

TCP NJ+: 높은 BER에 강인한 패킷 손실 원인별 처리기반 전송방식

TCP NJ+: Packet Loss Differentiated Transmission Mechanism Robust to High BER Environments

김 정 래* 이 유 호** 추 현 승***
Jungrae Kim Youho Lee Hyunseung Choo

요 약

무선 네트워크에서 네트워크 혼잡과 bit-error-rate (BER)에 의한 패킷 손실을 구분하는 기법과 현재 가용 대역폭을 예측하여 전송량을 조절하는 기법은 무선 TCP의 성능을 크게 향상시킨다. 네트워크 혼잡과 BER에 의한 패킷 손실을 명시적으로 구분하는 TCP New Jersey는 현재까지 제안된 무선 TCP 기법 중 가장 높은 성능을 보이지만 무선 링크 어려움이 높아지면 전송률이 크게 떨어지는 문제와 가용 대역폭 예측 알고리즘이 백그라운드 트래픽 패턴에 의존적인 문제가 있다. 또한 패킷 손실을 감지하여 패킷 손실의 원인을 판별하는 기법으로 원인을 구분하여 감소된 전송량을 회복하는 기법이 효율적이지 못하다. 본 논문에서는 TCP New Jersey의 문제점을 보완한 TCP NJ+를 제안한다. TCP NJ+는 향상된 전송량 회복 기법으로 BER에 의한 일반적인 패킷 손실뿐만 아니라 retransmission timeout이 발생하여도 보다 높은 전송률을 보장한다. 제안된 TCP NJ+는 NS-2 네트워크 시뮬레이터를 이용하여 성능평가 한 결과 5%의 높은 무선 링크 어려움을 갖는 네트워크에서는 TCP New Jersey에 비해 19%, TCP Westwood에 비해 57%의 성능향상을 보인다.

Abstract

Transmission mechanisms that include an available bandwidth estimation algorithm and a packet loss differentiation scheme, in general, exhibit higher TCP performance in wireless networks. TCP New Jersey, known as the best existing scheme in terms of goodput, improves wireless TCP performance using the available bandwidth estimation at the sender and the congestion warning at intermediate routers. Although TCP New Jersey achieves 17% and 85% improvements in goodput over TCP Westwood and TCP Reno, respectively, we further improve TCP New Jersey by exploring improved available bandwidth estimation, retransmission timeout, and recovery mechanisms. Hence, we propose TCP New Jersey PLUS (shortly TCP NJ+), showing that under 1% packet loss rate, it outperforms 3% by TCP New Jersey and 5% by TCP Westwood. In 5% packet loss rate, a characteristic of high bit-error-rate wireless network, it outperforms other TCP variants by 19% to 104% in terms of goodput even when the network is in bi-directional congestion.

☞ keywords: TCP, TCP Jersey, 혼잡제어

1. 서 론

유선 네트워크를 기반으로 구현된 TCP

* 준 회 원 : 성균관대학교 일반대학원 전자전기컴퓨터공학과 재학 (석사)

witjung@ece.skku.ac.kr

** 정 회 원 : 대구한의대학교 인터넷정보학과 교수

youho@sdu.ac.kr

*** 종신회원 : 성균관대학교 정보통신공학부 부교수

choo@ece.skku.ac.kr

[2007/03/02 투고 - 2007/03/19 심사 - 2007/05/18 심사완료]

(Transmission control protocol) [1]는 종단 간 신뢰성 있는 데이터 전송을 보장하는 연결 지향형 전송 프로토콜로서 셀룰러 네트워크, 무선랜, 이동 컴퓨팅과 같은 무선 네트워크에서 심각한 성능 저하를 보인다. 무선 네트워크는 네트워크의 혼잡에 의해 발생하는 패킷 손실 이외에 링크의 특성에 따른 높은 비트 어려움 (BER), 신호 감쇠 (Fading), 일시적인 연결단절 (Handoff) 등 다양한 원인으로 패킷 손실이 발생할 수 있다. 그러므로

모든 패킷 손실을 네트워크의 혼잡으로 가정하여 패킷 손실이 발견되면 전송량을 감소시키는 유선 TCP의 혼잡제어 [2] 기법은 무선 네트워크에서 TCP의 성능을 떨어뜨리는 주요 원인이 된다. 따라서 무선 TCP는 패킷 손실의 원인이 다양한 무선 네트워크의 특성을 지능적으로 반영하는 혼잡제어 기법이 필요하다.

