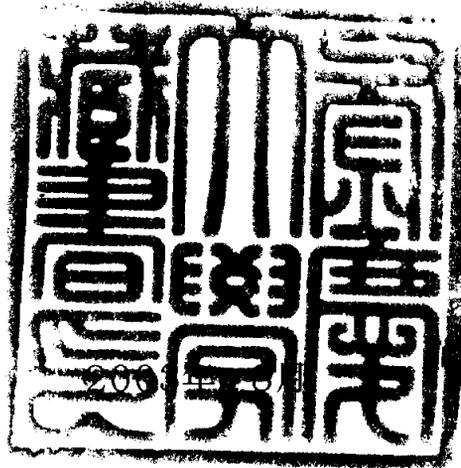


教育學碩士 學位論文

유선망에서의 TCP 혼잡 제어 알고리즘의  
성능비교에 관한 연구

指導教授 朴 勝 燮

이 論文을 教育學碩士 學位論文으로 提出함



釜慶大學校 教育大學院

電算教育專攻

李 幸 南

# 이행남의 교육학석사학위논문을 인준함

2003년 6월 21일

주심 공학박사 김 영 봉



위원 공학박사 여 정 모



위원 공학박사 박 승 섭



# 목 차

그림목차 -----	2
표목차 -----	3
Abstract -----	4
I. 서론 -----	6
II. TCP 메카니즘과 역사 -----	7
2.1 TCP Tahoe 혼잡 제어 알고리즘 -----	10
2.2 TCP Sack 혼잡 제어 알고리즘 -----	11
2.3 TCP NewReno 혼잡 제어 알고리즘 -----	14
III. 시뮬레이션 환경 -----	17
3.1 시뮬레이션 모델 -----	17
3.2 시뮬레이션 파라미터 -----	18
IV. 시뮬레이션 결과 분석 -----	19
V. 결론 -----	24
참고문헌 -----	25

## 그림 목차

<그림1> OSI 7 Layer과 TCP Layer	7
<그림2> Tahoe의 윈도우 변화	10
<그림3> TCP SACK 동작 방법	13
<그림4> NewReno의 동작 원리	16
<그림5> 시뮬레이션 네트워크 구성도	17
<그림6> TCP packet의 처리율	19
<그림7> ACK packet의 처리율	20
<그림8> TCP 처리율의 공정성	21
<그림9> NewReno 패킷의 순서번호	22
<그림10> Sack 패킷의 순서번호	23
<그림11> Tahoe 패킷의 순서번호	23

## 표 목차

<표 1> TCP 버전별 알고리즘의 비교	-----13
<표 2> 모의 실험 파라미터	-----18
<표 3> 패킷 순서 번호로 본 처리율	-----22

# A Study On Performance Comparison of TCP Protocols in Wired Network

Hang-Nam Lee

Graduate School of Education Pukyong National University

## Abstract

Recently, there is an asymmetrical transmission in Internet data stream. The asymmetrical transmission has much more downstream than upstream. Owing to this point, it needs to reduce the acknowledgement as element of the obstruction in downstream.

In this paper, we focus on a throughput of tcp packet in forward link and ack packet in backward link and compared tcp protocol versions in received sequence number and evaluated performance as comparing actually processed packet. As a simulation, we knew that Tahoe is better than Sack and NewReno not only throughput of tcp packet in upstream but also throughput of ack packet in downstream. And Tahoe is better performance than Sack and NewReno in actually processed packet. Therefore, Tahoe is superior to Sack and NewReno throughput of tcp packet in forward and backward wired network. And Tahoe has much more than Sack and NewReno throughput of actually processed packet in received sequence number.

In this point, Tahoe is better than Sack and NewReno downstream as well as forwardstream in wired network. So Tahoe is the most suitable protocol among them in an asymmetrical transmission.

## 1. 서론

최근의 인터넷 이용 현황을 볼 때 TCP 연결에 있어서 비대칭적 흐름을 볼 수 있다는 것은 정보의 순방향으로의 데이터 흐름보다는 역방향으로의 데이터 흐름이 많다는 것이다.

실제로 이러한 사실을 뒷받침해 주고 있는 것으로 국내 초고속 인터넷 시장에서 비대칭기술에 기반해 있는 ADSL의 성장에서 볼 수 있다.

2001년 우리 나라 초고속 인터넷 가입 현황을 보면 ADSL이 418만명으로 전년동기대비 117.8%, 케이블 모뎀이 약 247만명으로 전년 동기대비 88.6% 성장했다. 이처럼 ADSL이 케이블모뎀의 약 2배 정도의 점유율을 보이면서 국내 초고속 시장을 주도하고 있다[5].

이런 최근의 국내외적 상황으로 볼 때 현재 TCP에서도 비대칭링크의 흐름을 적절하게 수용하면서 속도와 성능을 향상시킬 수 있는 TCP 개발이 급선무이라는 것을 알 수 있다.

