

# 재전송 타임아웃이 TCP 성능에 미치는 영향과 완화 방안들의 모델링을 통한 성능 분석

정희원 김범준\*, 김석규\*\*, 이재용\*\*

## Effects of Retransmission Timeouts on TCP Performance and Mitigations: A Model and Verification

Beomjoon Kim\*, Seoggyu Kim\*\*, and Jaiyong Lee\*\* *Regular Members*

### 요약

TCP 처리율(throughput) 저하의 가장 큰 원인인 재전송 타임아웃(retransmission timeout)을 사전에 방지하기 위한 많은 노력들이 진행되어 왔다. TCP 손실 복구 알고리즘 자체의 오 동작으로 발생하는 타임아웃의 원인은 크게 세 가지로 분류될 수 있다. 현재 가장 널리 사용되고 있는 TCP Reno의 동일한 윈도우에서 발생한 여러 개의 패킷 손실로 인한 타임아웃은 TCP NewReno 혹은 선택 승인(selective acknowledgement) 옵션을 통해서 방지할 수 있고, 윈도우의 크기가 작은 상황에서 중복 승인 패킷(duplicate acknowledgement)의 부족으로 인해서 발생하는 패킷 손실은 제한 전송(Limited Transmit) 기법에 의해서 방지할 수 있다. 본 논문에서는 TCP 타임아웃이 발생하는 상황과 이를 완화하기 위한 방안들로 인한 개선 정도를 정확한 모델링을 통한 수학적 분석과 시뮬레이션을 통해서 비교 분석한다. 본 논문의 결과를 토대로 앞으로 사용할 TCP의 패킷 손실 정도에 따른 손실 복구(loss recovery)성능을 정량적으로 분석하고 예측하는 것이 가능하다. TCP의 성능은 손실 복구 과정의 성능에 크게 좌우된다는 점을 고려할 때 이는 매우 큰 중요성을 가진다.

키워드: TCP 혼잡 제어, TCP 손실 복구, TCP 모델 및 성능 분석, 제한 전송, 선택 승인

### ABSTRACT

There have been several efforts to avoid unnecessary retransmission timeouts (RTOs), which is the main cause for TCP throughput degradation. Unnecessary RTOs can be classified into three groups according to their cause. RTOs due to multiple packet losses in the same window for TCP Reno, the most prevalent TCP version, can be avoided by TCP NewReno or using selective acknowledgement (SACK) option. RTOs occurring when a packet is lost in a window that is not large enough to trigger fast retransmit can be avoided by using the Limited Transmit algorithm. In this paper, we comparatively analyze these schemes to cope with unnecessary RTOs by numerical analysis and simulations. On the basis of the results in this paper, TCP performance can be quantitatively predicted from the aspect of loss recovery probability. Considering that overall performance of TCP is largely dependent upon the loss recovery performance, the results shown in this paper are of great importance.

### I. 서론

1988년 혼잡 제어 메커니즘의 기본적인 동작이

TCP Tahoe[1]에 구현된 이래, TCP 손실 복구 메커니즘 자체의 오동작으로 인해서 발생하는 불필요한 재전송 타임아웃을 방지하기 위한 많은 노력들

\* LG전자 이동통신기술연구소 표준화그룹 (beom@lge.com),

\*\* 연세대학교 전기전자공학과 네트워크 연구실 (jyl@nasla.yonsei.ac.kr), 논문번호 : 030540-1206, 접수일자 : 2003년 12월 6일

※본 연구는 퀄컴-연세 합동 CDMA 연구 센터 (Qualcomm Yonsei CDMA Joint Research Center)지원으로 수행되었음.

(This work was supported by Qualcomm Incorporated through Qualcomm Yonsei CDMA Joint Research Center.)

논문번호 : #030366-0820, 접수번호 : 2003년 8월 20일

이 진행되어 왔다<sup>[1],[2],[3],[4],[5]</sup>. 이는 TCP의 전체적인 처리율(throughput)이 손실 복구 과정의 성능, 즉, 패킷 손실이 발생한 경우 이들이 재전송 타임아웃(retransmission timeout)을 유발시키지 않고 재전송에 의해서만 복구가 가능한지의 여부에 크게 영향을 받기 때문이다<sup>[7]</sup>.

불필요하게 발생하는 재전송 타임아웃은 그 원인에 따라 크게 세 가지로 분류되어질 수 있다. 첫 번째는 가장 오랫동안 주목받아온 문제로서 TCP Reno<sup>[2]</sup>가 동일한 윈도우에서 발생한 다수 개의 패킷 손실을 효율적으로 재전송을 통해 복구하지 못한다는 점이다. 이 문제점은 TCP Reno의 fast recovery 알고리즘을 개선한 TCP NewReno<sup>[3]</sup>를 도입하거나 혹은 선택 승인(selective acknowledgement) 옵션<sup>[4]</sup>을 사용함으로써 해결될 수 있다. 두 번째는 윈도우의 크기가 작은 시점에서 패킷 손실이 발생하는 경우 중복 승인(duplicate acknowledgement) 패킷의 개수의 부족으로 인해서 발생하는 타임아웃이다. 이 문제는 최근 HTTP(Hypertext Transfer Protocol)을 기반으로 동작하는 웹이 TCP 연결의 큰 비중을 차지하게 됨에 따라 주목을 받게 되었다<sup>[5]</sup>. 그<sup>[5]</sup> 결과에 따르면 웹 서버에서 전송하는 재전송 패킷들의 약 56%가 타임아웃에 의한 것이고 나머지 44%만이 fast retransmit에 의해 복구된다. 최근 이 문제를 해결하기 위한 제한 전송(Limited Transmit) 기법이 제안되었다<sup>[6]</sup>. 마지막 세 번째는 재전송 손실(retransmission loss)이 발생한 경우 발생하는 타임아웃으로 이 문제는 아직 공식적으로 해결되지 못한 상태이므로 이에 대한 논의는 향후 다른 논문에서 다루 예정이다.)

