

# 빠른 손실 감지를 이용한 TCP NewReno 분석

정회원 김 동 민\*, 한 제 찬\*, 김 석 규\*, 임 차 식\*, 종신회원 이 재 용\*

## Analysis of TCP NewReno using rapid loss detection

Dong min Kim\*, Je chan Han\*, Seog gyu Kim\*, Cha sik Leem\*, Jai yong Lee\* *Regular Members*

### 요 약

무선 통신 환경은 WiBro 등과 같은 새로운 무선 통신 기술들을 이용함에 따라 고속 인터넷 접속이 가능하도록 빠르게 변화하고 있다. 이에 따라 무선 구간에서 TCP(Transmission Control Protocol)를 이용한 신뢰성 있는 데이터 전송도 늘어날 전망이다. TCP는 유선상에서 사용될 목적으로 디자인되었기 때문에 유선과 다른 환경 특히 무선에서 사용될 때에는 비 혼잡 손실에 의해 성능 저하를 겪게 된다. 특히 RTO(Retransmission TimeOut)은 TCP의 성능에 많은 영향을 미친다. 본 논문에서는 기존 연구를 바탕으로 손실 복구 확률을 분석하고 모의실험 결과를 통해 제안 알고리즘이 fast recovery 과정 중에 발생한 패킷 손실을 빠르게 감지하여 RTO없이 복구함으로써 성능 저하를 줄일 수 있음을 보인다.

**Key Words :** TCP, fast recovery, rapid loss detection, recovery probability, wireless networks

### ABSTRACT

Wireless communication environment is changing rapidly as we use new wireless communication technology such as WiBro to access high speed Internet. As a result, reliable data transmission using TCP is also expected to increase. Since TCP assumes that it is used in wired network, TCP suffers significant performance degradation over wireless network where packet losses are related to non-congestion loss. Especially RTO imposes a great performance degradation of TCP. In this paper, we analyze the loss recovery probabilities based on previous researches, and use simulation results of our algorithm to show that it prevents performance degradation by quickly detecting and recovery losses without RTO during fast recovery.

### I. 서 론

인터넷 서비스가 유선 네트워크에서 무선 네트워크로 영역을 확장해 나감에 따라, 무선 사용자의 데이터 트래픽이 계속 증가할 것으로 예상된다. 전송 프로토콜로는 UDP(User Datagram Protocol)와 TCP(Transmission Control Protocol)가 사용되는데, TCP는 유선에서 사용됨을 기본 가정으로 설계되었다. 유선에서 패킷 손실의 원인은 네트워크 혼잡에 의해 중간 라우터들의 큐에서 버려지는 것과 전송

손실로 나눌 수 있는데, 유선에서는 전송손실이 거의 발생하지 않는다. 그렇기 때문에 TCP는 모든 패킷 손실의 원인을 네트워크 혼잡에 의한 것으로 보고, 손실된 패킷을 재전송할 때 전송율도 같이 줄이는 혼잡제어를 동시에 수행한다. 무선 링크는 유선 링크에 비해 multipath fading, shadowing, mobility 등으로 인해 높은 비트 손실율과 burst한 패킷 손실이 발생할 수 있다. 따라서 무선 링크를 포함하는 TCP 연결은 비 혼잡 손실로 인한 패킷 손실도 혼잡 제어로 복구하기 위해 불필요하게 윈도우

\* 연세대학교 전기전자공학과 고성능 멀티미디어 네트워크 연구실{danny, hjcyy, sgkion, jy1}@nasla.yonsei.ac.kr

논문번호 : KICS2004-10-238, 접수일자 : 2004년 10월 19일

※ 본 연구는 한국과학재단 목적기초연구(R01-2002-000-00531-0) 지원으로 수행되었음.

(congestion window) 크기를 감소시키고 또한 burst 한 손실 특성으로 인해 한 윈도우에서 다수개의 패킷 손실이 발생한 경우에는 RTO(Retransmission TimeOut)을 발생시켜 전송율을 떨어뜨린다 결과적으로 이러한 복구 과정은 TCP의 성능을 많이 떨어뜨리게 된다<sup>[1],[2],[3]</sup>. RTO의 경우는 재전송 타이머가 만료되기 이전까지 패킷을 더 이상 전송하지 못하고 slow start 모드로 전송을 시작하고 연속된 RTO가 발생하면 RTO 값을 두 배로 계속 증가시키기 때문에 링크의 이용도를 낮게 할 수 있다 실제 트래픽을 측정한 결과 TCP의 재전송 중에 56%가 RTO에 의해 발생하고, 나머지 44%가 fast retransmit에 의해 발생한다<sup>[4]</sup>. 따라서 패킷 손실을 RTO없이 복구될 수 있다면 TCP의 성능을 개선할 수 있다. 1)