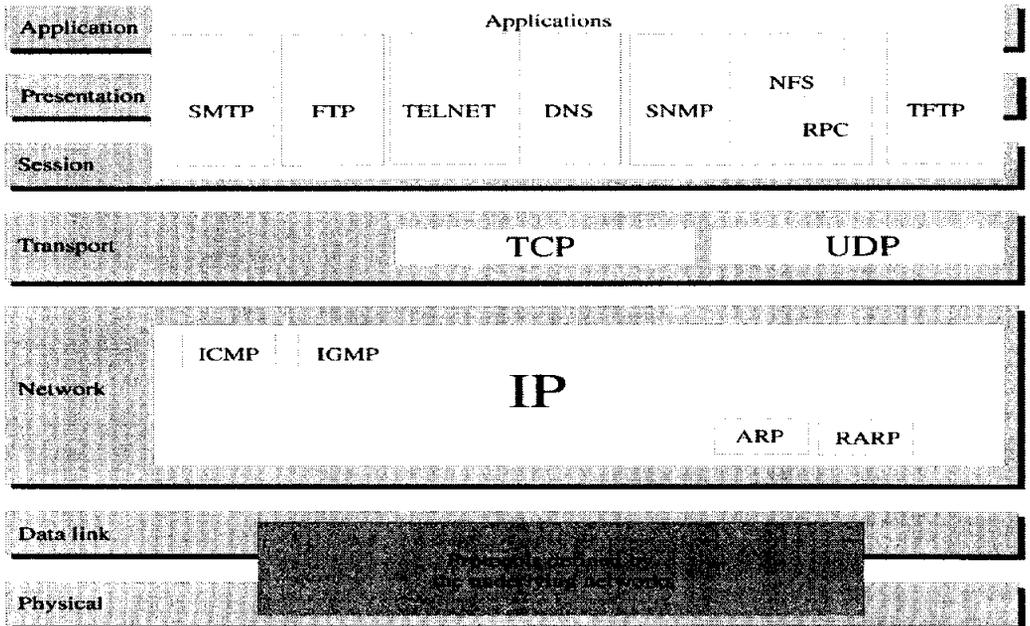
따라서 본 논문에서는 TCP 프로토콜 버전 별로 혼잡 링크에서의 처리율, 처리된 패킷 순서 번호를 통해서 하향링크와 상향링크에서의 실제 전송률을 알아보고 이를 통해서 하향링크에서 가장 적합한 TCP 버전을 알아보았다. 본 논문 내용은 1장 서론에 이어, 2장은 TCP 메카니즘과 역사를 기술하였고, TCP Tahoe version, TCP Sack version과 TCP NewReno version의 특성을 ACK에 초점을 맞추어 설명하였으며 3장에서는 시뮬레이션 환경, 4장에서는 시뮬레

이선 결과 분석, 5장에서는 결론을 서술하였다.

## II. TCP 메카니즘과 역사

TCP는 TCP/IP 인터넷 프로토콜에 의해 제공되는 신뢰성이 보장되는 데이터 전송 서비스 (TCP)이다.

TCP 프로토콜은 신뢰성 있는 전송을 수행하기 위하여 TCP헤더에 송신지 포트와 수신지 포트를 가지고 두 종단 간에 데이터를 실어 나르고 종단간의 대화를 통하여 데이터가 정확히 수신되었는지를 체크하여 전달상의 이상유무를 판단하고 에러가 생겼을 경우나 혼잡이 생겼을 경우 이를 적절히 제어하는 역할을 수행한다.



(그림 1) OSI 7 Layer와 TCP Layer

이러한 TCP프로토콜은 초기에는 신뢰성 없는 네트워크상에서 바이어 스트림의 효율적이고 신뢰성 있는 전송을 위해 타임아웃에 기반을 둔 재전송과 윈도우 흐름제어가 도입되었고 RFC 793에서 정의되었다[9].

두 번째 버전인 TCP Tahoe에서는 혼잡제어와 빠른 재전송이 추가되었다. 세 번째 버전은 TCP Reno로 빠른 회복알고리즘을 포함하는 혼잡제어알고리즘이 확장되었다.

Reno는 RFC2001에서 표준화되었고 RFC2581에서 일반화되었다 [11][12].

TCP Reno는 매우 대중적인 인기를 모았으나 한 윈도우에서 여러 개의 패킷이 유실되었을 때 성능이 크게 떨어지는 문제점을 보였다.

그래서 Newreno, Sack 버전에서는 이 문제를 두 가지 면에서 다르게 접근했는데 Newreno에서는 빠른 회복 구간에서 빠른 재전송을 시도함으로써 한 윈도우내에서 여러 개의 패킷이 유실되었을 때 패킷을 안정적으로 회복하려는 알고리즘을 구사하였다.

반면에 Sack에서는 수신자가 수신된 또는 Queue에 저장된 패킷에 대하여 송신자에게 알려줌으로써 송신자가 유실된 패킷에 대한 정보를 알고 이에 대한 패킷을 재전송하는 기법을 사용한다.

Newreno와 비교할 때 한 윈도우 내에서 여러 개의 패킷이 유실될 때 보다 나은 성능을 발휘한다.

그리고 Fack는 Duplicate ACK를 기다리기까지는 오랜 시간을 걸려야 한다는 점에 착안하여 좀 더 빨리 Selective ACK를 사용할 수

있도록 한 Sack를 개선하여 만든 알고리즘이다.

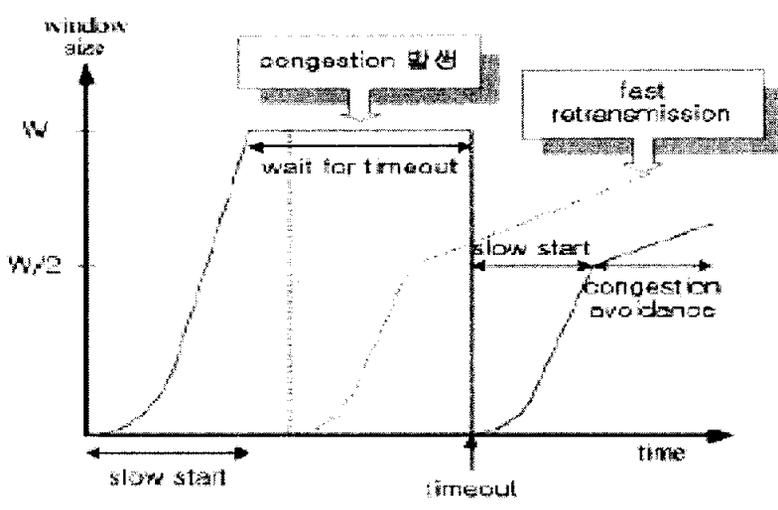
Brakmo와 Peterson에 의해서 새롭게 제안된 손실회피알고리즘이 Vegas로 Vegas는 패킷의 손실을 피하기 위하여 송신자가 송신할 수 있는 패킷의 양을 정확하게 계산하는 방법으로 측정된 RTT를 사용하는 알고리즘이다. 이와 같이 TCP 혼잡제어 알고리즘의 변화는 표1에서 정리하여 비교하였다[14].