최근 무선 네트워크 기반 기술의 발전과 더불어 무선 TCP의 성능 향상을 위한 연구가 활발히 진행되고 있다 [3]. 그 중 중단 간 접근 방법에서는 현재 네트워크의 가용 대역폭을 예측하여 이를 기반으로 혼잡제어를 수행하는 선형(Proactive) 혼잡제어 방식 (예를 들면, TCP Westwood [4], TCP Jersey [5], TCP New Jersey [6])과 같이 무선 환경에서 높은 전송률을 보이는 다양한 TCP가 제안되었다.

그러나 가용 대역폭 예측 방식의 TCP는 백그라운드 트래픽으로 네트워크 상태가 악화되면 가용 대역폭 예측을 적절히 하지 못하여 전송량을 크게 감소시켜 결과적으로 전송률이 떨어진다. 더불어 TCP New Jersey는 패킷 손실 발생 후 패킷 손실의 원인에 따라 다른 회복 동작을 수행하는 전송량 회복 기법이 무선 링크의 에러율이 높아지면 적절히 동작하지 못하는 문제가 있다. 그러므로 본 논문에서는 TCP New Jersey 수정하여 보다 효과적으로 전송량을 회복하여 높은 전송률을 보장하는 전송량 회복 기법인 TCP NJ+를 제안한다. TCP NJ+는 TCP New Jersey에 비해 5% 무선 링크 에러율에서 19%의 성능향상을 보인다.

본 논문은 다음과 같이 구성된다. 2장에서 관련 연구를 소개하고, 3장에서 TCP NJ+ 기법을 설명한다. 4장에서는 다양한 네트워크 상태에 따른 시뮬레이션에 대한 결과를 보여주며 이를 분석하고, 5장은 결론으로 끝맺음을 한다.

2. 관련 연구

유선 네트워크를 기반으로 구현된 대표적인

TCP는 TCP Reno [8]가 있다. 중단 간 접근 방법으로 구현된 무선 TCP로 TCP Westwood, TCP Jersey, TCP New Jersey가 있다.

2.1 TCP Reno

TCP Reno는 널리 사용되는 표준 TCP이며, 유선 네트워크에서 효과적인 데이터 전송을 보장한다. TCP Reno의 빠른 회복 (Fast recovery) 알고리즘은 패킷 손실이 발생된 경우에 빠른 재전송 (Fast retransmit) [7] 알고리즘을 수행한 후 슬로우 스타트(Slow start) 단계가 아닌 혼잡회피 (Congestion avoidance) 단계로 바로 진입하여 감소된 전송량을 빠르게 회복하는 알고리즘이다.

빠른 회복 알고리즘은 하나의 혼잡 윈도우 (Congestion window: cwnd)당 하나의 패킷 손실로 인해 감소된 전송량을 회복하는데 효과적인 알고리즘이다. 그러나 TCP Reno는 하나의 혼잡 윈도우에서 다중 패킷 손실이 발생하면 연속적으로 빠른 회복 알고리즘을 수행함으로써 전송량을 연속적으로 감소시키는 문제를 갖는다. 즉, 패킷 손실이 많이 일어나는 무선 네트워크에서는 불필요한 전송량의 감소를 가져온다.

2.2 TCP Westwood

TCP Westwood는 선형 혼잡제어 방식의 무선 TCP로 송신자가 받은 ACK패킷의 도착 비율을 바탕으로 현재 네트워크의 가용 대역폭을 예측하여, 패킷 손실이 발생하면 예측된 대역폭 값을 기반으로 전송량을 조절한다. 송신자가 현재 네트워크의 상태를 기반으로 전송량을 조절하기 때문에 네트워크 이용 효율이 높아져 전송률의 향상을 가져온다.