RTO 발생원인은 윈도우의 크기가 작아서 3 개 이상의 중복 승인 패킷(duplicate acknowledgement)을 수신하지 못하는 경우, 한 윈도우에서 다수개의 패킷 손실이 발생하는 경우, 재전송한 패킷이 다시 손실되는 경우, fast recovery 과정 중에 새로 전송하는 new packet<sup>3)</sup>이 손실된 경우로 나눌 수 있다<sup>[5]</sup>. 작은 윈도우로 인한 RTO는 Limited Transmit을 이용하여 수신되는 중복 승인 패킷의 수를 늘려 RTO를 막을 수 있다<sup>[5]</sup>. 다수개의 패킷 손실이 발생하는 경우는 TCP Reno를 대신해 TCP NewReno 또는 TCP SACK을 이용하여 RTO없이 복구할 수 있다. 하지만 TCP NewReno는 fast recovery과정 중에 패킷 손실이 발생하는 경우 RTO가 발생하는 문제를 가지고 있다 TCP NewReno는 fast recovery 과정 중에 전송할 수 있는 패킷은 손실된 패킷을 재전송하거나 new packet을 전송할 수 있다 재전송 패킷이 다시 손실될 때 RTO가 발생하고, 이 경우가 전체 타임아웃의 약 4%를 차지한다<sup>[6],[7]</sup>. 또한 new packet 손실 시에도 대부분의 경우 줄어든 윈도우 크기로 인해 충분한 중복 승인 패킷을 수신할 수 없기 때문에 fast retransmit에 의한 복구가 불가능하고 RTO이 발생한다.

1) TCP 전송계층에서 사용되는 세그먼트 대신 패킷으로 표시하였다.  
 2) 윈도우의 크기는 송신단의 혼잡윈도우와 수신단의 광고 윈도우(advertised window)중에 작은 값으로 설정한다.  
 3) fast recovery 과정 중에 usable window의 증가로 인해 fast retransmit 이후에 전송한 새로운 패킷을 new packet으로 정의한다.

본 논문에서는 TCP NewReno의 fast recovery 과정 중에 발생할 수 있는 RTO의 두 가지 원인(재전송 손실과 new packet 손실)을 RTO 없이 복구할 수 있는 방법인 Duplicate Acknowledgement Counting Plus(DAC<sup>+</sup>)와 Extended Fast Recovery (EFR)를 분석한다<sup>[13],[14],[15]</sup>.

## II. 제안 알고리즘

### 2.1 DAC<sup>+</sup>(Duplicate Acknowledgement Counting Plus)

TCP는 fast retransmit과 fast recovery 과정을 통해 손실된 패킷을 복구한다 DAC<sup>+</sup>는 이러한 복구 과정 중에 재전송된 패킷이 다시 손실되었을 때 RTO 없이 패킷을 복구할 수 있게 한다 이를 위해 DAC<sup>+</sup>는 재전송하기 바로 전의 윈도우의 크기( $s_{cwnd}$ )와 재전송된 패킷이 정상적으로 전송되었을 때 수신할 수 있는 예상 중복 승인 패킷의 수를 계산한다. Fast recovery 과정 중에 송신단은 손실된 패킷에 대한 중복 승인 패킷의 개수를 세고 이때 예상 중복 승인 패킷의 수보다 많은 중복 승인 패킷을 수신하면 송신단은 재전송된 패킷이 다시 손실되었다고 판단하고, 재전송 타임아웃을 기다리지 않고 손실된 패킷을 다시 재전송한다 한 윈도우에  $i$ 개의 패킷이 손실된 경우는 각각의 예상 중복 승인 패킷의 수는  $DAC_i$ 에 저장한다. 송신단은 한 윈도우에서 다수개의 패킷이 손실되었다 하더라도 수신되는 중복 승인 패킷에 의해서만 손실 여부를 판단하기 때문에 몇 개가 손실되었는지를 알 수 없다 다만 최소한 하나 이상의 패킷이 손실되었다는 것만 알 수 있으므로 첫 번째 fast retransmit된 패킷에 대한  $DAC_1$ 은 언제나  $s_{cwnd} - 1$ 이 된다. 두 번째 또는 세 번째 재전송한 패킷 손실도 예상 중복 승인 패킷수를 계산하여 복구할 수 있다

그림 1은 재전송된 패킷이 손실되었을 때 DAC<sup>+</sup>의 복구 과정을 나타내었다 송신단은 1.1초에 윈도우의 크기가 8이 되고, 패킷 7-14번까지 전송하지만 패킷 7이 손실되었다. 1.42초에 손실된 패킷에 대한 3개의 중복 승인 패킷을 수신하게 되면 송신단은 fast retransmit으로 패킷 7을 재전송한다 이때  $cwnd$ (congestion window)와  $ssthresh$ (slow start

4) 각각의 패킷을 구별하기 위해 0번부터 시작하는 패킷 시퀀스 번호를 사용하였고, 0번 패킷에 대한 승인 패킷은 1번으로 나타내었다.