TCP algorithms	Loss-recovery/avoidance	Modification Sender/Receiver	Feature Enhanced
Tahoe	Recovery	-	•Slow-Start •Congestion avoidance •Fast retransmission
Reno	Recovery	-	•Fast recovery
SACK	Recovery	Both	•Better estimation of outstanding packet during fast recovery •Earlier fast retransmission
Vegas	Recovery	Sender	•New Congestion avoidance •Modified slow-start mechanism
Newreno	Recovery	Sender	•Response to partial Acks

(표 1) TCP 혼잡 알고리즘 수행의 비교

## 2.1 TCP Tahoe version

TCP Tahoe는 초기에 Slow-Start를 시작하여 패킷을 송수신하다가 혼잡이 발생하면 ssthreshold를 패킷 손실전의 윈도우 크기의 반으로 재설정한 후 혼잡윈도우 크기를 1로 설정하고 혼잡윈도우의 크기를 ssthreshold까지 증가시킨다. 혼잡윈도우의 크기가 ssthreshold까지 증가하면 혼잡회피 단계로 들어간다. 그리고 혼잡이 발생한 네트워크에서 duplicate ACK가 연속적으로 3개가 수신된 경우 빠른 재전송에 들어가고 타임아웃이 발생하면 다시 Slow-Start를 시작한다. TCP Tahoe는 한 윈도우 내에서 처음 패킷의 손실이 발생하면 손실된 패킷을 재전송하고 Slow-Start를 실행하면서 이미 수신된 패킷들을 다시 재전송하므로 수신기가 수신한 패킷을 다시 재전송한다는 문제점이 있고 한 윈도우 내에서 여러 개의 패킷이 손실될 때 대처하는 시간이 늦다[2].



(그림 2) Tahoe의 윈도우 크기 변화

## 2.2 TCP Sack version

TCP Sack protocol에서는 수신자가 제공하는 Selective ACK를 가지고 송신자는 수신자에게 성공적으로 도착한 패킷에 정보를 가지고 실제로 손실된 세그먼트만을 재전송하고 있다.

이와 같은 기능을 수행하기 위한 수신자의 행위는 ① 데이터 송신자가 SYN내에 있는 Sack-permitted Option을 수신자에게 보내면 수신자는 Sack Option을 사용할 수 있게 된다.

② 수신자는 수신된 세그먼트들에 대한 대응으로 송신자에게 ACK를 보내야 하고 “duplicate ACK”도 Sack Option에서 배출하도록 해야 한다.

③ Sack Option을 가지는 ACK는 데이터 수신자의 버퍼 Q에서의 가장 최근의 상황을 반영할 수 있도록 ACK를 유발하는 세그먼트를 갖고 있는 인접 데이터 블록을 명시한다.

④ Sack Option에서 가능한 많은 정확한 Sack Block를 포함해야 하지만 최대로 사용할 수 있는 Sack Option은 적어도 3개의 성공적인 Sack Option이 유지되도록 해야 한다.

Sack Option을 해석하고 재전송하는 데이터 송신자의 행위는 다음과 같이 명시할 수 있다.

① Sack Option을 갖고 있는 ACK를 받았을 때 데이터의 송신자는 미래에 참조하기 위하여 Selective ACK를 기록해둔다.

② 데이터 송신자는 Sack Option을 받기 전에 재패킷화를 수행하려고 할 때 Sack Option과 중복되지 않도록 회피한다.

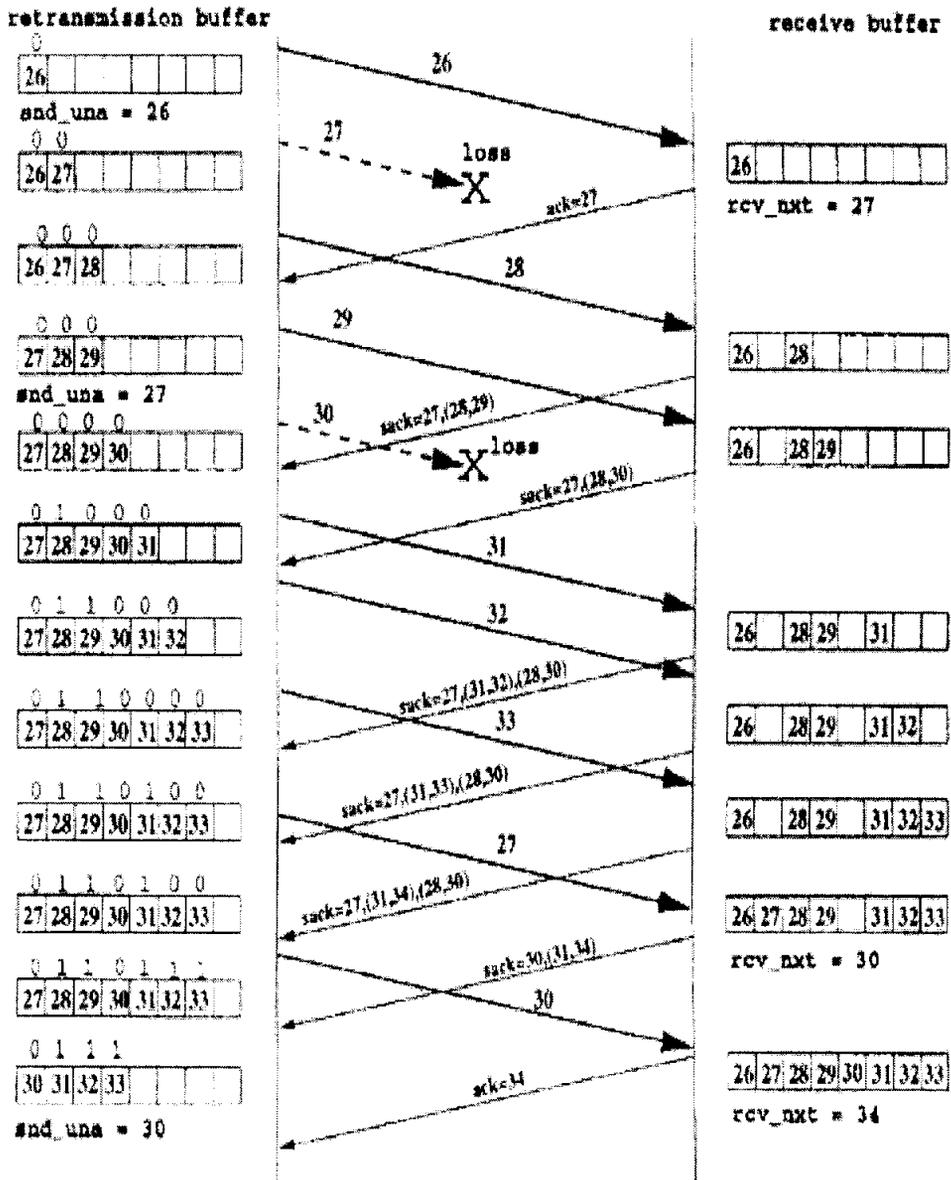
③ 재전송 큐에 있는 Sack Option 블록에 있는 sacked flegs bits는

