

## 에러 모델을 적용한 TCP의 성능 분석

### Performance Analysis of TCP Using ErrorModel

김유두\*, 문일영\*

Yu-Doo Kim\* and Il-Young Moon\*

#### 요 약

TCP(Transmission Control Protocol)는 인터넷 환경에서 널리 사용되고 있는 프로토콜의 하나이다. TCP는 흐름제어를 통해 데이터의 손실 및 재전송을 줄여 효율성을 높일 수 있으며, 이러한 흐름제어는 데이터 패킷 크기의 제한을 두는 윈도우 기법을 통해 이루어지고 있다. 흐름제어 방식에 의해 TCP는 다양한 버전으로 구분되어 있는데, 본 논문에서는 대표적인 Tahoe, Reno와 Reno의 개량형인 Newreno에 에러모델을 적용한 시뮬레이션 결과를 통해 그 성능을 분석하였다.

#### Abstract

TCP (Transmission Control Protocol) is one of the protocols which are widely used from the Internet environments. Through the flow control of TCP, it could be increased efficiency for the loss and a re-transmission of data and the flow control become accomplished through window technique which puts the limit of size. By the flow control, TCP divided in various versions. In this paper, it is analyzed the simulation result which applies the error model in the Newreno which is an improved model of the representative Tahoe, Reno.

Key words : error model, NS-2, TCP Newreno

#### I. 서 론

신뢰성 있는 전송을 위하여 설계된 TCP는 인터넷 환경에서 널리 사용되고 있는 프로토콜이다. 그러나 전송확인 및 에러감지를 위한 패킷 발생으로 인하여 UDP(User Datagram Protocol)에 비하여 속도가 떨어지는 단점이 존재한다. 따라서 실시간을 요하는 방송, 전화 등에서는 신뢰성보다는 속도에 초점을 맞춘 UDP를 많이 사용하고 있지만 손실이 많이 발생하여 낮은 품질의 서비스를 할 수 밖에 없다. 이에 TCP는 신뢰성은 높으면서 속도를 개선할 수 있도록 1988년

의 TCP Tahoe이래 1990년 TCP Reno, 1995년 TCP Vegas에 이르기 까지 다양한 버전으로 구현되고 있다[1]-[3].

본 논문에서는 다양한 인터넷 프로토콜에 대한 시뮬레이션 기능을 제공하여 전 세계적으로 널리 사용되고 있는 네트워크 시뮬레이터인 NS-2에 에러모델을 적용하여 처음 개발된 TCP Tahoe를 시작으로 현재 가장 많이 사용되고 있는 Reno와 그 개량형인 NewReno의 성능을 비교, 분석하였다.

\* 한국기술교육대학교 인터넷미디어공학부(School of Internet Media Engineering, Korea University of Technology and Education)

· 제1저자 (First Author) : 김유두

· 접수일자 : 2006년 11월 27일

## II. TCP 흐름제어

TCP는 흐름제어를 통해 데이터의 손실 및 재전송을 줄여 효율성을 높일 수 있으며, 이러한 흐름제어는 데이터 패킷 크기의 제한을 두는 윈도우 기법을 통해 이루어지고 있다[2]. TCP 구현 중 최초로 1988년에 제안된 Tahoe는 현재 광범위 하게 사용되고 있는 알고리즘 이며, Reno는 Tahoe를 개선한 알고리즘으로 하나의 패킷 손실이 발생한 경우에는 최적의 성능을 발휘하지만, 한 윈도우 내에 다수의 패킷 손실이 발생한 경우에는 성능이 Tahoe에 비해 떨어지는 문제점이 있다. NewReno는 Reno의 다중 패킷 손실의 문제점을 개선한 알고리즘이다[3].

### 2-1 TCP Tahoe

Tahoe 동작은 연결 초기에 Slow-start를 시작하여 패킷을 송수신하다가, 패킷이 손실되어 타임아웃이 발생하거나 중복 ACK 3개가 수신되면 TCP는 임계치(Threshold)를 패킷손실 전의 CWND(Congestion Window)의 절반으로 설정한 후, CWND를 1로 설정하여 Slow-start를 실행한 다음 혼잡회피 단계로 그림 1 과같이 동작한다[2].