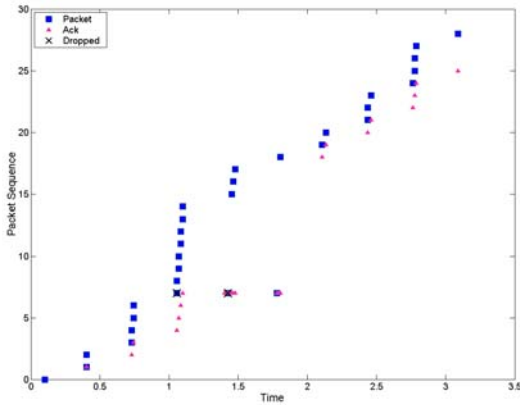


그림 1. 재전송 패킷 손실시 DAC+ 복구 과정

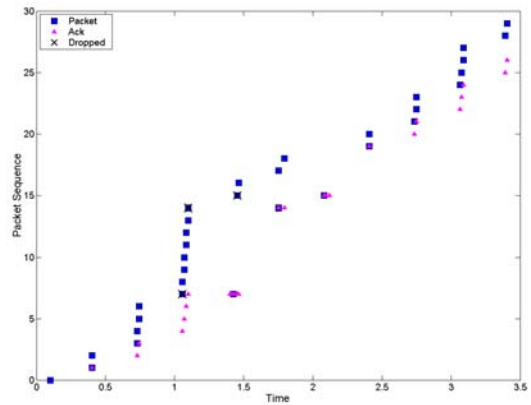


그림 2. Fast recovery 과정중 new packet 손실시 EFR 복구 과정

threshold)를 현재 윈도우의 반인  $4(=8/2)$ 로 설정하고 usable window를  $7(=cwnd+3)$ ,  $DAC_1$ 도  $7(=8-1)$ 로 설정한다<sup>6)</sup>. Fast retransmit된 패킷 7은 다시 손실되고, 이후에 패킷 7에 대한 4개의 중복 승인 패킷을 수신하면 usable window는 11로 증가하게 되어 새로운 패킷 15, 16, 17번을 전송할 수 있게 된다. 패킷 7번이 수신단에 제대로 도착하지 않았기 때문에 새로 전송한 15-17에 의해 3개의 중복 승인 패킷이 발생하게 된다. 패킷 15에 의해 발생하는 8번째 중복 승인 패킷을 수신하면 이 값은 예상 중복 승인 패킷 수( $DAC_1$ )보다 크기 때문에 송신단은 재전송 패킷이 손실되었다고 판단하고 패킷 7을 재전송하게 된다. 이때  $DAC^+$ 에서는 재전송된 패킷 손실과 처음 손실된 패킷은 서로 다른 혼잡으로 간주하고 현재 윈도우의 크기와  $ssthresh$ 를  $2(=4/2)$ 로 한 번 더 줄인다<sup>8)</sup>. 이후에 다시 재전송된 패킷 7번에 의한 정상 승인 패킷 18번이 도착하게 되면 송신단은 윈도우를  $2(ssthresh)$ 로 설정하고 fast recovery를 성공적으로 종료하고 congestion avoidance를 수행하게 된다. NewReno의 경우는 재전송된 패킷 7이 손실되어 정상 승인 패킷을 수신할 수 없어 fast recovery를 종료할 수 없기 때문에 재전송 타임아웃 이후에 윈도우를 1로 설정하고 패킷 7을 다시 재전송하고 slow start를 시작하게 된다.

2.2 EFR(Extended Fast Recovery)

TCP NewReno는 fast retransmit을 수행할 때 지금까지 전송한 패킷들 중에 가장 높은 시퀀스 번호를 recover에 저장한다<sup>9)</sup>. 송신단에서는 저장된 recover 값과 수신되는 승인 패킷의 시퀀스를 비교하여 recover 보다 작거나 같으면 부분 승인 패킷

(partial acknowledgement)으로, 그렇지 않은 경우는 정상 승인 패킷으로 분류한다. 부분 승인 패킷인 경우 TCP NewReno는 이 패킷의 시퀀스에 해당하는 패킷이 손실된 것으로 판단하고 재전송한다. TCP NewReno는 fast recovery중에 recover 값을 갱신하지 않는다. 따라서 fast recovery 과정 중에 new packet이 손실된 경우는 현재의 fast recovery가 종료된 이후에 손실된 패킷에 대한 새로운 3개의 중복 승인 패킷을 수신한 이후에나 복구 가능하다 하지만 이러한 경우는 usable window내에서 손실된 패킷의 위치뿐만 아니라 원래 손실이 발생한 윈도우의 크기와 손실된 패킷의 개수와 관련성을 갖고 있으며 대부분의 경우에 3개의 중복 승인 패킷을 받을 수 없기 때문에 RTO이 발생한다.