모델링을 통해서 TCP의 성능을 분석하고 예측하기 위한 대부분의 연구<sup>[7],[8],[9],[10],[11],[12],[13]</sup>들은 TCP 처리율의 근사화에 중점을 두고 있는 반면, TCP의 성능이 손실 복구 성능에 크게 좌우되는 점에도 불구하고 손실 복구 과정의 정확한 모델링의 중요성은 간과되어왔다. 대표적인 모델<sup>[8]</sup>에서는 TCP Reno만을 고려하고 있고 다른 대표적인 모델<sup>[7]</sup>은 TCP Reno와 NewReno를 모두 고려하고 있지만, 두 모델 모두 제한 전송이 TCP의 손실 복구 성능에 미치는 영향을 고려하고 있지 않다. 따라서 본 논문에서는 제한 전송이 TCP의 손실 복구 과정의 성능에

미치는 정도를 확률적으로 분석하기 위한 모델을 제시하고, 이를 통해 얻어진 결과를 시뮬레이션을 이용해서 검증하였다. 추가적으로 이 TCP NewReno에 대한 모델<sup>[8]</sup>은 재전송 손실에 의해서 발생하는 타임아웃의 영향이 고려되어 있지 않다. 본 논문에서는 이를 고려함으로써 TCP NewReno의 손실 복구 과정을 보다 정확하게 모델링하는 것이 가능함을 보인다. 본 논문의 분석 결과를 통해서 재전송 타임아웃이 TCP의 성능에 미치는 영향을 그 원인과 그에 대응하는 개선 방안에 세부적으로 평가할 수 있다는 점에 그 중요성이 있다.

## II. TCP의 손실 복구 확률

TCP 혼잡 제어 메커니즘은 패킷 손실을 감지하고 복구하기 위한 기능을 포함하는데, 흔히 이를 간단히 손실 복구(loss recovery)라고 한다. TCP 손실 복구는 fast retransmit과 fast recovery의 기본적인 두 개의 알고리즘으로 이루어져 동작하는데, 손실 복구 성능을 향상시키기 위해서 fast recovery 알고리즘은 지속적인 수정을 거쳐왔다. 본 논문에서 고려하고 있는 TCP Reno와 NewReno의 손실 복구 과정의 동작에 대해서는 이미 많은 연구<sup>[1],[2],[3]</sup>를 통해서 언급되어 있으므로 본 절에서는 최근 제안된 제한 전송<sup>[6]</sup>을 사용하는 TCP의 동작에 대해서만 언급하기로 한다.

제한 전송은 다음의 두 가지 조건 하에서 중복 승인 패킷이 수신될 때마다 새로운 패킷을 전송한다. 첫 째, 수신원이 공시하는 윈도우(advertised window)의 크기가 새로운 패킷의 전송을 제한하지 않아야 한다. 둘째, 제한 전송이 동작하는 시점에서의 승인되지 않은 패킷(outstanding packet)들의 개수는 현재 혼잡 윈도우의 크기에 두 개의 패킷 크기만큼을 더한 값보다 커질 수 없다. 쉽게 말해서 어떤 손실된 패킷에 대해서 적어도 하나의 중복 승인 패킷이 수신되는 경우 그 패킷은 fast retransmit에 의해 재전송이 가능하다. 그림 1에는 제한 전송의 동작을 ns 시뮬레이션을 통해서 나타내었다.

윈도우가 세 개의 패킷을 포함하는 시점에서 전송한 세 개의 패킷들 가운데 첫 번째 패킷이 손실된 경우를 가정해보자. 2) 0.1초경에 혼잡 윈도우의 크기가 1인 상태에서 패킷 0을 전송하고, 이에 대한 승

1) 이 문제를 해결하기 위한 재전송 손실 복구 알고리즘들은 본 논문의 저자에 의해 개발되어 [16]을 통해 발표될 예정이다.

2) 종류에 관계없이 모든 TCP의 손실 복구 과정은 첫 번째 손실된 패킷의 fast retransmit으로 시작되므로 그림 1의 동작은 모든 TCP에 공통적으로 해당한다.

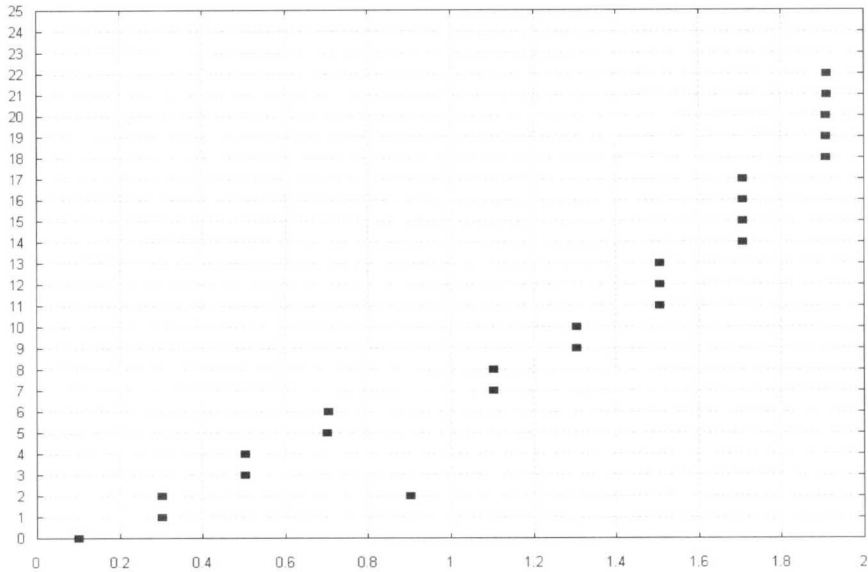


그림 5. 세 개의 패킷을 포함하는 윈도우에서 첫 번째 패킷이 손실된 경우 제한 전송의 동작 (x-axis: time, y-axis: packet sequence number)