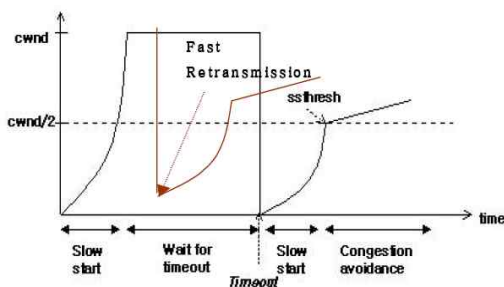


그림 1. Tahoe의 동작  
Fig. 1.Tahoe motion

Tahoe는 한 윈도우 내에서 처음 손실이 발생된 패킷을 재전송한 후 차례로 Slow-start를 실행한다. 손실 이후의 모든 패킷들을 재전송하므로 이미 수신한 패킷을 다시 수신하게 되는 문제점은 있으나 결과적으로 패킷 손실이 발생하지 않아 안전성은 높다. Tahoe는 Slow-start, 혼잡회피 및 빠른 재전송 알고리즘을 지원 한다[2][4].

### 2-2 TCP Reno

Reno는 기존의 Slow-start, 혼잡회피 알고리즘 외에 빠른 회복(Fast Recovery)과 빠른 재전송(Fast Retransmission) 알고리즘을 추가한 버전이다[1]-[4].

빠른 재전송은 만일 수신 호스트에서 순서 번호에 어긋나는 세그먼트가 수신되면, 수신 호스트는 수신되는 앞은 순서번호로 확인 응답을 보내게 된다. 이 상황은 올바른 순서번호의 세그먼트를 수신할 때까지 계속된다. 따라서 송신 측이 중복된 확인 응답이 3개 이상 수신하게 되면 세그먼트가 손실되었다는 것을 인식하여 세그먼트를 재전송함으로써 타이머가 만료될 때까지 기다리는 시간을 줄이는 방법이다.

빠른 회복은 초기에는 Slow-start로 동작하지만, 패킷 손실이 발생하면 임계값의 크기를  $CWND/2$ 로 설정하여 Slow-start 단계로 무시하고 혼잡회피 알고리즘으로 그림 2 같이 동작함으로써 효율적으로 망을 사용할 수 있다[2][4].

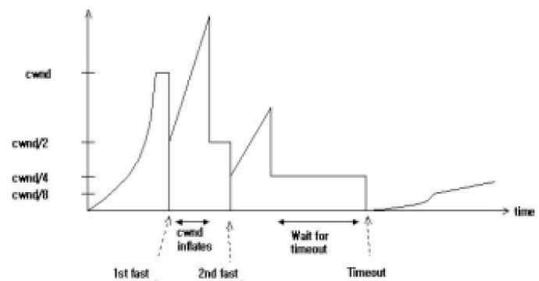


그림 2. Reno의 동작  
Fig. 2. Reno motion

혼잡상황이 발생했을 때 Slow-start 단계를 거치지 않고 충돌 회피 단계를 바로 시작하여 Tahoe 버전을 개선하였다. 하지만 한 윈도우 내에 한, 두 개의 세그먼트가 손실되었을 때는 성능이 우수하나 여러 개의 패킷이 손실되면 Window의 크기가 계속 줄어들어 일정 수 이상을 전달 할 수 없게 되는 상황이 발생할 수 있다[5].

### 2-3 TCP NewReno

New-Reno는 Reno의 빠른 재전송 알고리즘과 빠른 회복 알고리즘을 일부 수정하고 보완된 TCP 구현 이

다[1]-[5]. Reno와 가장 큰 차이점은 다중 패킷 손실이 발생한 경우 모든 손실된 패킷이 재전송되기 전까지는 빠른 회복 알고리즘을 종료하지 않는 것이다.

New-Reno는 빠른 재전송이 처음 동작하기 전에 TCP 송신단 에서 전송했던 최고 순서번호를 기억함으로써 다중 패킷 손실을 가지는 빠른 회복 알고리즘을 개선하였다.