EFR을 사용하는 송신단은 fast recovery 과정 중에 손실된 패킷을 재전송할 때마다 recover를 현재 전송한 패킷 중에 가장 높은 시퀀스로 갱신하면 재전송 이전에 발생한 new packet 손실에 대해 항상 부분 승인 패킷을 수신할 수 있게 되므로 송신단은 fast recovery를 종료하지 않은 상태에서 손실된 new packet을 복구할 수 있다. 이 경우에도  $DAC^+$ 처럼 윈도우와  $ssthresh$ 를 반으로 줄인다<sup>8)</sup>.

Fast recovery 과정 중에 new packet 손실이 발생한 경우 EFR의 복구 과정을 그림 2에 나타내었다. 윈도우의 크기가 8이고 패킷 7번부터 14번까지 전송하였을 때 7번 패킷과 14번 패킷이 손실되었고, fast recovery 과정 중에 new packet 15번이 손실된 경우이다. 첫 번째 7번 패킷을 재전송할 때까지의 모든 과정은 그림 1에 나타난  $DAC^+$ 의 경우와 동일하다. 송신단은 7번을 재전송하면서 recover를 14로 설정한다. 이후에 3개의 중복 승인 패킷을 더 수신

한 후 *usable window*는 10으로 증가되고 *new packet* 15와 16을 전송하지만 15번이 손실된다. 약 1.75초에 송신단은 두 번째 손실 패킷인 14번에 대한 부분승인 패킷을 수신하게 되면 *recover*를 16으로 갱신하고 14번을 재전송한다. 이때 윈도우는 4(*ssthresh*)로 설정되기 때문에 17번 패킷도 새로 전송된다. 이후에 패킷 16에 의해 발생하는 중복 승인 패킷 14번을 수신하고 *usable window*가 5로 증가하여 패킷 18번을 전송한다. 이후 2.08초에 갱신된 *recover* 값 16보다 작은 부분 승인 패킷 15번이 수신되면 송신단은 패킷 15번을 재전송한다. 이때 *cwnd*와 *ssthresh*를 2([4/2])로 한 번 더 줄인다. 이후에 패킷 17과 18번에 의한 중복 승인 패킷을 수신하면 *usable window*를 4로 증가시키지만 이미 패킷을 전송한 상태이므로 새로 전송되는 패킷은 없다. 재전송한 패킷 15에 의해 승인 패킷 19번을 수신하면 이 값은 현재의 *recover* 16보다 크기 때문에 윈도우를 2(*ssthresh*)로 설정하고 *fast recovery*를 종료하게 된다. NewReno의 경우는 재전송한 패킷 14번에 의해 손실된 *new packet* 15번에 대한 승인 패킷을 수신하게 되면 이 값이 *recover*에 저장된 14보다 크기 때문에 윈도우를 4로 설정하고 *fast recovery*를 종료하게 된다. 이후에 패킷 17번과 18번에 의해 2개의 중복 승인 패킷 15를 수신하지만 중복 승인 경계 값 3보다 적은 수를 수신하였기 때문에 *fast retransmit*을 수행하거나 새로운 패킷을 전송하지 못하므로 RTO가 발생한다.

### III. 손실 복구에 관한 모델링

본 논문에서 손실 복구 확률은 재전송을 통해 복구된 패킷과 전체 손실 패킷 수의 비로 정의한다. TCP는 사용하는 *fast recovery* 알고리즘에 따라 재전송을 통한 손실 복구 확률이 서로 다르다는 것이 이미 여러 연구에서 분석되었다<sup>[12],[13]</sup>. Kumar는 마코프 체인(Markov chain) 분석을 통해 식 (1)과 (2)에 나타낸 것처럼 손실 윈도우 크기의 정상 상태 분포,  $\pi(M)$ 과 패킷 손실 복구 확률,  $F_{M,Kumar}^{NR}$  분석을 통해 *fast retransmit* 확률을 구하였다<sup>[12]</sup>.

$$P_{FR} = \sum_{M=1}^{W_{max}} F_M \cdot \pi(M) \quad (1)$$

5) *fast recovery*과정 중에 *fast retransmit*, 부분 승인 패킷, DAC+, EFR에 의해 복구된 패킷

$$F_{M,Kumar}^{NR} = \begin{cases} 0, & 1 \leq M \leq K \\ \sum_{n=K}^{M-1} \binom{M-1}{n} (1-p)^n p^{M-1-n}, & K+1 \leq M \leq W_{max} \end{cases} \quad (2)$$