송신자가 좀 더 나중에 세그먼트들이 전송되어야 할 것임을 알고 전송을 할 것이다.

④ 데이터의 송신자는 재전송타임아웃이 일어나면 모든 sacked bits를 turn off하고 윈도우 왼쪽 가장자리에 있는 세그먼트들을 재전송할 것이다[7].

TCP Sack Protocol은 이와 같은 수신자와 송신자의 행위로 데이터의 신뢰성을 높인다.

특히 무선 구간에서는 패킷의 유실이 많기 때문에 효율적이나 유선 구간에서의 그 성능은 복잡성에 비하여 효용성이 크지 않은 것으로 보이는데 이것은 시뮬레이션에서 검증해 갈 것이다[1].



(그림 3) TCP Sack의 동작원리

## 2.3 The Newreno TCP's Fast Recovery Algorithm

Newreno Algorithm은 Sack가 없을 때 “Partial acknowledgement”로 대응할 수 있게 만들었다.

Partial ACK는 하나의 윈도우에서 다수의 패킷이 유실될 때 송신자는 첫 번째로 빠른 재전송에 들어가고 이 때 이미 전송되었던 패킷에 대한 ACK를 받는데 이를 Partial ACK라고 한다.

만일 거기에 하나의 패킷이 손실되었다면 이 ACK는 빠른 전송에 들어가기 전에 전송되었던 모든 패킷에 대한 ACK가 될 것이고 만일 거기에 다수의 패킷이 손실되었다면 이 ACK는 빠른 전송이전에 전송되었던 모든 패킷이 아니고 약간의 패킷에 대한 ACK가 될 것이다.

Newreno의 “Fast Recovery proudre”는 three Duplicate ACK를 받았을 때 시작하고 재전송타임아웃이나 빠른 회복과정이 시작되었을 때 나타나는 데이터를 포함하여 그 때까지 모든 데이터를 응답하는 ACK가 도달했을 때 끝난다.

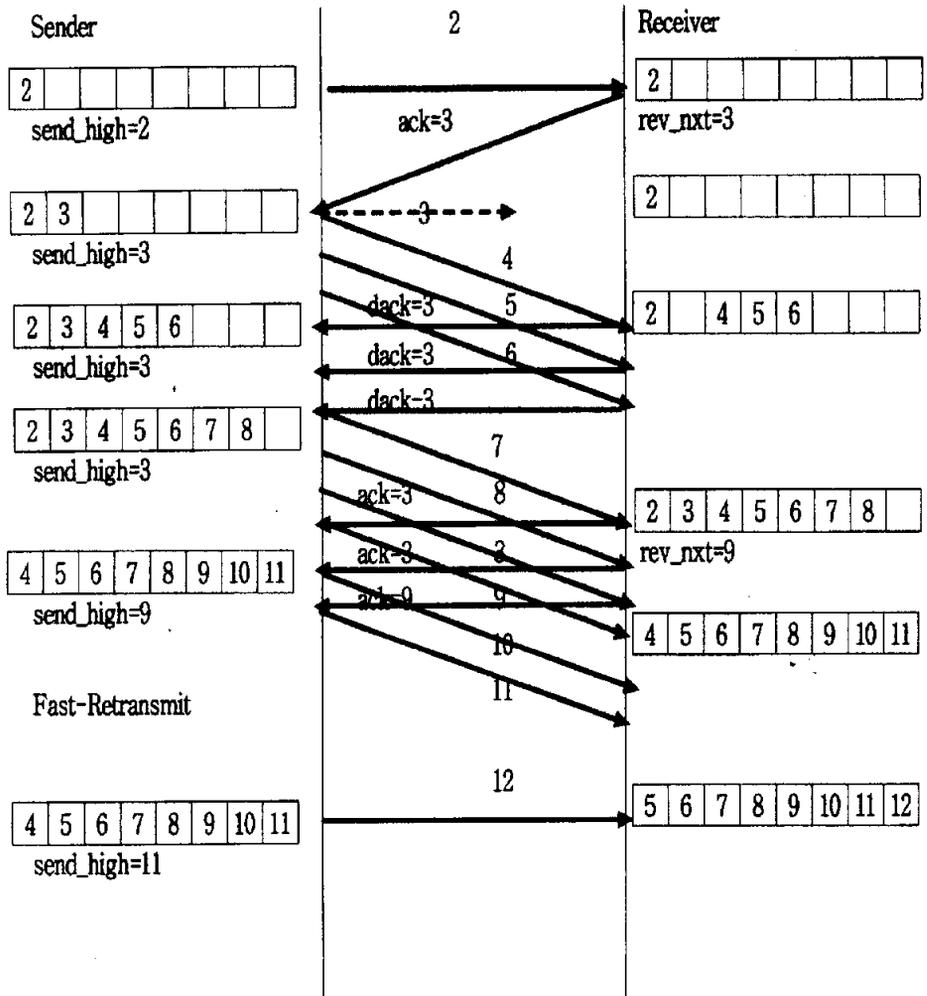
three Duplicate ACK를 받고 송신자가 아직 빠른 회복과정에 들어가지 않았을 때 임계치의 값은  $SStresh = \max(Flightsize/2, 2*MSS)$ 이고 손실된 세그먼트를 재전송하고 혼잡윈도우를  $ssthresh + 3*MSS$ 로 설정한다.