그러나 패킷 손실의 원인에 대한 판단 없이 패킷 손실이 발생할 때 마다 전송량을 감소하기 때문에 링크의 에러율이 높아지면 성능이 떨어지며, 전송 중인 네트워크의 트래픽 정도에 따라 가용

대역폭 예측의 정확도가 영향을 받기 때문에 항상 높은 전송률을 보장하지 못한다.

2.3 TCP Jersey와 New Jersey

TCP Jersey는 가용 대역폭 예측 알고리즘과 Congestion Warning (CW) 기법을 이용하여 무선 네트워크에서 TCP의 성능 향상을 도모한다. 가용 대역폭 예측 알고리즘은 송신자가 ACK를 받은 시간 간격을 기준으로 그 시간당 보내진 전송량을 판단하여 현재 네트워크의 가용 대역폭을 예측하여 그 예측 값과 혼잡과 BER을 구별하는 CW 기법을 바탕으로 전송량을 조절한다.

TCP New Jersey는 TCP Jersey의 가용 대역폭 예측 알고리즘과 슬로우 스타트 경계점 (sssthreshold) 조절 방법을 수정하였다. TCP New Jersey의 가용 대역폭 예측 알고리즘은 수신자가 데이터를 받은 시간 간격이 기준이다. TCP New Jersey는 ACK가 돌아오는 방향에 백그라운드 트래픽이 있을 경우 보다 정확한 가용 대역폭 예측 값을 보장한다. 또한 가용 대역폭 예측 값을 바탕으로 sssthreshold 값을 동적으로 조절한다.

그러나 TCP New Jersey는 백그라운드 트래픽 패턴에 따라 전송률이 떨어지는 문제가 있으며, 패킷 손실 발생 시 감소된 전송량 회복을 효과적으로 하지 못하여 BER에 의한 패킷 손실이 많아지면 결과적으로 전송률이 떨어지는 문제가 있다.

3. TCP NJ+

TCP New Jersey는 가용 대역폭 예측 시 백그라운드 트래픽 패턴에 따라 전송률이 떨어지는 문제가 있다. 또한 패킷 손실이 발생 하거나 RTO (Retransmission timeout)가 발생 했을 때 전송량을 회복하는 기법이 효과적으로 전송량을 회복하지 못한다. TCP NJ+는 앞에 언급한 TCP New Jersey의 두 가지 문제점을 수정하여 높은 전송률을 보장한다.

3.1 향상된 가용 대역폭 예측 알고리즘

TCP Jersey와 TCP New Jersey는 다음 식 1에 표현된 공식으로 현재 가용 대역폭을 예측한다.

$$R_n = \frac{RTT \times R_{n-1} + L_n}{(t_n - t_{n-1}) + RTT} \quad (1)$$

TCP Jersey에서 R_n 은 n 번째 ACK패킷이 t_n 시간에 송신자에게 도착했을 때 예측한 대역폭이며, t_{n-1} 은 이전 ACK패킷이 송신자에게 도착한 시간이다. L_n 은 n 번째 ACK패킷에 대한 데이터 세그먼트의 크기이며, RTT 는 종단 간 왕복시간 (Round trip time)이다. 그러나 TCP New Jersey는 TCP Jersey와 마찬가지로 식 1로 현재 가용 대역폭을 예측하지만 TCP New Jersey에서의 t_n 은 타임 스탬프 옵션 [9]을 이용하여 수신자가 n 번째 데이터 세그먼트를 받은 시간이며, t_{n-1} 은 수신자가 $n-1$ 번째 데이터 세그먼트를 받은 시간이다.

TCP Jersey의 가용 대역폭 예측 알고리즘은 ACK패킷이 돌아오는 역방향에 백그라운드 트래픽이 있는 상황과 같이 네트워크의 환경이 악화 되면 송신자가 ACK패킷을 제대로 받지 못함으로써 정확한 예측을 보장하지 못하여 전송률이 떨어진다. TCP New Jersey는 TCP Jersey와 반대로 데이터 세그먼트가 가는 순방향의 네트워크 환경이 악화 될 경우 전송률이 떨어진다. 위에서 언급 하대로 TCP Jersey와 New Jersey는 백그라운드 트래픽 진행방향에 따라 높은 전송률을 보장하지 못하는 문제가 있다.