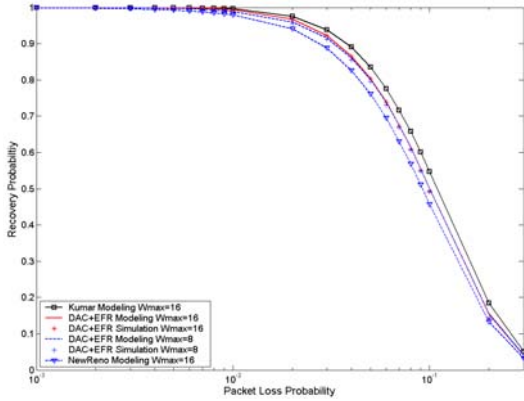
식 (2)의  $F_M$ 은 패킷 손실 확률이  $p$ 일 때 TCP NewReno의 *fast recovery* 성공 확률이다.  $K(=3)$ 는 중복 승인 패킷의 경계 값이고,  $W_{max}$ 는 수신단의 최대 윈도우 크기,  $n$ 은 정상적으로 수신단에 도착한 패킷에 의해 발생한 중복 승인 패킷의 수를 나타낸다. 손실된 패킷이 1개일 때 최대  $M-1$ 개의 중복 승인 패킷을 수신할 수 있고 이것이  $n \geq K$ 를 만족시켜야 한다. 즉 한 윈도우에서 최소 3개 이상의 패킷이 정상적으로 수신되어야 NewReno는 *fast retransmit*을 수행할 수 있다. Kumar의 TCP NewReno 모델은 첫 번째 손실 패킷이 *fast retransmit*에 의해 복구된다면 어떠한 경우에도 RTO이 발생하지 않는 이상적인 경우를 모델링 하였다<sup>[12]</sup>. 이후에 재전송된 패킷을 고려한 TCP NewReno의  $F_M$ 이 연구되었다<sup>[13]</sup>. 본 논문에서는 Kumar의 유도 방식을 따라 재전송된 패킷뿐만 아니라 *fast recovery*과정 중에 전송된 *new packet*까지 고려한 *fast recovery* 성공 확률을 구한다.

$$F_M^{NR} = \begin{cases} 0, & 1 \leq M \leq K \\ \sum_{n=K}^{M-K} \binom{M-1}{n-1} p^{n-1} \cdot (1-p)^{M+\alpha}, & K+1 \leq M \leq W_{max} \end{cases} \quad (3)$$

(where  $\alpha = \sum_{i=1}^{n-1} \phi_i$ )

식 (3)의  $n$ 은 식 (2)의  $n$ 과 달리 손실된 패킷의 개수이다. *Fast retransmit*이 수행되기 위해서는 손실된 패킷의 최대 개수가  $M-K$ 보다 작아야 한다.  $\phi_i$ 는  $i$ 번째 손실 패킷이 재전송된 이후에 전송된 *new packet*의 개수이다.  $\phi_1 = M/2 - n$ 으로 쉽게 구할 수 있고  $\phi_{i \geq 2}$  값은 윈도우 내에 손실된 패킷의 위치에 따라 값이 변한다.  $\alpha$ 는 첫 번째 손실된 패킷의 재전송 시점부터  $n$ 번째 손실 패킷이 재전송되기 전까지의 *new packet*의 합이다. 따라서  $n$ 개의 재전송된 패킷들이 전송에 성공할 확률  $(1-p)^n$  과 *new packet*들이 전송에 성공할 확률  $(1-p)^\alpha$  을 고려하여  $F_M$  값을 구하면, 손실 윈도우의 크기가  $M$ 인 상태에서의 *fast retransmit* 확률이 아닌 재전송 패킷과 *new packet* 손실 복구까지 포함한 손실 복구 확률을 구할 수 있다.

DAC 모델의  $F_M$ 은 식 (4)으로 나타낼 수 있다<sup>[13]</sup>.



(a) 각 알고리즘별 손실 복구 확률

그림 3. 모델링 분석과 모의실험 결과 비교

EFR 모델링은 식 (5)와 같다 .

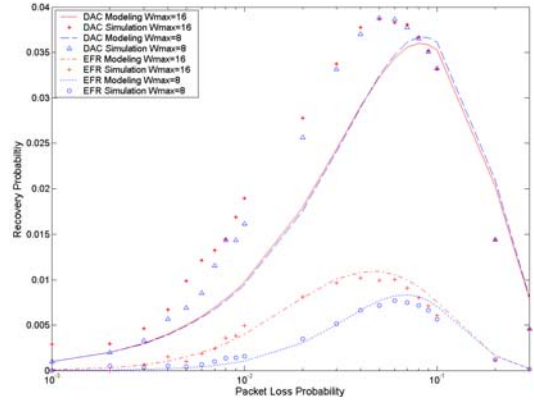
$$F_M^{DAC} = p(1-p)^M \cdot \left\{ \sum_{n=1}^{[M/2]} \binom{M-1}{n-1} p^{n-1} + \sum_{n=1}^{[M/2]} \binom{[M/2]-1}{n-1} p^{n-1} + \sum_{n=3}^{M-K} \sum_{m=3}^n \binom{M-m}{h-2} \binom{m-1}{n-h+1} p^{n-1} \right\} \quad (4)$$

$$F_M^{EFR} = \sum_{n=1}^{M-K} p^{n-1} (1-p)^M \cdot \sum_{k=1}^{\alpha} \binom{\alpha}{k} p^k (1-p)^{\alpha} \quad (5)$$