New-Reno는 재전송된 패킷에 대한 ACK의 순서번호를 보고 송신단 에서 보낸 패킷 중에서 다중 패킷 손실이 발생이 발생했다면 부분적인 ACK으로 간주하고 손실이 발생한 패킷을 매 RTT마다 하나씩 재전송한다. Reno는 재전송한 패킷에 해당하는 ACK가 수신되면 빠른 회복 알고리즘을 종료하고 혼잡회피 동작 단계를 수행하지만, New-Reno는 그림 3 처럼 ACK가 수신되어도 CWND 크기를 변화시키지 않고 계속해서 두 번째, 세 번째 패킷에 대한 복구를 수행함으로써 멀티 패킷 손실이 일어날 경우에도 뛰어난 성능을 나타낸다[4][6].

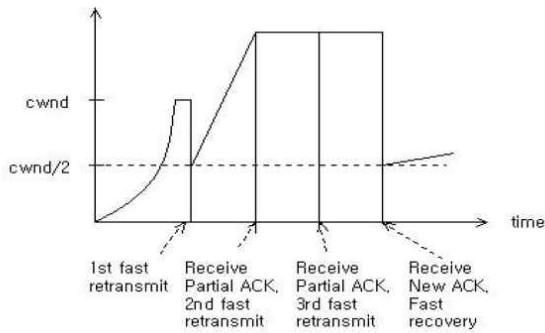


그림 3. NewReno의 동작  
Fig. 3. NewReno motion

### III. 에러모델의 적용

#### 3-1 에러모델

TCP Tahoe, Reno, NewReno는 신뢰성 있는 전송을 위해 데이터의 손실을 감지하고 재전송 해 주는 방법에 따라 구분되어진 것으로, 성능을 비교하기 위해서는 데이터가 손실되는 상황을 임의로 만들어야 한다.

에러모델은 실제 환경에서 일어나는 모델로써 NS-2에서 데이터 패킷을 전송할 때 임의적으로 에러를 생성할 수 있도록 하는 기능을 제공하는 모듈이

다. 이것은 두 개의 OTCL 메소드에 의해서 제공되며, 이 메소드들은 ns/tcl/lib/ns-lib.tcl에 정의되어 있다 [7].

본 장에서는 NS-2를 활용하여 노드 사이에 에러모델을 적용한 TCP와 에러모델을 적용하지 않은 TCP의 각 방식에 대한 성능을 시뮬레이션 하여 그래프로 그 결과를 나타낸다.

#### 3-2 Tahoe 시뮬레이션

그림 4는 약 25초간 에러모델을 적용하지 않은 Tahoe의 패킷 전송에 대한 시뮬레이션을 수행한 것이다. 그래프가 거의 일직선상으로 나타나는 것은 에러모델이 적용되지 않아 손실 없이 패킷이 일정하게 전송되었다는 것을 뜻한다.

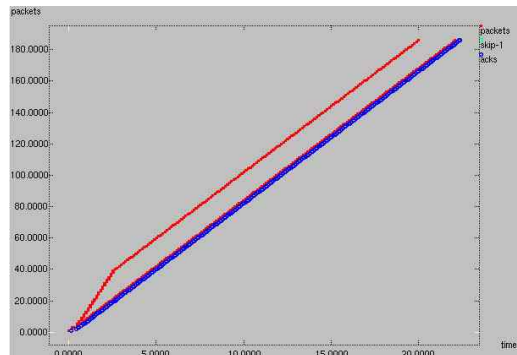


그림 4. 에러모델 미 적용 Tahoe 시뮬레이션  
Fig. 4. Tahoe simulation without ErrorModel

그림 5는 위에서와 같은 환경에 에러모델을 적용한 것으로 임의의 패킷손실이 발생하면서 그에 따른 Tahoe의 동작을 나타내고 있다[8].