따라서 TCP NJ+의 가용 대역폭 예측 알고리즘은 식 2를 이용하여 t_n 시간에 송신자가 n 번째 ACK패킷을 받은 시간을 기준으로 첫 번째 가용 대역폭을 예측한다.

$$R_{s_n} = \frac{RTT \times R_{s_{n-1}} + L_n}{(t_n - t_{n-1}) + RTT} \quad (2)$$

식 2에서 R_{s_n} 은 송신자가 n 번째 ACK패킷을 t_n 시간에 받았을 때 예측한 가용 대역폭 값이며, $R_{s_{n-1}}$ 은 송신자가 $n-1$ 번째 ACK패킷을 받았을 때 예측한 가용 대역폭 값이다. 나머지는 TCP New Jersey와 동일하다. 또한, 식 3을 이용하여 수신자에서 n 번째 데이터 세그먼트를 받은 시간을 기준으로 두 번째 가용 대역폭을 예측한다.

$$Rr_n = \frac{RTT \times Rr_{n-1} + L_n}{(t_n - t_{n-1}) + RTT} \quad (3)$$

식 3에서 Rr_n 은 수신자에 n 번째 데이터 세그먼트가 t_n 시간에 도착했을 때 예측한 가용 대역폭 값이며, Rr_{n-1} 은 수신자에 $n-1$ 번째 데이터 세그먼트가 도착했을 때 예측한 가용 대역폭 값이다. 위의 식 2와 식 3으로 산출된 R_{s_n} 과 Rr_n 을 비교하여 큰 값으로 현재 가용 대역폭 예측값을 산출한다.

결론적으로 TCP NJ+는 두 가지 방법으로 가용 대역폭을 예측한 후 그 값의 최대값으로 최종 가용 대역폭을 산출하기 때문에 백그라운드 트래픽 패턴에 민감한 TCP Jersey나 TCP New Jersey와 달리 트래픽 진행 방향성에 관계없이 항상 최적의 전송률을 보장한다. 다음의 그림1은 TCP NJ+의 가용 대역폭 예측 알고리즘이다.

```
(ACK packet received by sender)
if (timestamp)
  bwe_sender_n = (rtt * double(bwe_sender_{n-1}) + seg_size)
                  / (st_n - st_{n-1} + rtt);
  bwe_receiver_n = (rtt * double(bwe_receiver_{n-1}) + seg_size)
                   / (rt_n - rt_{n-1} + rtt);
  if (bwe_sender_n > bwe_receiver_n)
    current_bwe = bwe_sender_n;
  else
    current_bwe = bwe_receiver_n;
  end if
end if
```

(그림 2) 향상된 가용대역폭 예측 알고리즘

3.2 향상된 RTO 기법

TCP 송신자는 타임아웃 시간만큼 ACK패킷을 받지 못하면 패킷 손실이 일어났으며 현재 네트워크의 상태가 악화되었다고 판단하여 $ssthreshold$ 를 현재 $cwnd$ 의 1/2로 설정하고, $cwnd$ 는 1로 설정하여 전송량을 감소시킨다. 즉, 유선 TCP인 TCP Reno는 AIMD알고리즘을 기반으로 RTO기법이 동작하여 RTO발생 확률이 높은 무선 네트워크에서 큰 폭으로 전송량을 감소시킨다.