#### IV. 실험 환경 및 성능 평가 방법

본 논문의 제안 알고리즘 성능 평가를 위해 ns-2를 사용하였다<sup>10)</sup>. TCP 연결은 송신단과 수신단 사이에 10Mbps의 전송속도와 50msec의 전달지연을 갖는 링크를 사용하였고 데이터 패킷의 크기는 1KB이다. 패킷 손실은  $p$ 의 확률로 랜덤하게 발생하고 각 패킷 손실당  $10^6$ 개 패킷을 전송하는 것으로 가정하였다. 모든 TCP 변종들은 패킷 손실율이 너무 낮거나 높으면 거의 동일한 성능을 갖기 때문에 실험에서는  $10^{-3}$ 부터  $3 \cdot 10^{-1}$ 까지만 고려하였다. 승인 패킷의 크기는 데이터 패킷에 비하여 크기가 작기 때문에 전송 중에 손실되지 않고 하나의 TCP연결 설정을 고려하였기 때문에 “ACK-compression” 현상은 발생하지 않는 것으로 가정하였다<sup>11)</sup>. 윈도우는 연결 설정시 수신단에서 허용하는 최대 윈도우 크기,  $W_{max}$ 까지 증가할 수 있다.

제안 알고리즘은 DAC<sup>+</sup>와 EFR을 같이 사용하였



(b) DAC와 EFR의 손실 복구 확률

고, 대부분의 성능에 대한 결과는 이전 논문<sup>15)</sup>에서 다루었으므로 본 논문에서는 모의실험 결과와 제안 알고리즘의 손실 복구 확률 결과에 대한 정확성과 패킷 손실의 burst에 따른 성능을 평가한다

#### V. 결과 분석

그림 3은 식 (1)~(5)를 이용한 모델링 분석과 모의실험 결과를 비교한 그림이다. 그림 3-(a)는  $W_{max}$ 가 16일 때 TCP NewReno의 Kumar 모델, *new packet*과 재전송된 패킷의 손실을 고려한 NewReno 모델과  $W_{max}$ 가 8과 16일 때 NewReno에 DAC<sup>+</sup>와 EFR를 같이 적용하였을 때의 손실 복구 확률을 나타내었다. TCP NewReno 모델과 Kumar의 모델의 손실 복구 확률의 차이는 Kumar의 모델이 한 윈도우안에서  $K(=3)$ 개 이상의 패킷만 전송에 성공하면 fast retransmit이 수행되므로 그 이후에는 모든 패킷이 전송에 성공한다는 가정 때문이다 이에 비해 NewReno 모델은 fast retransmit에 의해 재전송된 패킷이 다시 손실될 수 있고, 또한 fast recovery과정 중에 전송한 *new packet*들이 손실될 수 있음을 고려하였기 때문에 Kumar의 모델보다 낮은 값을 보인다. DAC와 EFR을 NewReno에 적용하였을 때는 모의실험과 분석 모델의 값이 잘 일치하는 것을 볼 수 있다. 이들의 손실 복구 확률은  $W_{max}$  값에 의해 근소한 차이를 보인다 이것은  $W_{max}$  값에 의해 fast recovery 과정 중에 전송한 *new packet*의 개수와 손실의 개수가 달라졌기 때문이다 즉 하나의 패킷이 손실되었을 때 윈도우의 크기가 8인 경우 송신단은 fast retransmit 이후에 3 ( $=[8/2]-1$ )개의 *new packet*을 전송할 수 있고 윈도우의 크기가

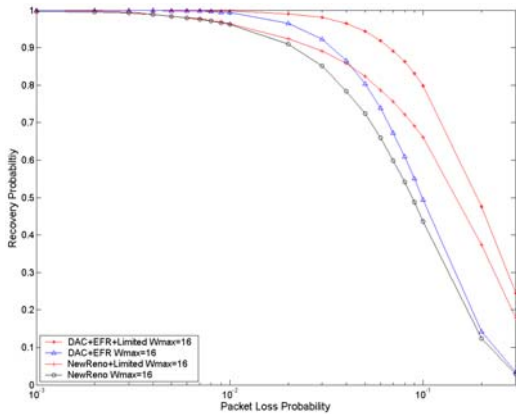


그림 4. Limited Transmit 적용 시 손실 복구 확률( $W_{max}=16$ )