인이 약 0.2초 후에 수신됨에 따라 혼잡 윈도우의 크기가 2로 증가하므로 두 개의 패킷 1~2가 전송된다. 패킷 1에 의해 패킷 2에 대한 승인 패킷을 수신함에 따라 윈도우의 크기는 3으로 증가하여 약 0.3초경에 두 개의 패킷 3과 4가 전송된다. 이 시점의 윈도우가 포함하는 첫 번째 패킷은 손실된 패킷 2가 된다. 현재 전송중인 패킷 3~4에 의해 손실된 패킷 2에 대한 두 개의 중복 승인 패킷을 수신하게 되는데 fast retransmit이 동작하기 위해서는 세 개의 중복 승인 패킷을 수신해야 하므로 제한 전송을 사용하지 않는 경우라면 패킷 2는 타임아웃 이후에 전송되게 된다. 그러나 제한 전송을 사용하는 경우라면 패킷 3과 패킷 4에 의해서 중복 승인 패킷이 수신될 때마다 새로운 패킷을 전송하므로 패킷 5와 패킷 6이 윈도우의 크기와 관계없이 전송되고 이들에 의해서 추가적인 두 개의 중복 승인 패킷을 수신하게 된다. 결과적으로 송신원은 손실된 패킷 2에 대한 총 네 개의 중복 승인 패킷을 수신하게 되는데 0.9초경에 세 번째 중복 승인 패킷이 수신되었을 때 이를 fast retransmit에 의해 재전송하고 이후의 전송을 congestion avoidance 상태에서 계속 진행하는 것이 가능하다.

제한 전송이 TCP Reno와 함께 사용되는 경우 다수개의 패킷 손실에 의한 TCP Reno의 손실 복구

성능을 부분적으로 향상시키는 것이 가능하다. 예를 들어, 여섯 개의 패킷을 포함하는 혼잡 윈도우내의 첫 번째와 두 번째 패킷이 손실되었다면 두 번째 손실된 패킷에 대해 하나의 중복 승인 패킷을 수신하는 것이 가능하다. 물론 제한 전송을 사용하지 않는 경우라면 두 번째 손실된 패킷은 타임아웃 후에 전송된다. 제한 전송을 사용하는 경우라면 적어도 하나의 손실된 패킷에 대한 중복 승인 패킷을 수신하기만 하면 중복 승인의 개수 부족으로 인한 타임아웃은 발생하지 않는다. 이와 같이 제한 전송을 사용하면 TCP Reno의 다수개의 패킷 손실에 의한 성능 저하를 어느 정도 피하는 것이 가능하지만, 이 문제점은 TCP NewReno 혹은 선택 승인 옵션을 사용하면 근본적으로 해결이 되기 때문에 실제 제한 전송은 동일 윈도우내의 단 하나의 패킷 손실을 복구하는 경우에만 적용되도록 구현되었다<sup>6)</sup>.

### III. TCP 손실 복구 지연

패킷 손실 복구 과정을 분석하기 위해서 몇 가지 가정과 정의가 필요하다. 우선 각 패킷은 임의의 (random) 확률  $p$ 로 손실되고 패킷 손실들은 서로 독립적인 것으로 가정한다. 송신원은 전송해야 할 패킷들을 충분히 가지고 있어서 항상 윈도우가 허



용하는 최대 개수의 패킷들을 전송하는 것으로 가정한다. 수신원은 지연(delayed) 승인 방식을 사용하지 않으므로 하나의 패킷을 수신할 때마다 하나의 승인 패킷을 송신원에 전달하는 것으로 가정한다.

패킷 손실이 감지되기 전에 전송된 패킷들에 의한 모든 정상 승인을 수신했을 때의 윈도우의 크기를  $\Omega$ 로 나타내고 이를 손실윈도우(loss window)라고 한다<sup>[7]</sup>.<sup>3)</sup> 패킷 손실 과정 모델은 '라운드' 단위로 동작하고<sup>[8]</sup>,  $\Phi_i$ 는  $i$ 번째 손실된 패킷의 재전송 이후에 전송되는 패킷들의 수를 나타낸다. 패킷 손실에 따른 손실윈도우의 정상 상태 분포를 얻고 패킷 손실 복구 확률을 구하는 과정은 Markov 프로세스를 통한 분석 과정을 도입한다<sup>[7],[8],[12]</sup>.

1. TCP Reno

$\Omega = u$  를 만족하는 손실윈도우에서 손실된 패킷의 개수  $n$ 에 대한 TCP Reno의 손실 복구 확률을  $R_R^{(n)}(u)$ 라 했을 때, 이는 다음의 식 (1)과 같이 주어진다<sup>[15]</sup>.<sup>4)</sup>

$$R_R(u) = \begin{cases} R_R^{(1)}(u) & \text{if } 4 \leq u < 10 \\ R_R^{(1)}(u) + R_R^{(2)}(u) & \text{if } 10 \leq u < 24 \\ R_R^{(1)}(u) + R_R^{(2)}(u) + R_R^{(3)}(u) & \text{if } 24 \leq u \end{cases}$$

$$\begin{aligned} R_R^{(1)}(u) &= (1-p)^u \\ R_R^{(2)}(u) &= (u-1)p(1-p)^{u+\phi_1} \\ R_R^{(3)}(u) &= \binom{\lceil u/4 \rceil - K}{2} p^2 (1-p)^{u+\phi_1+\phi_2} \end{aligned}$$