TCP New Jersey는 가용 대역폭 예측값을 바탕으로 다음 식 4를 이용하여 최적의 전송 윈도우(Optimized Window: $ownd$)를 구한다.

$$ownd_n = \frac{RTT \times R_n}{segment\ size} \quad (4)$$

n 번째 패킷에 의해 RTO가 발생했을 때, 혼잡에 의한 RTO이면 $cwnd$ 를 1로, $ssthreshold$ 를 $ownd_n$ 으로 설정하며, BER에 의한 RTO이면 $cwnd$ 와 $ssthreshold$ 를 $ownd_n$ 로 설정한다. 그러나 BER에 의한 RTO는 혼잡에 의한 RTO와 달리 현재 네트워크의 링크 상태가 일시적으로 악화된 것으로서 링크의 대역폭은 충분히 사용할 수 있으므로 $cwnd$ 를 네트워크 상태 악화로 인해 작은 값으로 산출된 현재의 $ownd_n$ 로 설정하는 것은 부적절하다. 또한 네트워크의 상태가 갑자기 크게 악화된 경우에 $cwnd$ 를 적절히 설정하지 못하는 문제로 인해 전송률이 크게 떨어진다.

따라서 TCP NJ+는 번째 패킷에 의한 RTO 발생 시 혼잡에 의한 RTO와 BER에 의한 RTO를 판단하여 혼잡에 의한 RTO이면 $cwnd$ 를 1로 $ssthreshold$ 를 $ownd_n$ 으로 설정한다. 그러나 BER에 의한 RTO는 $ssthreshold$ 를 $ownd_n$ 으로 설정하고, $cwnd$ 는 $ownd_n$ 뿐만 아니라 $ownd_{n-1}$ 을 고려하여 다음 그림 2에서 표현된 알고리즘으로 값을 설정한다.

```

if (RTO expired)
  if (Congestion Warning)
    /*if RTO due to congestion*/
    cwnd = 1;
    ssthresh = owndn;
  else if
    /*if RTO due to BER*/
    cwnd = (owndn + owndn-1) / 2;
    ssthresh = owndn;
  end if
end if

```

(그림 3) 향상된 RTO 기법

$ownd_{n-1}$ 은 네트워크 상태가 악화되기 이전의 가용 대역폭 예측값을 기반으로 산출된 값이므로 항상 $ownd_n$ 보다 크다. 즉, $ownd_{n-1}$ 과 $ownd_n$ 의 평균값은 $ownd_n$ 보다 크기 때문에 항상 높은 $ownd$ 를 보장하며 결과적으로 높은 $cwnd$ 를 얻는다. 그러므로 TCP NJ+는 RTO에 의해 감소된 $cwnd$ 의 회복속도를 높여 감소된 링크의 이용효율을 높임으로서 TCP New Jersey에 비해 전체적인 전송률을 향상시킬 수 있다. 또한 높은 링크 에러율을 갖는 무선 네트워크에서는 RTO가 발생 될 확률 역시 높기 때문에 향상된 RTO기법으로 인해 다른 무선 TCP에 비해 높은 전송률을 보장한다.

3.3 향상된 전송량 회복 기법

TCP New Jersey는 CW 기법을 이용하여 혼잡에 의한 패킷 손실과 BER에 의한 패킷 손실을 판단한다. 3개의 중복 ACK를 받으면 에러원인에 따라 다음과 같은 동작을 수행한다. 첫째, 혼잡에 의한 에러이면 $ssthresh$ 를 $ownd_n$ 값으로, $cwnd$ 가 $ssthresh$ 보다 작으면 현재의 $cwnd$ 값으로, $cwnd$ 가 $ssthresh$ 보다 크면 $ownd_n$ 값으로 설정함으로써 rate control 기법을 수행한다. 둘째, BER에 의한 패킷 손실이면 현재 $cwnd$ 와 $ssthresh$ 를 유지한다. 즉, 혼잡과 BER을 구분함에도 불구하고 BER에 의한 패킷 손실일 경우 $cwnd$ 를 조절하는 동작이 적절하지 못하다. TCP New Jersey는

혼잡회피 과정에서 일반적인 ACK를 받으면 $cwnd$ 를 증가시키나 3번째 중복 ACK를 받았을 때는 혼잡회피 과정임에도 불구하고 $cwnd$ 를 증가시키지 않는다. 이는 BER에 의한 패킷 손실이 많이 발생함에 따라 3개의 중복 ACK를 받았을 때 증가되지 않는 $cwnd$ 의 손실이 더 커짐을 의미한다.