그리고 각각의 부가적인 Duplicate ACK를 받고 있는 동안 MSS에 의해 혼잡윈도우가 증가한다.

만일 혼잡윈도우의 새로운 값이 허락되고 수신자에게 윈도우를 광고했다면 세그먼트를 전송한다.

재전송타이머는 첫 번째 Partial ACK 후에 갱신되고 만일 많은 수의 패킷이 하나의 윈도우 데이터로부터 손실될 때 TCP 송신자의 재전송타이머의 크기는 일반적으로 TRR보다 크지 않기 때문에 작은 수의 패킷이 손실될 때에도 재전송타임아웃이 일어날 수가 있으며 빠른 회복기간에 재전송타임아웃이 일어나면 또 불필요하게 빠른 재전송이 일어날 수 있다. 다시 말하면 데이터 송신자는 모든 데이터가 다 수신되었다는 응답이 있을 때까지 빠른 회복기간에 머무를 수 있다는 것이다.

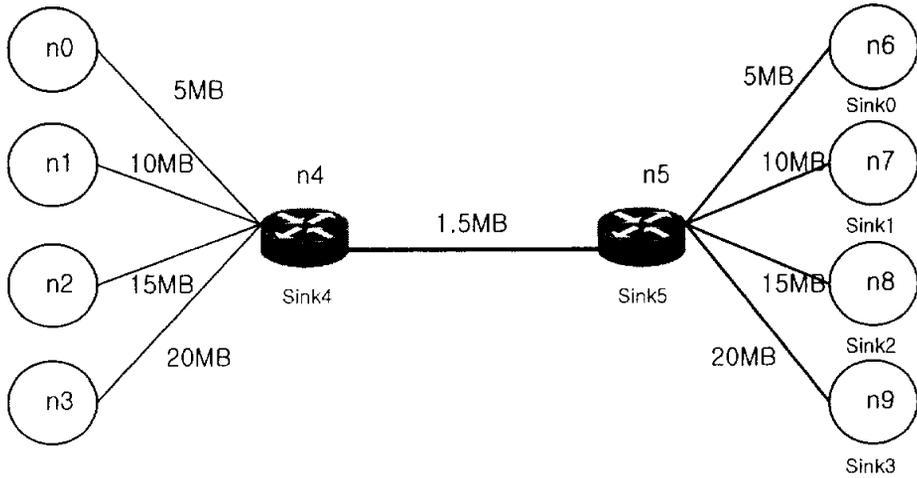
그러므로 빠른 재전송을 회피하기 위한 전략이 필요한데 이것은 세 개의 Duplicate ACK가 왔을 때 Duplicate ACK가 "send\_high"로 들어갈 때 "send\_high"보다 번호가 높은 패킷 순서 번호를 가지면 빠른 재전송에 들어가지 않으며 그보다 낮은 패킷 순서 번호가 왔다면 빠른 재전송에 들어가게 하는 방법을 구사하고 있다[10]. 그러나 여전히 Newreno는 Sack Protocol이 없을 때만 사용할 수 있는 유용한 방법이라는 사실에는 변함이 없다.



(그림 4) TCP Newreno의 동작원리

### III. 시뮬레이션 환경

본 연구에서는 네트워크의 대표적인 시뮬레이션 도구인 ns-2를 가지고 실험하였다[5].



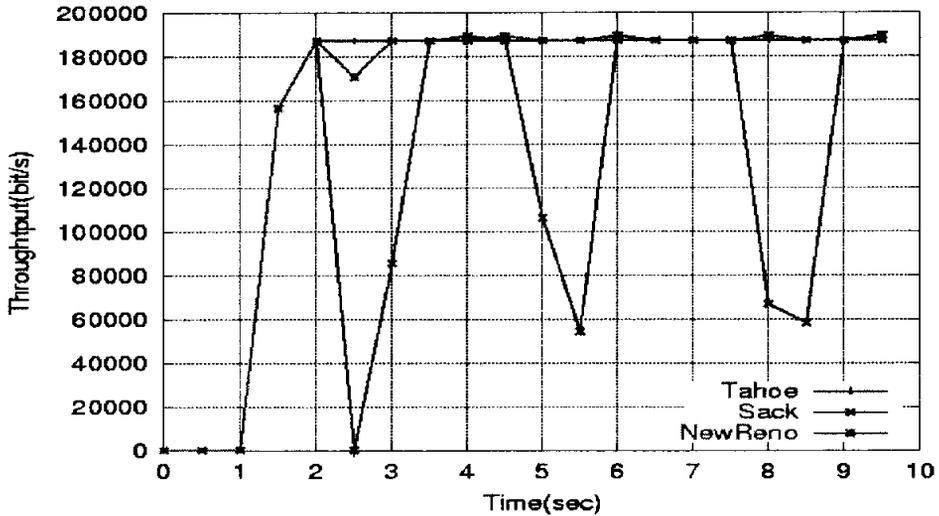
(그림 5) 시뮬레이션 네트워크 구성도

구현된 망에 사용된 파라미터는 다음 표와 같다.