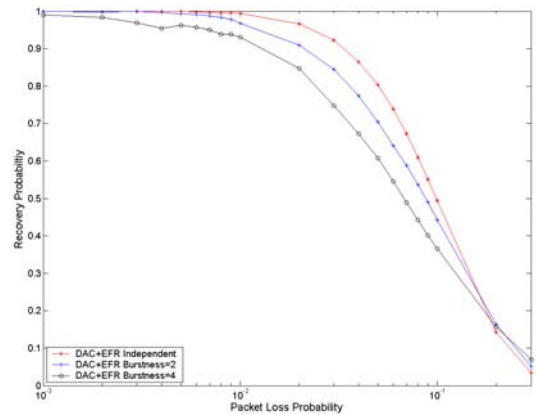


그림 5. Burstness에 따른 손실 복구 확률( $W_{max}=32$ )

16인 경우는  $7(=[16/2]-1)$ 개의 *new packet*을 전송할 수 있다. 즉 같은  $p$ 에서 많은 수의 *new packet*은 많은 *new packet* 손실을 발생시킬 수 있고, EFR은  $W_{max}$ 가 16 일 때 더 많은 *new packet* 손실을 RTO 없이 복구할 수 있다고 추정할 수 있다. 그러나  $p$ 가 증가할수록 윈도우의 크기는 최대값인  $W_{max}$ 에 도달하기 전에 크기를 계속 줄이기 때문에 큰  $p$ 의 값에서는 큰 차이를 보이지 않는다. 하지만 패킷 손실이 존재할 경우 윈도우의 크기가 작게 유지되더라도 제안 알고리즘의  $W_{max}$ 가 8인 경우가 TCP NewReno의  $W_{max}$ 가 16일 때보다 큰 손실 복구 확률을 갖는다. 이것은 제안 알고리즘이 윈도우의 크기가 작은 경우에도 효과적으로 패킷손실을 복구하는 것을 나타낸다. DAC<sup>+</sup>의 경우는 중복 승인 패킷의 수에 의해 동작하기 때문에  $W_{max}$ 의 값과는 관련이 없다. 이를 확인하기 위해  $W_{max}$ 이 8과 16일 때 DAC와 EFR의 모델링과 모의 실험의 결과를 그림 3-(b)에 나타내었다.

Limited Transmit은 TCP NewReno에 적용할 경우 윈도우의 크기가 4보다 작을 때에도 손실 패킷을 RTO없이 복구할 수 있는 효과적인 방법이다. 이를 DAC와 EFR에 적용하였을 때 손실 복구 확률을 그림 4에 나타내었다. Limited Transmit을 사용하는 경우에 NewReno보다는 높은 손실 복구 확률을 얻을 수 있지만 재전송 손실과 *new packet* 손실이 발생할 수 있기 때문에 Limited Transmit을 DAC와 EFR과 같이 사용하는 경우가 TCP NewReno와 같이 사용하였을 때보다 훨씬 좋은 성능을 보인다. 특히 전송하려는 데이터의 양이 적거나, 수신단의 버퍼 제한 또는 중단간 혼잡 제어로 인해 윈도우의 크기를 크게 할 수 없는 환경에서는

이 세 가지 알고리즘을 같이 사용하는 것이 효과적이다.

그림 5는 패킷 손실이 burstness를 갖고 있는 경우에 대한 실험 결과이다. 특히 무선은 유선에 비해 패킷 손실에 대한 burstness가 강하므로 이러한 손실 특성에 따른 손실 복구 확률을 확인해 볼 필요가 있다. 패킷 손실에 대한 burstness가 커질수록 손실 복구 확률이 떨어지는 것을 확인해 볼 수 있다.

지금까지의 결과를 통해 DAC<sup>+</sup>와 EFR을 NewReno에 적용하였을 때 각 알고리즘이 어느 정도 손실 복구 확률에 도움을 주는지를 모의실험과 모델링을 통해 분석하였다.

## VI. 결론

TCP 송신단이 패킷을 전송할 때 어떤 패킷이 손실되었는지 빠르고 정확하게 알 수 있다면 EFR과 DAC<sup>+</sup>와 같은 재전송 방법을 통해 RTO 이전이라도 손실 패킷을 복구하는 것은 효율적인 방법일 뿐 아니라 네트워크에 추가적인 혼잡을 초래하지 않는다. 이것은 fast recovery 과정 중에 네트워크가 혼잡 상황이라면 수신되지 않을 중복 승인 패킷을 수신함으로써 손실된 패킷을 재전송하고 윈도우를 반으로 줄이기 때문이다. 즉 fast recovery 과정 중에 발생한 재전송 손실과 *new packet* 손실을 빠르게 감지하여 RTO 없이 복구할 수 있다. 특히 제안 알고리즘은 무선 상에서 web과 같은 응용에서 짧은 데이터 전송 중에 발생하는 재전송 손실과 *new packet* 손실로 인한 성능 저하를 방지할 수 있고 Limited Transmit과 함께 사용된다면 효율적으로 모든 RTO의 원인에 대처할 수 있게 된다. 본 논문

에서는 제안 알고리즘에 대한 모델과 모의실험을 통해 성능 향상을 검증하였다

참 고 문 헌

[1] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", *IEEE/ACM Transactions on Networking*, vol. 5. no. 6, pp. 756-769, 1997.