TCP NJ+는 TCP New Jersey의 이런 문제점을 개선하여 다음 그림 3과 같이 전송량 회복 기법을 수행한다.

```

if (3 dupack received by sender)
  if (Congestion Warning)
    /*if packet loss due to congestion*/
    ssthresh = owndn;
    if (ssthresh < cwnd)
      cwnd = owndn;
    end if
  else if
    /*if packet loss due to BER*/
    ssthresh = owndn;
    cwnd = cwnd + 1;
  end if
end if

```

(그림 4) 향상된 전송량 회복 기법

3개의 중복 ACK를 받으면 CW기법으로 혼잡에 의한 패킷 손실인지 BER에 의한 패킷 손실인지를 판단하여 다음의 동작을 수행한다. 첫째, 혼잡에 의한 패킷 손실이면 $ssthresh$ 를 $ownd_n$ 값으로, $cwnd$ 가 $ssthresh$ 보다 작으면 현재의 $cwnd$ 값으로, $cwnd$ 가 $ssthresh$ 보다 크면 $ownd_n$ 값으로 설정한다. 둘째, BER에 의한 패킷 손실이면 $ssthresh$ 를 $ownd_n$ 값으로 설정하고, $cwnd$ 는 현재 $cwnd$ 에서 하나 증가시킨다. 이는 3번째 중복 ACK를 받았을 때 $cwnd$ 를 증가시키지 않는 TCP New Jersey의 문제점을 개선한 기법이며, $cwnd$ 를 하나 증가시키는 이유는 2번째 중복 ACK 수신까지는 $cwnd$ 를 증가시키고 3번째 중복 ACK 수신시 빠른 회복 알고리즘을 수행하여 $cwnd$ 가 증가되지 않으므로 이때 증가되지 않는 하나의 $cwnd$ 손실에 대한 보정을 수행한 것이다.

BER에 의한 패킷 손실은 네트워크의 대역폭을 더 이용할 수 있지만 무선 링크의 특성으로 패킷

손실이 발생한 것이므로 이런 경우 *cwnd*의 적절한 조절은 전체 전송률 향상을 가져온다. 그러므로 TCP NJ+는 BER에 의한 패킷 손실이 많은 경우 다른 무선 TCP 보다 더 높은 *cwnd*를 보장하여 전송률을 향상시킬 수 있다.

4. 성능 평가

4.1 시뮬레이션 환경

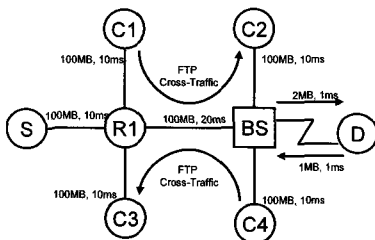
본 논문은 NS-2 시뮬레이터 [10]를 이용하여 TCP NJ+의 *goodput*을 측정한다. 파라미터는 표 1과 같이 대역폭, 패킷크기, 전송지연시간 큐 크기를 결정한다.

(표 1) 시뮬레이션 파라미터

대역폭		패킷크기	전송지연시간		큐크기
유선	무선		유선	무선	
100MB	2MB	762byte	10~20ms	1ms	20~200 (개)

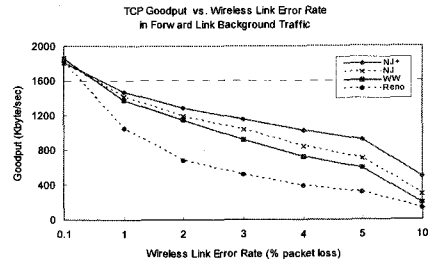
4.2 백그라운드 트래픽 패턴에 따른 성능평가

TCP NJ+의 가용 대역폭 예측 알고리즘은 백그라운드 트래픽에 관계없이 항상 높은 전송률을 보장한다. 다음 그림 4와 같은 토폴로지에서 백그라운드 트래픽 방향을 3가지 경우로 설정하고 무선 링크의 에러율을 0.1%, 1%, 2%, 3%, 4%, 5%, 10%로 변화시켜 TCP NJ+, TCP New Jersey, TCP Westwood, TCP Reno의 *goodput*을 평가한다.