(표 2) 모의 실험 파라미터

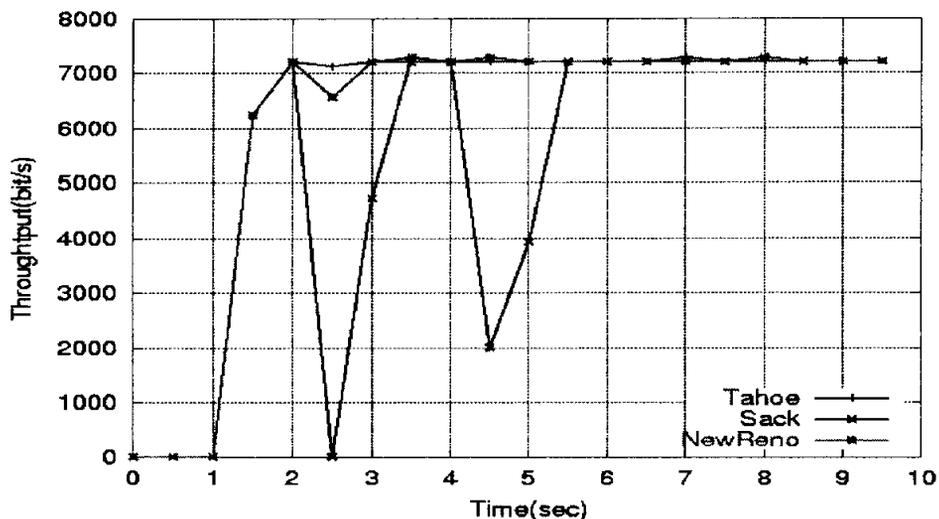
window size	64
packet size	1000byte
application	FTP
휴지시간	2ms
end time	10 seconds
queue	drop tail
선의 종류	duplex-link(양방향)
망의 속도	5MB/10MB/15MB/20MB
TCP Sender	Tahoe/Sack1/NewReno
TCP Reciver	TCPSink0/TCPSink1/TCPSink2/ TCPSink3/TCPSink4/TCPSink5

## IV. 시뮬레이션 결과 분석



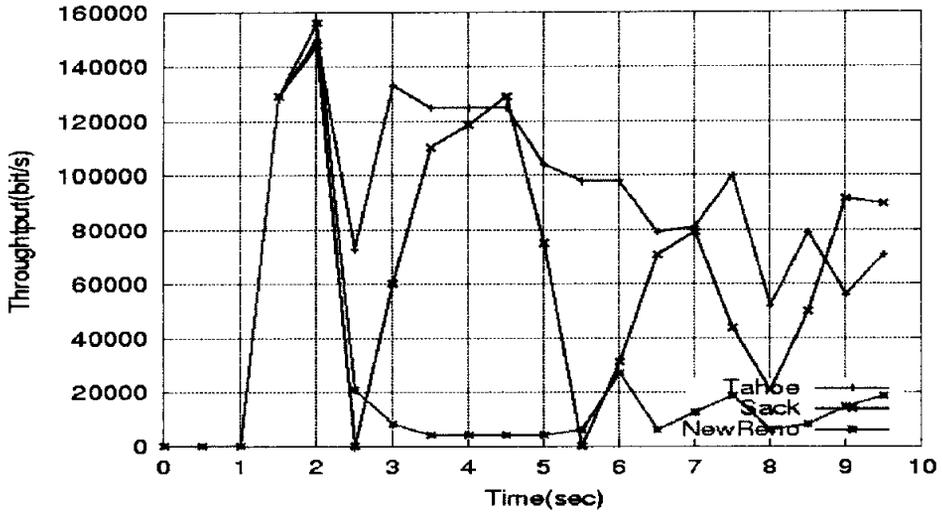
(그림 6) TCP 버전별 전체 처리율

그림 6은 TCP 버전별 전체 처리율을 비교한 것이다. TCP Tahoe가 처리율이 가장 높게 나타났고 TCP Sack가 가장 낮게 나타났다. 유선 구간에서는 유실된 패킷에 대하여 Slective ACK를 사용하거나 Partial ACK를 사용하는 것보다 재전송하는 것이 훨씬 더 효율적이라는 것을 알 수 있다.



(그림 7) TCP 버전별 ACK패킷에 대한 처리율

그림 7은 TCP 버전별 ACK packet에 대한 전체 처리율이다. 역시 Tahoe가 가장 처리율이 높게 나타났고 Sack가 가장 처리율이 낮게 나타났다. Tahoe와 NewReno가 처리율이 비슷하게 나타나고 있는 것을 볼 수 있는데 Tahoe의 경우에는 보내는 패킷이 많은 만큼 이에 대한 응답 패킷이 많았다고 볼 수 있고 NewReno의 경우에는 보낸 패킷에 대한 응답패킷과 Partial Packet으로 말미암아 전체 응답 패킷의 수가 많고 이에 처리된 응답패킷수도 많았다는 것을 알 수 있다.