(1)

제한 전송을 사용하는 경우 손실윈도우가 두 개 혹은 세 개의 패킷들을 포함할 때 발생한 하나의 패킷 손실을 fast retransmit에 의해 복구하는 것이 가능하다. 따라서  $\Omega = u$ 을 만족하는 손실윈도우에 대해 제한 전송을 사용하는 경우의 TCP Reno의 손실 복구 확률  $R_{RL}(u)$

은 식 (2)으로 표현할 수 있다.

$$R_{RL}(u) = R_R(u) + R_L(u) \tag{2}$$

$R_L(u)$ 은 크기가  $u$ 인 손실윈도우에서 발생한 패킷 손실이 제한 전송에 의해 복구될 확률을 의미한다. 이는  $\Omega = u$ 를 만족하는 손실윈도우에서 단 하나의 패킷 손실이 발생하고 제한 전송에 의해 추가적으로 전송되는 패킷들과 fast retransmit에 의해 재전송되는 패킷은 손실되지 않을 때의 확률과 동일하므로,  $2 \leq u \leq 3$ 을 만족하는 손실윈도우에 대해 다음과 같이 유도된다.

$$\begin{aligned} R_L(u) &= (1-p)^{u-1}(1-p)^2(1-p) \\ &= (1-p)^{u+2} \end{aligned} \tag{3}$$

2. TCP NewReno

TCP NewReno의 손실 복구 확률은 이전 연구<sup>[7]</sup>에 유도되어 있다. 그러나 이전 모델<sup>[7]</sup>에서는 첫 번째 손실된 패킷에 대해서 세 개의 중복 승인 패킷이 수신되는 경우에 TCP NewReno의 손실 복구 과정은 언제나 성공하는 것으로 가정하였다. 그러나 실제로는 재전송된 패킷들이 다시 손실되는 경우에도 타임아웃이 발생할 수 있기 때문에 본 논문에서는 TCP NewReno의 성공적인 손실 복구를 위한 조건에 이를 고려하였다. 따라서  $\Omega = u$ 인 손실윈도우에서  $n$ 개의 패킷 손실들에 대한 TCP NewReno의 손실 복구 확률  $R_{NR}(u)$ 은 다음의 식 (4)로 주어진다.

$$\begin{aligned} R_N(u) &= \sum_{n=1}^{u-3} \binom{u-1}{n-1} p^{n-1} (1-p)^{u-n} (1-p)^n \\ &= (1-p)^u \sum_{n=1}^{u-3} \binom{u-1}{n-1} p^{n-1} \end{aligned} \tag{4}$$

제한 전송을 사용하는 경우에 첫 번째 손실된 패킷에 대해서 적어도 하나의 중복 승인 패킷을 수신할 수 있는 경우 모든 패킷 손실은 재전송에 의해서 복구되는 것이 가능하므로 제한 전송을 사용하는 TCP NewReno의 손실 복구 확률  $R_{NL}(u)$ 은 다음과 같이 주어진다.

3) 손실윈도우가 포함하는 첫 번째 패킷은 항상 손실된 패킷이 된다.

4) [15]에서 본 논문의 저자는 TCP의 손실 복구 과정을 분석하기 위한 기본적인 틀을 제시하였으며, 이를 기반으로 TCP Reno의 손실 복구 확률을 패킷 손실의 개수에 대해서 유도하였다. 특히, 세 개의 패킷 손실도 경우에 따라 TCP Reno에 의해서 복구될 수 있음을 증명하였으며, 본 논문에서는 이 연구에서 유도된 식을 그대로 사용하였다.

$$\begin{aligned}
 R_{NL}(u) &= \sum_{n=1}^{u-1} \binom{u-1}{n-1} p^{n-1} (1-p)^{u-n} (1-p)^n \\
 &= (1-p)^u \sum_{n=1}^{u-1} \binom{u-1}{n-1} p^{n-1}
 \end{aligned}
 \tag{5}$$

식 (4)와 마찬가지로 식 (5)의  $(1-p)^n$ 은  $n$ 개의 재전송 패킷들은 모두 손실되지 않아야 한다는 것을 의미한다.

#### IV. 결과 분석

##### 1. 손실 복구 확률

각 TCP의 손실 복구 기능을 비교 분석하기 위해 서 평균 손실 복구 확률을 다음의 식(6)과 같이 정의한다. 이 값은 손실된 패킷들이 모두 재전송에 의해서만 복구될 패킷 손실 확률에 대한 평균 확률을 의미한다.

$$R_{tcp} = \sum_{u=1}^{W_{max}} R_{tcp}(u) \cdot \pi(u)$$

(6)  $R_{tcp}(u)$ 는 각 TCP의 손실 복구 확률이고  $W_{max}$ 는 수신원이 공시하는 최대 윈도우(advertised window)의 크기를 의미한다.  $\pi(u)$ 는 정상 상태에서의 윈도우 크기의 확률 분포를 의미하는데, 이는 Markov 프로세스를 도입하여 윈도우의 주기적인 변화 특성을 분석함으로써 얻어진다 [7],[8],[12].