(그림 5) 백그라운드 트래픽 패턴에 따른 시뮬레이션 토폴로지

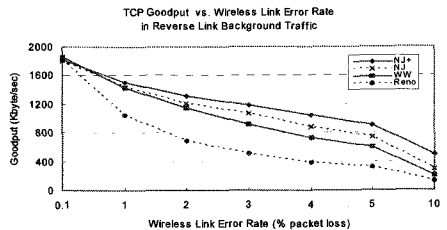
그림4의 토폴로지에서 백그라운드 트래픽이 노드C1에서 노드C2 방향으로 FTP 백그라운드 트래픽이 있을 때 (순방향 트래픽), 노드C4에서 노드C3 방향으로 FTP 백그라운드 트래픽이 있을 때 (역방향 트래픽), 양방향으로 FTP 백그라운드 트래픽이 있는 3가지 경우를 시뮬레이션 한다.



(그림 6) 순방향 백그라운드 트래픽이 있을 때 에러율에 따른 Goodput

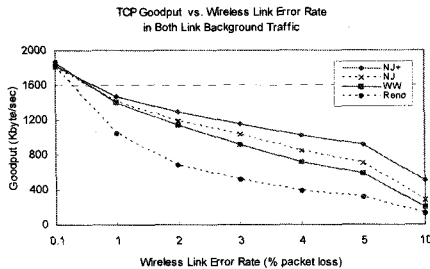
그림 5은 데이터가 가는 순방향 FTP백그라운드 트래픽이 있을 때 무선 링크 에러율에 따른 각 TCP들의 *goodput*을 나타낸다. TCP NJ+는 5%의 무선 링크 에러율에서는 TCP New Jersey에 비해 30%, TCP Westwood에 비해 55%의 성능향상을 보인다.

그림 6은 ACK가 돌아오는 역방향에 FTP백그라운드 트래픽이 있을 때 무선 링크 에러율에 따른 각 TCP들의 *goodput*을 나타낸다. 그림 8과 같이 TCP NJ+는 5%의 무선 링크 에러율에서는 TCP New Jersey에 비해 20%, TCP Westwood에 비해 45%의 성능향상을 보인다.



(그림 7) 역방향 백그라운드 트래픽이 있을 때 에러율에 따른 Goodput

그림 7는 데이터가 전송되는 방향과 ACK가 돌아오는 양방향에 FTP백그라운드 트래픽이 있을 때 무선 링크 에러율에 따른 각 TCP들의 goodput을 나타낸다. 그림 7와 같이 TCP NJ+는 5%의 무선 링크 에러율에서는 TCP New Jersey에 비해 31%, TCP Westwood에 비해 56%의 성능향상을 보인다. 시뮬레이션 결과 TCP NJ+는 백그라운드 트래픽 패턴에 관계없이 TCP New Jersey에 비해 항상 높은 전송률을 보이며, 무선 링크 에러율이 높아지면 더 높은 성능 향상을 보인다.



(그림 8) 양방향 백그라운드 트래픽이 있을 때 에러율에 따른 Goodput

5. 결론

본 논문에서는 TCP New Jersey의 문제점을 수정한 TCP NJ+의 다음 세가지 기법을 제안한다. 첫째, 가용 대역폭 예측 알고리즘은 두 가지 기준을 통해 가용 대역폭을 예측하므로 백그라운드 트래픽에 영향을 받지 않고 항상 높은 전송률을 보장한다. 둘째, BER에 의한 RTO가 발생 했을 때 이전 네트워크의 상태를 반영함으로써 cwnd 회복 속도를 높여 결과적으로 높은 전송률을 보장한다. 셋째, 전송량 회복 기법은 무선 링크 에러로 인한 패킷 손실이 발생했을 때 감소된 cwnd의 증가 속도를 높여 높은 전송률을 보장한다.