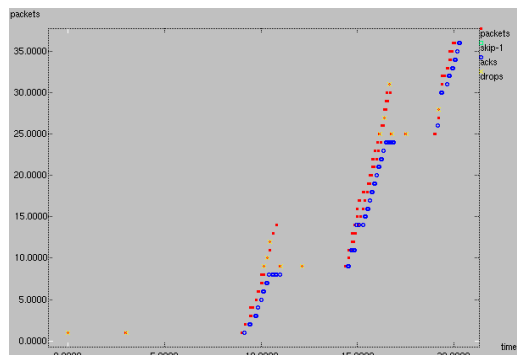


그림 5. 에러모델 적용 Tahoe 시뮬레이션  
Fig. 5. NewReno motion with ErrorModel

### 3-3 Reno 시뮬레이션

그림 6은 Tahoe시뮬레이션과 같은 환경에서 에러 모델이 적용되지 않은 상태의 결과를 나타낸다. 이 또한 Tahoe와 같이 패킷 손실이 일어나지 않아 일정한 직선 상태의 그래프로 나타난다.

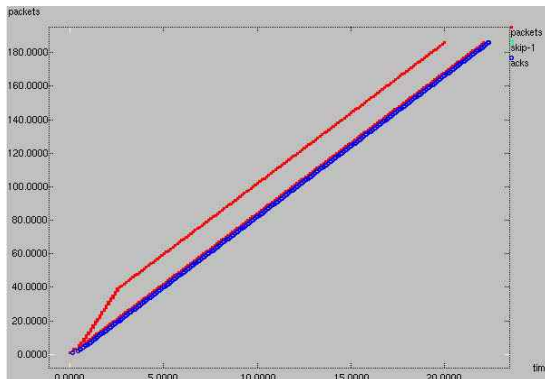


그림 6. 에러모델 미 적용 Reno 시뮬레이션  
Fig. 6. Reno simulation without ErrorModel

위의 시뮬레이션 환경에 에러모델을 적용한 결과는 그림 7에 나타나고 있다[8]. 자세한 내용은 전체적인 결과의 비교를 통해 설명하겠지만 대략적으로 보아도 Tahoe와는 다른 결과가 나타나는 것을 알 수 있다.

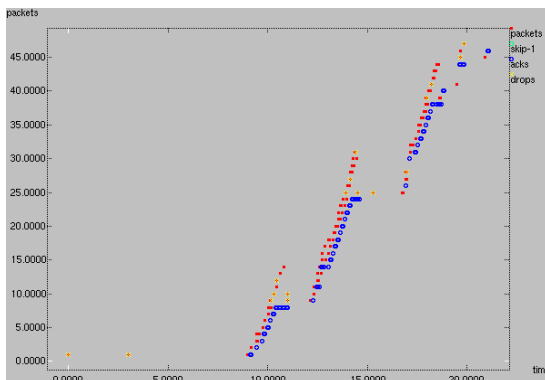


그림 7. 에러모델 적용 Reno 시뮬레이션  
Fig. 7. Reno simulation with ErrorModel

### 3-4 NewReno 시뮬레이션

그림 8은 에러모델이 적용되지 않은 NewReno의 동작을 나타내고 있다. 이 결과 또한 패킷 손실이 없어 직선모양의 그래프로 나타난다.

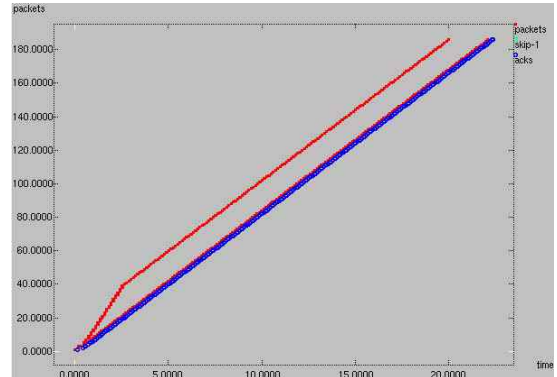


그림 8. 에러모델 미 적용 NewReno 시뮬레이션  
Fig. 8. NewReno simulation without ErrorModel

그림 9는 NewReno의 동작에 에러모델이 적용된 것으로 Reno와 비슷하지만 약간의 다른 결과를 나타내고 있다[8].