##### 2. 결과 및 분석

그림 2는 제한 전송을 사용하는 경우와 그렇지 않은 경우의 랜덤 패킷 손실 확률에 대한 손실 복구 확률을 보여준다. TCP NewReno가 다수 개의 패킷 손실을 복구할 수 있다 할지라도, 이를 위해서는 언제나 첫 번째 손실된 패킷을 fast retransmit에 의해 재전송하는 것이 필요하다. 따라서, 모든 TCP의 손실 복구 성능은 패킷 손실의 개수와는 관계없이 첫 번째 손실된 패킷이 fast retransmit에 의해 재전송될 수 있는지에 영향을 받는다. 그림 2에서는 첫 번째 손실된 패킷을 fast retransmit하는데 있어서 제한 전송을 사용함으로써 얻을 수 있는 성능 향상만을 고려하기 위해서  $W_{max}$ 의 값을 4로 제한하였다. 손실된 패킷에 대한 fast retransmit이 동작하기 위해서는 세 개의 중복 승인이 필요한 점을 고려했을

때, 제한 전송을 사용하지 않는 TCP는 하나의 패킷 손실에 대해 대부분의 경우, 즉, 윈도우의 크기가  $W_{max}$ 와 동일한 4가 아닌 이상, 타임아웃을 유발하게 된다. 반면 제한 전송을 사용하는 TCP는 손실된 패킷에 대해 적어도 하나 이상의 중복 승인을 수신하기만 한다면 fast retransmit은 동작할 수 있다.

패킷 손실 확률이 증가할수록 제한 전송으로 인한 성능 향상의 폭이 증가하는 것을 볼 수

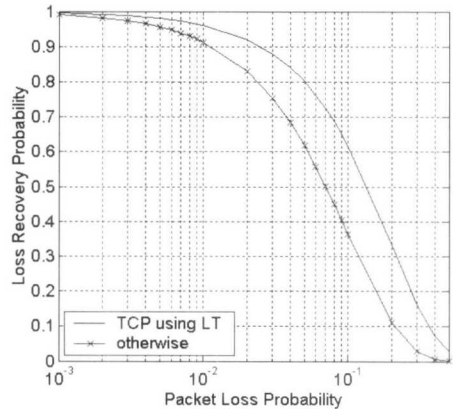


그림 6. 제한 전송을 사용하는 경우와 그렇지 않은 경우의 첫 번째 fast retransmit의 성공 확률 비교 ( $W_{max}=4$ )

있는데, 이는 윈도우의 크기가 작을수록 제한 전송을 사용하는 경우의 fast retransmit이 성공할 확률이 그렇지 않은 경우에 비해 증가하기 때문이다. 패킷 손실 확률이 증가함에 따라 대부분의 윈도우의 크기 분포는 작은 상태에 머물러 있게 되고, 이 때 발생한 패킷 손실은 제한 전송을 사용하지 않는 경우 대부분 타임아웃을 발생시키는데 반해, 제한 전송을 사용하는 경우 윈도우의 모든 패킷이 손실되지 않는 이상 타임아웃은 발생하지 않는다. 결과적으로 제한 전송은 윈도우의 크기가 작을수록 더 큰 성능 향상을 가져온다는 점을 확인할 수 있다.

앞의 2장에서 언급한 것과 같이 제한 전송의 주된 목적은 다수 개의 패킷 손실에 의한 TCP Reno의 성능 저하 문제를 개선하기 위한 것이 아니기 때문에, 그림 3에서 볼 수 있듯 제한 전송이 TCP Reno와 사용되는 경우와 TCP NewReno와 사용되는 경우의 손실 복구 성능은 상당한 차이를 보인다. 제한

전송을 사용하지 않는 경우의 두 그래프의 차이는 다수 개의 패킷 손실로 인한 손실 복구 성능의 차이를 의미한다. TCP NewReno의 손실 복구 능력은 패킷 손실 확률 약  $10^{-2}$ 에서  $10^{-1}$ 인 구간에서 TCP Reno에 비해서 최대 15%정도 우월하다. 그러나  $10^{-2}$  이상의 패킷 손실 확률에 대해서는 거의 차이가 나지 않고 오히려 제한 전송을 사용하는 TCP Reno의 손실 복구 능력보다도 떨어지는 것을 볼 수 있다. 이는 TCP NewReno의 손실 복구 과정

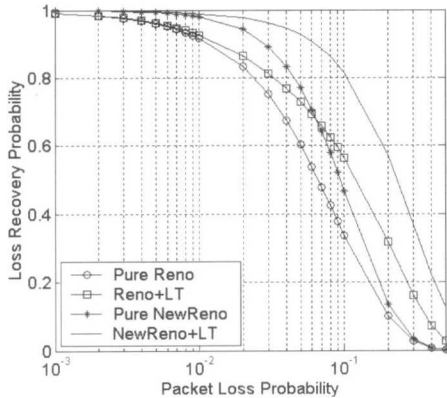


그림 7. 제한 전송에 따른 TCP Reno와 TCP NewReno의 손실 복구 확률 비교 ( $W_{max}=8$ )

이 첫 번째 손실된 패킷을 fast retransmit할 수 없는 경우에는 정상적으로 동작하지 않는다는 점을 반영하는 것이다. TCP NewReno와 제한 전송을 사용하는 경우 다수 개의 패킷 손실로 인한 성능 저하와 작은 윈도우에서 첫 번째 fast retransmit이 동작하지 않는 문제를 동시에 해결하는 것이 가능하므로 전 구간의 패킷 손실 확률에 대해 가장 높은 손실 복구 성능을 보인다.

본 논문의 모델을 통해서 예측된 TCP의 손실 복구 성능의 산술적인 계산 결과를 검증하기 위해서 ns 시뮬레이션을 수행하고 그 결과를 TCP Reno와 TCP NewReno의 경우에 대해 각각 그림 4와 그림 5에 비교하였다. 시뮬레이션 상에서의 손실 복구 확률은 총  $10^4$ 개의 패킷을 전송하는 동안 발생한 패킷 손실의 개수와 재전송에 의해 복구되는 패킷들의 개수의 비로 정의된다.

그림 4는 제한 전송을 사용하는 경우와 그렇지 않은 경우 TCP Reno의 손실 복구 확률의 산술적 계산 결과와 시뮬레이션 결과를 비교해서 보여준다.