시뮬레이션 결과에서 무선 링크의 에러율이 높아지면 다른 TCP와 비교하여 성능 향상의 폭이 커짐이 나타난다. 백그라운드 트래픽이 있는 환경에서는 5%의 무선 링크 에러율에서 TCP New

Jersey에 비해 평균 27%, TCP Westwood에 비해 평균 52%의 높은 성능향상을 보인다. 결과적으로 TCP NJ+는 새롭게 제안된 기법을 통해 다른 TCP에 비해 링크 에러율이 높은 환경에서 더 높은 전송률을 보장한다.

Acknowledgement

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2007-(C1090-0701-0046)).

교신저자: 추현승.

참고 문헌

- [1] J. Postel, "Transmission control protocol," RFC 793, 1981.
- [2] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion control," RFC 2581, 1999.
- [3] Y. Tian, K. Xu, and N. Ansari, "TCP in wireless environments: problems and solutions," IEEE Radio Communications, vol. 43, no. 3, pp.27-32, Mar 2005.
- [4] C. Casetti, M. Gerla, S. Mascolo, M.Y. Sanadidi, and R. Wang, "TCP Westwood: bandwidth estimation for enhanced transport over wireless links," ACM/IEEE MobiCom, pp.287-297, Jul 2001.
- [5] K. Xu, Y.Tian, and N. Ansari, "TCP-Jersey for wireless IP communications," IEEE Journal of Selected Areas in Communications, vol. 22, issue 4, pp.747-756, May 2004.
- [6] K. Xu, Y. Tian, and N. Ansari, "Improving TCP performance in integrated wireless communications networks," Computer Networks, vol. 47, no. 2, pp.219-237, Feb 2005.
- [7] V. Jacobson, "Congestion avoidance and control," ACM SIGCOMM 1988, vol. 18,

pp.314-329, Aug 1988.

[8] W. Stevens, "TCP slow start, congestion avoidance, fast retransmit and fast recovery algorithms," RFC 2001, 1997.

[9] V. Jacobson, R. Braden, and D. Borman, "TCP extensions for high performance," RFC 1323, 1992.

[10] UCB/LBNL/VINT Network Simulator [Online]. Available: <http://www.isi.edu/nsnam/ns>

● 저 자 소 개 ●



김 정 래(Jungrae Kim)

2001년 인천대학교 컴퓨터공학과 졸업(학사)

2006년 ~ 현재 성균관대학교 일반대학원 전자전기컴퓨터공학과 재학 (석사)

관심분야 : 유/무선 TCP, 센서 네트워크

E-mail : witjung@skku.edu



이 유 호(Youho Lee)

1986년 성균관대학교 수학과 졸업(학사)

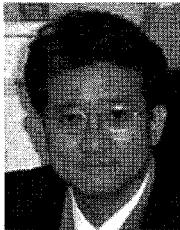
1989년 성균관대학교 일반대학원 수학과 졸업(석사)

1996년 성균관대학교 일반대학원 수학과 졸업(박사)

2002년 ~ 현재 대구한의대학교 인터넷정보학과 교수

관심분야 : 정보보호, 네트워크보안, 라우팅프로토콜

E-mail : youho@dhu.ac.kr



추 현 승(Hyunseung Choo)

1988년 성균관대학교 수학과 졸업(학사)

1990년 University of Texas 컴퓨터공학과 졸업(석사)

1996년 University of Texas 컴퓨터공학과 졸업(박사)

1997년 특허청 심사4국 컴퓨터심사담당관실(사무관)

1998년 ~ 현재 성균관대학교 정보통신공학부 부교수

2001년 ~ 현재 한국인터넷정보학회/한국시뮬레이션학회 이사

2004년 3월 ~ 2006년 8월 대통령직속 교육혁신위원회 전문위원

2004년 8월 ~ 현재 한국인터넷정보학회 논문지편집위원장

2005년 1월 ~ 현재 한국 건강보험심사평가원 전문위원

2005년 10월 ~ 현재 정보통신부ITRC 지능형HCI융합연구센터장,

정보통신공학부 컨버전스 연구소장

관심분야 : 유/무선/광 네트워크, 모바일컴퓨팅, 임베디드S/W, 그리드컴퓨팅

E-mail : choo@ece.skku.ac.kr