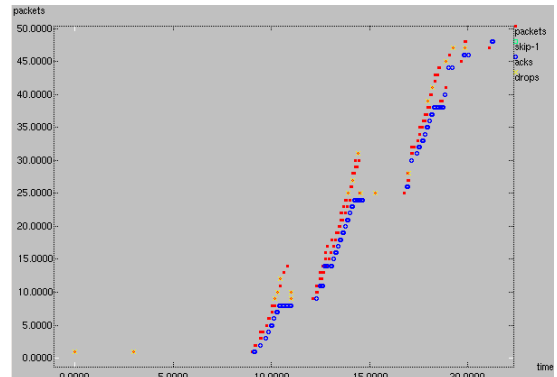


그림 9. 에러모델 적용 NewReno 시뮬레이션  
Fig. 9. NewReno simulation with ErrorModel

## IV. 시뮬레이션 결과 비교

본 장에서는 에러모델을 적용하였을 때와 적용하지 않았을 때의 TCP 동작 결과를 한 좌표에 나타내었다.

### 4-1 에러모델을 적용하지 않은 결과

에러모델을 적용하지 않으면 TCP는 패킷 손실 없이 일정하게 전송된다는 것을 위에서 살펴보았다.

그림 10에서는 TCP Tahoe, Reno와 NewReno의 동작을 한 좌표에 나타낸 것으로 모두 같은 결과를 나타내고 있어 하나의 그래프와 같이 표현되었다.

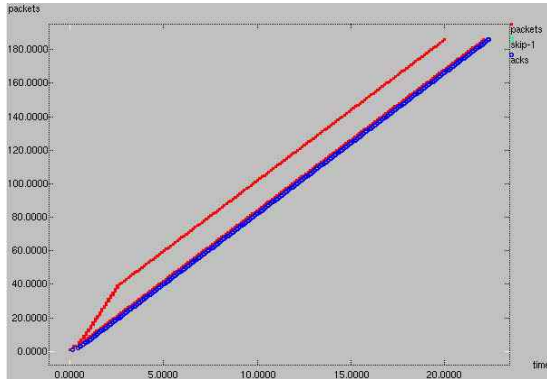


그림 10. 에러모델 미 적용  
Fig. 10. Simulation without ErrorModel

이는 패킷 손실이 일어나지 않을 경우 모든 TCP는 같은 성능을 보여준다는 것을 알 수 있다.

#### 4-2 에러모델을 적용한 결과

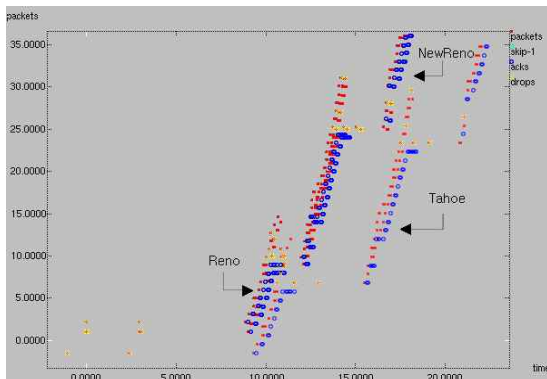


그림 11. 에러모델 적용  
Fig. 11. Simulation with ErrorModel

그림 11에서는 Tahoe, Reno와 NewReno에 에러모델을 적용한 결과를 한 좌표에 나타낸 그래프로, 각 TCP에 따라 다른 결과값을 보여주고 있다.

Tahoe의 경우 같은 시간의 패킷 전송량을 보면 Reno와 NewReno에 비하여 현저하게 차이가 나는 것을 볼 수 있으며, Reno와 NewReno는 아주 큰 차이는 아니지만 눈으로 식별 가능 할 정도의 성능 차이를 나타내고 있으며 역시 개량형인 NewReno가 약간 앞선 패킷 전송량을 보여주고 있다.