먼저 제한 전송을 사용하지 않는 TCP Reno의 경우, 앞의 3장에서 유도한 식에 의해 예측된 결과가 시뮬레이션 결과와 정확하게 들어맞는 것을 볼 수 있다. 추가적으로 패킷 손실의 연속성이 TCP Reno의 손실 복구 확률에 미치는 영향을 비교하기 위해서, 기존 모델<sup>[8]</sup>에서 근사적으로 유도된 TCP Reno의 손실 복구 확률도 함께 비교하였다. 기존 모델<sup>[8]</sup>에서는 동일한 라운드에서 패킷 손실들은 서로 상호 관련성을 가지고 있다고 가정하기 때문에, 즉, 하나의 라운드에서 손실된 패킷 이후에 전송되는

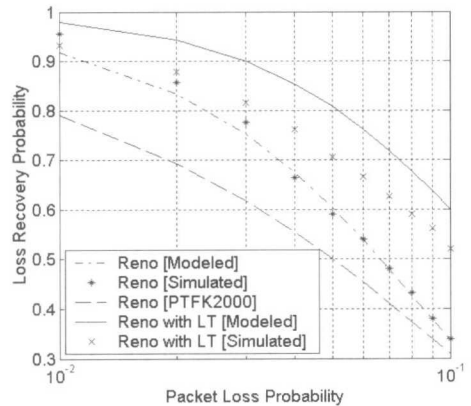


그림 8. 제한 전송을 고려한 TCP Reno의 손실 복구 확률의 산술적 계산 결과와 시뮬레이션 결과의 비교 ( $W_{max}=8$ )

모든 패킷 역시 손실되는 것으로 가정하기 때문에 송신원이 수신하게 될 중복 승인의 개수가 상당히 작은 값으로 계산되게 되어 TCP Reno의 손실 복구 확률을 상당히 저 평가하게 되는 것을 볼 수 있다.

제한 전송을 사용하는 경우의 손실 복구 확률의 계산 결과와 시뮬레이션 결과 사이에는 상당한 차이가 있음을 볼 수 있다. 이 현상을 설명하기 위해서 두 개의 패킷 손실을 포함하는 윈도우에서 첫 번째 패킷이 손실된 경우를 생각해 보자. 두 개의 패킷을 전송한 송신원은 한 RTT(round-trip time) 이후에 손실된 패킷에 대한 첫 번째 중복 승인을 받고 하나의 새로운 패킷을 전송한다. 다시 한 RTT 이후에 송신원은 두 번째 중복 승인을 수신하고 다시 하나의 새로운 패킷을 전송한다. 이 경우 송신원이 손실된 패킷에 대해서 세 개의 중복 승인을 수신하는 시점은 패킷 손실이 발생하기전의 전송 시점보다  $3*RTT$ 후가 된다. 만약 당시의 타임아웃 값이  $3*RTT$ 보다 작다면 해당 패킷은 세 개의 중복



승인 패킷을 수신하기 전에 타임아웃이 발생하게 된다. 즉, 세 번째 중복 승인이 수신되기 전에 발생하는 타임아웃으로 인해서 실제 제한 전송이 가져오는 성능 향상은 본 논문의 모델링 결과보다는 다소 낮은 값을 가지게 되는데, 이는 본 논문의 손실 복구 모델링 과정에서는 송신원에서 계산되는 타임아웃의 값의 변화를 고려하지 않고 중복 승인의 부

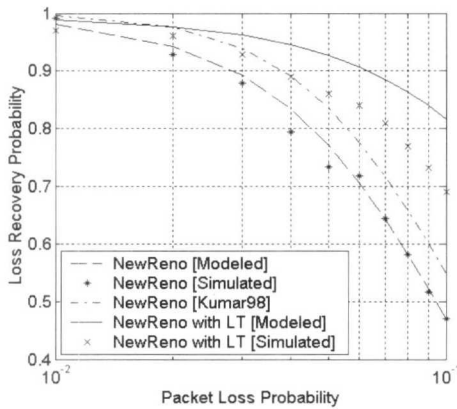


그림 9. 제한 전송을 고려한 TCP NewReno의 손실 복구 확률의 산술적 계산 결과와 시뮬레이션 결과의 비교 ( $W_{max}=8$ )

그림 5에서는 제한 전송을 사용하는 경우와 그렇지 않은 경우의 TCP NewReno의 손실 복구 확률의 모델링을 통한 산술적 계산 결과와 시뮬레이션 결과를 비교한다. 추가적으로 기존 연구<sup>7)</sup>에서 모델링한 TCP NewReno의 손실 복구 확률을 함께 비교한다. 기존 연구<sup>7)</sup>에서는 재전송 패킷 손실로 인해서 발생하는 RTO를 고려하지 않기 때문에 앞의 3장에서 식 (4)로 주어지는 손실 복구 확률의 계산 결과보다는 상당히 큰 값으로 계산된다. 시뮬레이션

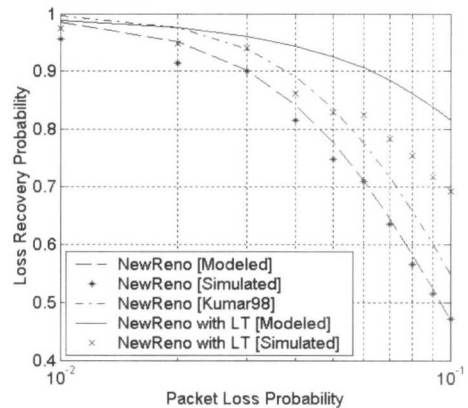


그림 11 제한 전송을 고려한 TCP NewReno의 손실 복구 확률의 산술적 계산 결과와 시뮬레이션 결과의 비교 ( $W_{max}=32$ )

