신공학부 교수
 <주관심분야 : Home Network, Ubiquitous 융합
 기술, Fixed Mobile Convergence>