[2] M. Zorzi, A. Chockalingam, and R. R. Rao, "Throughput Analysis of TCP on Channel with Memory", *IEEE Journal on Selected Areas in Communications*, vol. 18. no. 7, pp. 1289-1300, 2000.

[3] C. Barakat, E. Altman, and W. Dabbous, "On TCP Performance in A Heterogeneous Network: A Survey", *IEEE Communications Magazine*, pp. 40-46, 2000.

[4] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, M. Stemm, and R. H. Katz, "TCP Behavior of a Busy Internet Server: Analysis and Improvements", *IEEE INFOCOM'98*, pp. 252-262 vol1, 1998.

[5] M. Allman, H. Balakrishnan, S. Floyd, "Enhancing TCP's Loss Recovery Using Limited Transmit", *RFC 3042*, January 2001.

[6] K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP", *ACM Computer Communication Review*, vol. 26. no. 3, pp. 5-21, 1996.

[7] Dong Lin and H. T. Kung, "TCP Fast Recovery Strategies: Analysis and Improvements", *IEEE INFOCOM'98*, pp. 263-271, 1998.

[8] M. Allman, V. Paxson, W. Stevens, "TCP Congestion Control", *RFC 2581*, April 1999.

[9] S. Floyd and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm", *RFC 2582*, 1999.

[10] <http://www.isi.edu/nsnam/ns/index.html>

[11] J. Mogul, "Observing TCP dynamics in real networks," in *Proc. SIGCOM'92*, pp. 305-

317, 1992

[12] Anurag Kumar. "Comparative Performance Analysis of Versions of TCP in a Local Network with a Lossy Links," *IEEE/ACM Transactions on Networking (ToN)*, vol. 6, no. 4, pp. 485-498, Aug. 1998

[13] 김범준, 김동연, 이재용, "임의 패킷 손실에 대한 TCP의 손실 복구 과정 모델링 및 분석", *한국통신학회논문지*, 제 28권 4B호, April 2003.

[14] Dongmin Kim, Beomjoo Kim, Jechan Han and Jaiyong Lee, "Enhancements to the Fast Recovery Algorithm of TCP NewReno", *Lecture Notes in Computer Science 3090*, Springer-Verlag, pp.332-341, 2004.

[15] 김동민, 김범준, 김석규, 이재용, "빠른 손실 감지를 통한 TCP NewReno의 Fast Recovery 개선 알고리즘", *한국통신학회논문지*, 제27권 7B호, July 2004.

김 동 민(Dong min Kim)

정회원



1999년 2월 건국대학교 전자공학과 졸업  
 2001년 8월 연세대학교 전기전자공학과 석사  
 2001년 9월~현재 연세대학교 전기전자공학과 박사과정  
 <관심분야> TCP 모델링 및 성능 분석, 무선 TCP, 무선 랜, IPv6, Mobile IP, 센서네트워크

한 제 찬(Je chan Han)

정회원



2002년 8월 연세대학교 기계전자공학부 졸업  
 2004년 8월 연세대학교 전자공학과 석사  
 2004년 9월~현재 연세대학교 전기전자공학과 박사과정  
 <관심분야> 무선 TCP, RLP

김 석 규(Seog gyu Kim)

정회원



1990년 2월 연세대학교 전자  
공학과 졸업

1992년 8월 연세대학교 전자  
공학과 석사

1997년 8월 연세대학교 전자  
공학과 박사

1997년 9월~2004년 3월 SK

텔레콤 선임연구원

2004년 3월~현재 연세대학교 전기전자공학과 IT  
연구단 연구 교수

<관심분야> 유,무선 통합망에서 QoS Architecture,  
B3G Convergence Network, TCP 성능 분석, 무  
선 링크상의 TCP 성능 향상 방안, IP기반 유무  
선 통합 네트워크

임 차 식(Cha sik Leem)

정회원

1982년 2월 한국항공 대학교 전자공학과 졸업

1988년 6월 Georgia Institute of Technology 대학  
원 전자과 석사

2001년 9월~현재 연세대학교 전기전자공학과 박  
사과정

<관심분야> 차세대 정보통신 네트워크, 유비쿼터스  
센서 네트워크

이 재 용(Jai yong Lee)

중신회원



1977년 2월 연세대학교 전자  
공학과 졸업

1984년 5월 IOWA State Uni-  
versity 공학석사

1987년 5월 IOWA State Uni-  
versity 공학박사

1987년 6월~1994년 8월 포항

공과대학 교수

1994년 9월~현재 연세대학교 전자공학과 교수

<관심분야> Protocol Design for QoS Management,  
Network Management, High Speed Networks