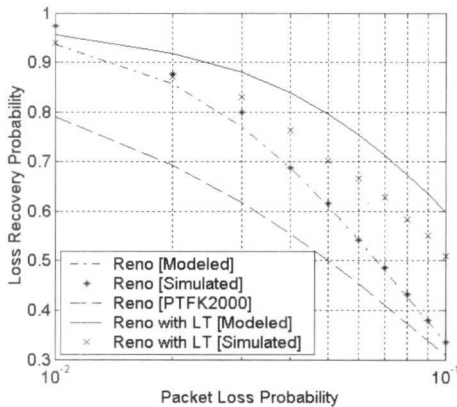


그림 6. 제한 전송을 고려한 TCP Reno의 손실 복구 확률의 산술적 계산 결과와 시뮬레이션 결과의 비교 ( $W_{max}=32$ )

족으로 인해서 발생하는 타임아웃만을 고려하였기 때문이다.

결과를 통해서 재전송 패킷 손실을 고려함으로써 보다 정확한 TCP NewReno의 손실 복구 과정을 모델링할 수 있다는 것을 확인할 수 있다. TCP Reno와 마찬가지로 제한 전송을 사용하는 경우의 손실 복구 확률은 산술적 계산 결과와 시뮬레이션 결과와 상당한 차이가 있다. 단지 TCP NewReno의 경우 낮은 패킷 손실에 대해서는 두 결과가 거의 일치하는 것을 볼 수 있는데, 이는 앞에서 설명한 이른 재전송 타임아웃 만료의 빈도로 설명할 수 있다. 동일한 패킷 손실 확률에 대해 TCP NewReno의 평균 윈도우의 크기 분포는 TCP Reno보다 상당히 큰 값을 가진다. 따라서, 낮은 패킷 손실 확률에 대해서는 fast retransmit 이전에 발생하는 타임아웃의 빈도가 TCP Reno보다는 훨씬 낮을 것이라 추정할 수 있다. 패킷 손실 확률이 증가할수록 윈도우의 크기 분포는 작은 값에 머물러 있게 되고 이른 재전송 타임아웃 만료의 빈도 역시 증가하게 되므로 계산 결과와 시뮬레이션 결과의 차이는 증가하는 것으로 분석할 수 있다.

5) 이 논문에서는 이를 '이른 재전송 타임아웃 만료 (early retransmission timeout expiry)'라고 부르기로 한다.

크기가 32인  $W_{max}$ 에 대해서도 같은 실험을 반복하였다. 그림 6과 그림 7에 나타난 결과가 앞의 그림 4와 그림 5의 결과와 거의 동일하다는 사실을 통해서  $W_{max}$  값의 증가가 손실 복구 확률에 거의 영향을 주지 않는다는 점을 확인할 수 있다. 앞서서도 언급한 TCP의 손실 복구가 성공하느냐 실패하느냐는 첫 번째 패킷 손실의 fast retransmit에 의한 재전송 가능 여부에 달려있다는 사실에 의해 다음

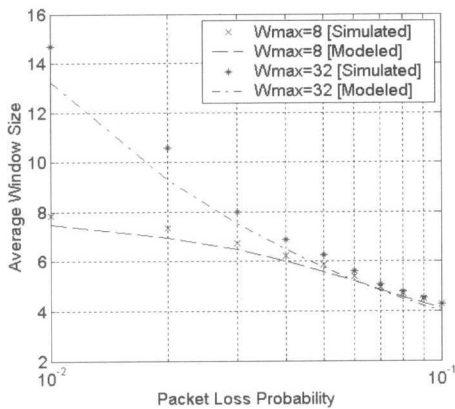


그림 12. 패킷 손실 확률에 대한 TCP Reno의 윈도우의 크기 변화 분포

과 같은 추론이 가능하다. 패킷 손실 확률이 낮은 경우, 하나의 윈도우에 포함되는 패킷 손실의 개수는 대부분 하나이고 이는 TCP Reno, NewReno에 관계없이 윈도우의 크기가 4보다 (제한 전송을 사용하는 경우 2보다) 크기만 하면 거의 모두 재전송에 의해서 복구되는 것이 가능하다. 패킷 손실 확률이 증가함에 따라 다수 개의 패킷 손실이 동일한 윈도우에 포함될 수 있다. 반면, 패킷 손실 확률이 증가함에 따라 윈도우의 크기는 잦은 패킷 손실로 인해서 그림 8에서와 같이 상당히 작은 값에 머무르기 때문에 큰  $W_{max}$  값으로 인한 영향은 거의 없다. TCP Reno가 두 개의 패킷 손실을 재전송에 의해서 복구하기 위해서는 윈도우의 크기가 적어도 10보다 커야한다는 것을 생각했을 때 이 사실은 쉽게 이해될 수 있다. 반면, TCP NewReno의 경우 첫 번째 fast retransmit만 동작하는 경우 손실된 패킷의 개수는 관계없이 모든 패킷 손실들을 재전송에 의해서 복구될 수 있기 때문에 역시  $W_{max}$  값을 증가시키더라도 손실 복구 차원에서의 성능 향상은 극히 미미하다.

## V. 결론

본 논문에서는 패킷 손실 정도에 따른 TCP Reno와 NewReno의 손실 복구 과정의 성능을 분석하기 위한 모델을 제시하고 이를 통해 예측된 결과를 시뮬레이션을 통하여 검증하였다. 특히, 본 논문의 모델링 과정은 기존의 모델에서는 고려하고 있지 않은 제한 전송이 TCP의 손실 복구 성능에 미치는 영향을 포함하고 있다. 제한 전송이 TCP Reno에 적용되는 경우와 NewReno에 적용되는 경우에 대해서 고려하였고, 분석 결과를 토대로 제한 전송을 사용함으로써 얻을 수 있는 성능 향상을 정량적으로 분석하였다. 추가적으로, TCP NewReno의 경우 재전송 패킷 손실을 고려함으로써 보다 정확한 모델링이 가능함을 시뮬레이션 결과를 통하여 입증하였다. 본 논문의 결과는 보다 정확한 TCP 손실 복구 성능을 미리 예측하고 이를 개선하기 위한 하나의 지표로서 그 의미가 있다.