(그림 8) TCP 버전별 패킷 처리에 대한 공정성 비교

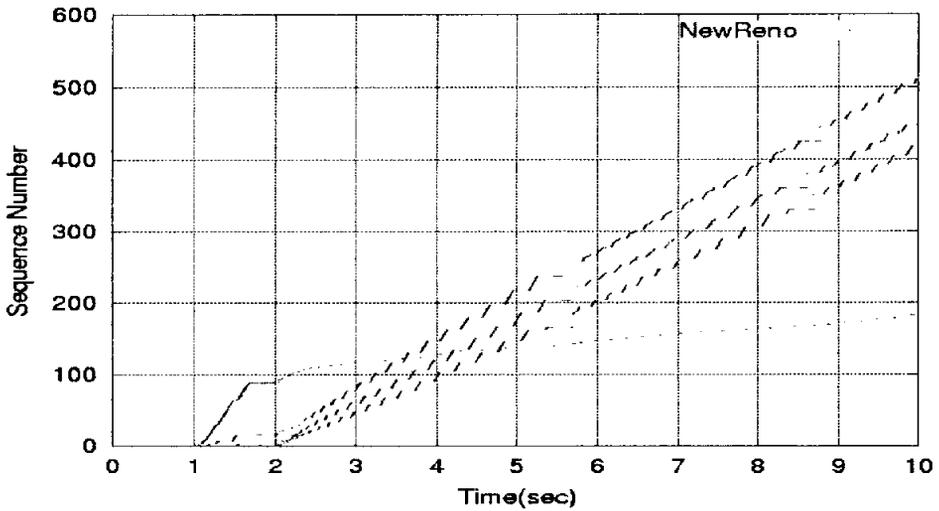
그림 8은 노드 6에서의 패킷처리율을 TCP 버전별로 비교한 것이다.

노드 6에서는 Tahoe가 다른 버전보다 대역폭을 공정하게 사용하고 있으나 다른 노드에서 실험해 본 결과를 종합한다면 노드 별로 약간의 차이는 있으나 전체적으로 큰 차이는 볼 수 없다.

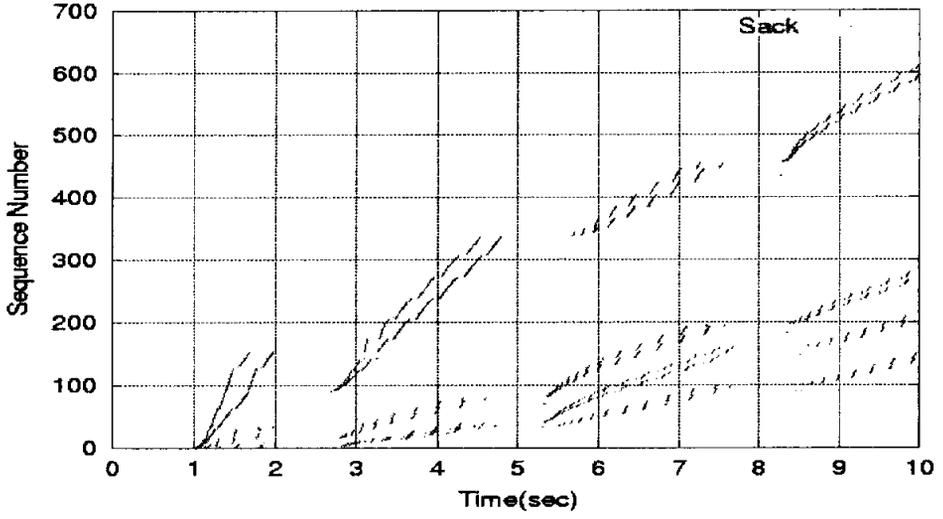
그림 6과 그림 7은 전체 처리율을 알아본 반면 그림 9, 그림 10, 그림 11은 실제로 처리된 패킷의 처리율을 패킷 순서 번호를 통하여 알아본 것이다. 전체 처리율은 Tahoe, NewReno, Sack 순으로 나타났으나 패킷 순서 번호로 본 실제 처리율은 표 3과 같이 Tahoe, Sack, NewReno 순으로 나타났다.

version	Sequence Number	Throughput(byte)
NewReno	512	512,000
Sack	612	612,000
Tahoe	830	830,000

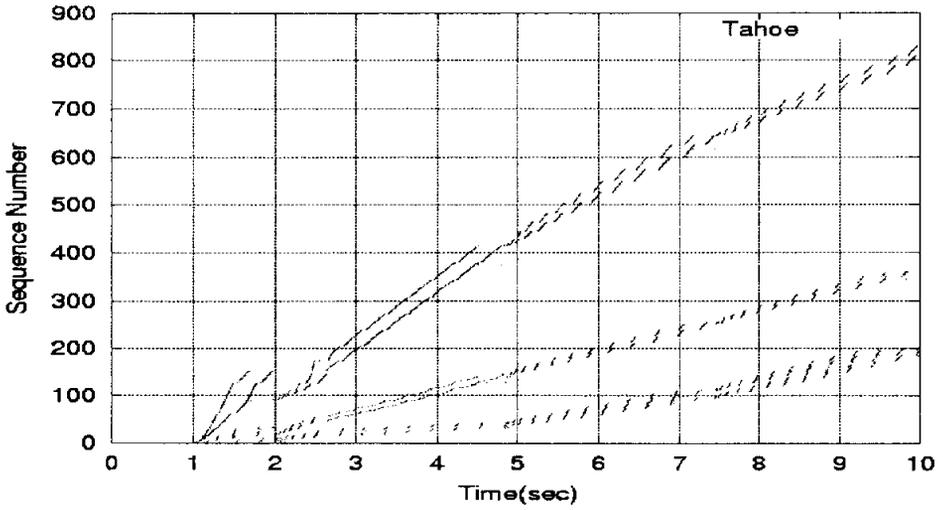
(표 3) TCP 버전의 패킷 순서 번호에 따른 실제 처리율



(그림 9) NewReno의 패킷 순서 번호



(그림 10) Sack의 패킷 순서 번호



(그림 11) Tahoe의 패킷 순서 번호

## VI. 결론

병목 구간을 가진 유선망에서 각 TCP 버전별 처리율을 비교해 본 결과 다음과 같은 사실을 알 수 있었다.

Tahoe가 가장 처리율이 높고 NewReno, Sack 순으로 나타났다. 이러한 결과는 패킷의 손실이 많이 일어나지 않은 유선 구간에서는 Tahoe의 성능이 높다는 것을 보여준 결과라 하겠다. 그리고 실제로 전송된 패킷의 수를 알기 위해서 패킷 순서 번호를 비교해 보면 Tahoe, Sack, NewReno 순으로 나타났다. 실제로 패킷의 손실이 많이 일어나지 않은 유선 구간에서는 Tahoe의 처리 결과가 좋고 Sack는 선별적으로 손실된 패킷을 재전송하고 있기 때문에 NewReno보다 더 많은 패킷을 보낼 수 있었던 것으로 보인다.