#### V. 결 론

본 논문은 에러모델을 통하여 TCP ACK신호에 Error를 임의로 발생 시켜 TCP의 Tahoe, Reno 그리고

NewReno에 대한 성능을 비교, 분석 하였다. 에러모델은 NS-2에서 기본적으로 제공하는 모듈로 패킷에 임의로 에러를 발생하여 그에 따른 처리에 대한 결과를 알아내기 위하여 사용된다.

TCP의 경우 패킷이 손실되었을 경우 손실을 감지하고 재전송하는 방식에 따라 Tahoe, Reno, NewReno로 구분되어 지기 때문에 에러모델을 적용하지 않은 패킷 전송에서는 세 방식 모두 동일한 결과를 나타낸다. 에러모델을 적용하여 임의의 패킷 손실을 발생시킬 경우 정상적인 ACK에서는 일정하게 패킷이 전송되지만 중복 ACK수신 또는 패킷 지연이 되면 세 방식에 따라 다른 성능을 보여주고 있다. 그 결과는 가장 처음에 제안된 Tahoe가 가장 좋지 못한 성능을 보여주고 있으며 Reno와 NewReno는 큰 차이는 아니지만 Reno의 개량형인 NewReno가 더 좋은 성능을 보여주고 있다. 이것은 같은 시간 동안 처리된 패킷의 양에서 나타나고 있다.

TCP는 신뢰성 있는 전송으로 인해 널리 사용되고 있지만 최근의 인터넷 활용은 방송, 인터넷 전화 등 실시간 중심의 서비스로 그 중심이 변화되고 있다. 시뮬레이션에서 가장 좋은 결과를 나타낸 NewReno의 경우도 에러모델을 적용하지 않은 결과에 비교하면 약 1/5의 패킷 처리량을 보여주고 있다. 따라서 TCP는 많은 연구를 통해 신뢰성을 유지하면서 더욱 빠르게 패킷 전송이 될 수 있는 알고리즘을 지속적으로 연구해야 한다.

#### 참 고 문 헌

- [1] V.Jacobson, " Congestion Avoidance and Control ", *Proceeding of the SIGCOMM'88 Symposium*, Aug. 1988.
- [2] W.Stevens, " TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", *RFC 2001*, Jan. 1997
- [3] V.Jacobson, " Modified TCP Congestion Avoidance Algorithm", *end2end-interest mailing list*, Apr. 1990.
- [4] 김대영 외 4명, 네트워크 효율 향상을 위한 개선된 TCP 혼잡제어 알고리즘, *전자공학학회 논문지*, 제 4권 TC편 제 8호, pp. 31-33, 2003. 8

- [5] 나상완, 박대준, 이재용, 김병철, 광대역 네트워크에서의 혼잡 제어 성능 개선을 위한 ACA-TCP 설계 및 성능 분석, *전자공학회 논문지*, 제 43권 TC편 제 1호, pp. 8-10, 2006. 10.
- [6] 김경희, 김낙명, 에러 보고를 통한 무선 TCP의 성능 향상, *한국통신학회 논문지*, pp. 1656-1658, 2005. 10.
- [7] <http://www.isi.edu/nsnam/ns/>
- [8] Jae Chung, Mark, NS by Example Claypool nile.wpi.edu/NS

### 김 유 두 (金裕斗)



2007년 2월 : 한국기술교육대학교 인터넷공학(공학사)  
 2007년 2월~현재 : 한국기술교육대학교 대학원 정보미디어공학과 (석사과정)

관심분야 : 모바일 인터넷, 모바일 콘텐츠

### 문 일 영 (文日永)



2000년 한국항공대학교 항공통신정보공학과 졸업(공학사)  
 2002년 한국항공대학교 대학원 항공통신정보공학과 졸업(공학석사)  
 2005년 한국항공대학교 대학원 정보통신공학과 졸업(공학박사)

2004년~2005년 한국정보문화진흥원 선임연구원  
 2005년 ~ 현재 한국기술교육대학교 인터넷미디어공학부 전임강사

관심 분야 : 무선 인터넷 응용, 모바일 인터넷, 모바일 IP