## 참고 문헌

- [1] V. Jacobson, "Congestion Avoidance and Control," *ACM SIGCOMM '88*, Aug. 1988.
- [2] V. Jacobson, "Modified TCP congestion avoidance algorithm," note sent to end2end-interest mailing list, 1990.
- [3] J. Hoe, "Improving the Start-up Behavior of a Congestion Control Scheme for TCP," *ACM SIGCOMM '96*, Aug. 1996.
- [4] M. Mathis, S. Floyd, and A. Romanow, "TCP Selective Acknowledgement Options," RFC 2018, Oct. 1996.
- [5] H. Balakrishnan, Venkata Padmanabhan, Srinivasan Seshan, Mark Stemm, and Randy Katz, "TCP Behavior of a Busy Web Server: Analysis and Improvements," *IEEE INFOCOM '98*, 1998.
- [6] H. Balakrishnan and S. Floyd, "Enhancing TCP's Loss Recovery Using Limited Transmit," RFC 3042, 2001.
- [7] Anurag Kumar, "Comparative Performance Analysis of Versions of TCP in a Local Network with a Lossy Link," *IEEE/ACM Transactions on Networking (ToN)*, vol. 6, no. 4, pp. 485-498, Aug. 1998.



[8] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose, "Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation," *IEEE/ACM Transactions on Networking (ToN)*, vol.8, no.2, pp. 133-145, Apr. 2000.

[9] P. P. Mishra, D. Sanghi, and S. K. Tripathi, "TCP flow control in lossy networks: Analysis and enhancements," *Computer Networks, Architecture and Applications*, IFIP Transactions C-13, S. V. Raghavan, G. V. Bochman, and G. Pujolle, Eds. Amsterdam, The Netherlands: Elsevier North-Holland, pp. 181-193, 1993.

[10] T. V. Lakshman and Upamanyu Madhow, "The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss," *IEEE/ACM Transactions on Networking (ToN)*, vol.5, no.3, pp. 336-350, Jun. 1997.

[11] Farooq Anjum and Leandros Tassiulas, "On the Behavior of Different TCP Algorithms over a Wireless Channel with Correlated Packet Losses," *ACM SIGMETRICS'99*, pp. 155-165, May 1999.

[12] Michele Zorzi and A. Chockalingam, "Throughput Analysis of TCP on Channels with Memory," *IEEE J. Selected Areas in Comm. (JSAC)*, vol. 18, no. 7, pp. 1289-1300, Jul. 2000.

[13] Alhussein A., Abouzeid, S. Roy, and M. Azizoglu, "Stochastic Modeling of TCP over Lossy Links," *IEEE INFOCOM'2000*, pp. 1724-1733.

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

[15] Beomjoon Kim and Jaiyong Lee, "Analytic Models of Loss Recovery of TCP Reno with Packet Losses," *Lecture Note in Computer Science (LNCS)* no. 2662, Oct. 2003.

[16] Beomjoon Kim and Jaiyong Lee, "Retransmission Loss Recovery by Duplicate Acknowledgement Counting," *IEEE Communications Letters*, vol.8, no. 1, 2004.

[17] Beomjoon Kim *et al.*, "Lost Retransmission Detection for TCP Part 2: TCP using SACK Option," in Proc. IFIP TC-6 *Networking'2004 (LNCS no. 3042)*, 2004.

[18] Beomjoon Kim, Dongmin Kim, and Jaiyong Lee, "Lost Retransmission Detection for TCP SACK," accepted for publication in *IEEE Communications Letters*, 2004.

[19] Beomjoon Kim and Jaiyong Lee, "A Simple Model for TCP Loss Recovery over Wireless Networks," accepted for publication in *Journal of Communications and Networks (JCN)*, 2004.

김 범 준(Beomjoon Kim)

정회원



1996년 2월 : 연세대학교

전자 공학과 졸업

1998년 8월 : 연세대학교

전자 공학과 석사

2003년 8월 : 연세대학교

전자 공학과 박사

2003년 6월~2004년 1월:

연세대학교 전기전자공학과 전자정보통신 연구소  
박사후 과정 연구원

2004년 2월~현재 : LG전자 이동통신기술연구소  
표준화그룹

<주관심분야> IEEE 802.16, IEEE 802.21, B3G  
Convergence Network, TCP 성능 분석, 무선 링  
크상의 TCP 성능 향상 방안, IP기반 유무선 통  
합 네트워크, IP 트래픽 엔지니어링.

김 석 규(Seoggyu Kim)

정회원



1990년 2월 : 연세대학교

전자공학과 졸업

1992년 8월 : 연세대학교

전자공학과 석사

1997년 8월 : 연세대학교

전자공학과 박사

1996년~2004년 3월 :

SK Telecom

2004년 4월~현재 : 연세대학교 전기전자공학과  
IT 사업단 박사후과정 연구원

<관심분야> IEEE 802.16, B3G Wireless Network

이재용(Jaiyong Lee)

정회원



1977년 2월 : 연세대학교

전자 공학과 졸업

1984년 5월 : IOWA State

University 공학 석사

1987년 5월 : IOWA State

University 공학 박사

1987년 6월~1994년 8월 :

포항공과대학 교수

1994년 9월~현재 : 연세대학교 전자공학과 교수

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