그러나 NewReno는 패킷이 손실될 때 마다 즉시 재전송한다고 시간이 많이 소요되어 실제 처리율이 낮게 나온 것으로 보인다.

결론적으로 패킷의 손실이 많이 일어나지 않는 유선 구간에서는 순방향 뿐만 아니라 역방향에서 Tahoe의 성능이 좋은 반면에 패킷의 손실이 많이 일어나는 구간에는 손실된 패킷을 복구하면서 패킷을 전송해야 하므로 NewReno와 Sack가 각각의 ACK packet을 사용함으로 손실된 패킷을 보다 효율적이고 안정적으로 처리할 수 있을 것이다. 그러나 이 같은 경우 역방향에서 패킷의 양이 증가할 수 있으므로 역방향에서의 ACK 패킷을 감소시키는 방안이 마련되어야 한다.

## 참고문헌

- [1] 이상연, 정충교(1998) “유무선 통합망에서의 SACK TCP 프로토콜 성능 개선 방안”, Journal of Telecommunication and Information, Vol.2.
- [2] 이은상(2000-7) “TCP ACK 패킷의 차등 처리에 의한 인터넷 트래픽 성능에 관한 연구”, 한국통신학회논문지, vol.25. No.7B
- [3] Farooq Anjum and Ravi Jain(2000) “Performance of TCP over Lossy Upstream and Downstream Links with Link-level Retransmission”, IEEE
- [4] <http://isis.nic.or.kr>
- [5] <http://www.isi.edu/naman/ns>
- [6] K. Fall, S. Floyd(September, 1996) “Simulation based comparisons of Tahoe, Reno, and SACK TCP”, Computer Communication Review.
- [7] M. Mathis, J. Mahdavi, S. Floyd, A. Ronanow(1996) “TCP select acknowledgment option”, RFC 2018.
- [8] Mark A. Smith and Ramakrishnan, Member(2002) “Formal Specification and Verification of Safety and Performance of TCP Selective Acknowledgment”, IEEE
- [9] Postel. J(September, 1981) “Transmission Control Protocol”, RFC793.
- [10] S. Floyd, T. Henderson(April, 1999) “The NewReno Modification to TCP’s Fast Recovery Algorithm, ” ,RFC 2582.
- [11] Stevens. W(January,1997) “TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms”, RFC2001.

- [12] Stevens. W, Allman. M and V. Paxson(April, 1999) "TCP Congestion Control", RFC 2581.
- [13] Yuan-Cheng Lai and Chang-Li Yao(2001) "TCP Congestion control algorithms and a performance comparison", Computer Communication and Networks.
- [14] Zhu Jing, Li Zhengbin, Niu Zhisheng(Oct, 1999) "A Modified TCP-NewReno Retransmission Scheme for Lossy Network", APCC/OECC'99, p204-208.

## 감사의 글

약 3년 전 저는 학문에 대한 막연한 호기심과 새로운 결의로 교정에 들어섰습니다. 그동안 참 많은 일들이 일어났습니다. 그것이 괴로운 일이었던 기쁜 일이었던 지금은 그것이 중요한 것 같지 않습니다. 중요한 사실은 이제 그 많은 일을 뒤로 하고 졸업이라는 두 단어를 앞두고 있으니 참으로 긴 터널을 하나 빠져 나온 것 같습니다. 지금 생각해 보니 대학원에 와서 배운 정말 중요한 것은 학문이라는 세계에 임하는 자세와 또 학문을 하는 사람들과의 교류라는 것입니다. 그리고 또 생각해 보니 내가 정말 중요하게 생각하여야 할 것은 무엇이었냐는 것입니다. 혹시 공부한다는 핑계로 사람들과의 관계를 소홀히 한 일은 없는지 후회가 됩니다. 공부는 목적이고 목적을 위한 방법에서 이 방법이 정말 최선이었는지 다시 한 번 묻게 됩니다. 그리고 다음에 만일 나에게 공부할 기회가 다시 한 번 주어진다면 정말 공부보다 더 중요한 일을 후회 없이 하면서 공부도 열심히 하고 싶습니다. 지면을 빌어 오늘 이 자리에 있기까지 늘 관심을 가져 주시고 저에게 기대를 걸고 끝까지 지켜봐 주신 박승섭 지도교수님께 먼저 깊은 감사를 드립니다. 그리고 김영봉 교수님의 명쾌하신 논리와 여정모 교수님의 학문에 대한 열정과 높은 인격에 다시 한번 머리 숙여 감사드립니다. 박만곤 교수님의 학문에 대한 끊임없는 추구와 박지환 교수님의 높은 식견에도 많이 배웠습니다.

논문을 쓸 때는 많은 힘이 되어준 육동철 선배, 말없이 큰 그늘이 되어준 전홍진 선배, 같은 동기로 들어와 함께 할 시간이 많았던 영하씨, 공옥춘 선생님, 홍승희 선생님 아마도 잊지 못할 것 같습니다. 연구실 후배로 무엇보다도 내가 많이 의지하고 도움을 청하기도 하여 미안한 경현씨와 학부학생으로 또 필요한 도움을 아낌없이 주었던 형두씨에게 감사드리며, 기석씨, 형기씨, 문경씨, 정원씨에게도 감사드립니다. 끝으로 사랑하는 나의 가족 남편과 두 아들 지훈, 정민에게 아내로서 엄마로서 공부를 핑계로 함께 하는 시

간을 많이 가질 수 없었음을 미안하게 생각하고 또 지금까지 말없이 인내해 준 것에 대해서 감사하며 다시 한 번 저를 아는 모든 분들께 감사드립니다.