

유선망에서의 TCP 프로토콜 버전의 성능 비교 분석

이행남*, 서경현**, 박승섭**

*부경대학교 전산교육학과, **부경대학교 전자계산학과

e-mail : hnam82@yahoo.co.kr

Performance Comparison and Analysis of TCP Protocol Versions in Wired Network

Hang-Nam Lee*, Kyoung-Hyun Seo**, Seung-Seob Park**

*Dept. of Computer Science Education, Pukyong National University

**Dept. of Computer Science, Pukyong National University

요 약

최근의 인터넷에서의 데이터 흐름을 보면 비대칭흐름의 경향이 있다. 비대칭흐름은 주로 하향링크의 데이터 흐름이 많은 것이 특징이며 하향링크에서 데이터흐름을 저해하는 요소인 응답(acknowledgement)을 줄여나가는 기법이 필요하다.

본 논문에서는 순방향에서의 TCP 패킷 처리율과 역방향에서의 ACK 패킷 처리율에 초점을 맞추어 수신된 패킷 순서번호를 TCP 버전별로 관찰해 보고, 실제로 처리된 패킷의 처리율로 비교 평가하였다. 시뮬레이션 결과로, 비대칭링크 유선망에서 Tahoe 버전이 Sack, NewReno보다 성능이 좋다는 것을 알 수 있었다.

1. 서론

최근의 인터넷 이용 현황을 보면 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 초고속인터넷 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 전년동기대비 88.6% 성장했다. 이처럼 ADSL이 케이블모뎀의 약 2배 정도의 점유율을 보이면서 국내 초고속 시장을 이룬 최근의 국내외적 상황으로 볼 때, 현재 TCP에서도 비대칭링크의 흐름을 적절

하게 수용하면서 속도와 성능을 향상시킬 수 있는 TCP 개발이 급선무이라는 것을 알 수 있다[5].

따라서, 본 논문에서는 TCP 버전별로 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 메커니즘과 역사를 기술하였고, TCP Tahoe version, TCP Sack version과 TCP NewReno version의 특성을 ACK에 초점을 맞추어 설명하였으며, 그리고 3장에서는 시뮬레이션 환경, 4장에서는 시뮬레이션 결과 분석, 5장에서는 결론을 서술하였다[3].

2. TCP 메커니즘과 역사

TCP/IP 인터넷 프로토콜에 의해 제공되는 신뢰성이 보장되는 데이터 전송 서비스를 TCP라고 한다.

TCP는 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.

이러한 TCP는 초기에는 신뢰성 없는 네트워크상에서 바이어 스트림의 효율적이고 신뢰성 있는 전송을 위해 타임아웃에 기반을 둔 재전송과 윈도우 흐름제어가 도입되었고 RFC 793에서 정의되었다[10].

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

Reno는 RFC2001에서 표준화되었고 RFC2581에서 일반화되었다[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 기다리가지는 오랜 시간을 걸려야 한다는 점에 착안하여 좀 더 빨리 Selective ACK를 사용할 수 있다는 점에 착안하여 만든 Sack를 개선하여 만든 알고리즘이다.

Brakmo와 Peterson에 의해서 새롭게 제안된 손실회피 알고리즘이 Vagas로 Vagas는 패킷의 손실을 피하기 위하여 송신자가 송신할 수 있는 패킷의 양을 정확하게 계산하는 방법으로 측정된 RTT(Round Trip Time)를 사용하는 알고리즘이다. 이와 같이 TCP 혼잡제어 알고리즘의 변화는 <표 1>에서 정리하여 비교하였다[14].

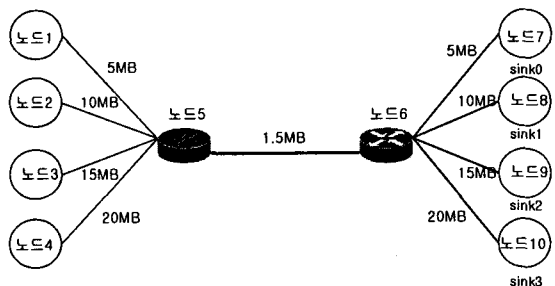
<표 1> Comparison of TCP version algorithm

TCP algorithms	loss-recovery/avoidance	Modification sender/receiver	Features Enhanced
oldTahoe	Recovery	-	· Window-based flow control · Timer Granularity
Tahoe	Recovery	-	· Slow-Start · Congestion avoidance · Fast retransmission
Reno	Recovery	-	· Fast recovery
TCP congestion control	Recovery	No	· General congestion control Scheme of Reno
SACK	Recovery	Both	· Better estimation of outstanding packet during fast recovery · Earlier fast retransmission
Rate-Halving	Recovery	Both	· Smoothing window reduction during fast recovery · Additional updating the TCP state variables after fast recovery
SSDR	Recovery	Sender	· Smooth-start · Dynamic recovery
Vagas	Recovery	Sender	· New congestion avoidance · Modified slow-start mechanism · Earlier packet loss detection
Pseudo-Rate	Recovery	Sender	· New RTT estimation · Direct-drop decrease · Exponential increase
FAck	Recovery	Both	· Better estimation of outstanding packet during fast recovery · Earlier fast retransmission
NewReno	Recovery	Sender	· Response to partial Acks

3. 시뮬레이션 환경

본 연구에서는 네트워크의 대표적인 시뮬레이션 도구인 ns-2를 가지고 실험하였다.[6]

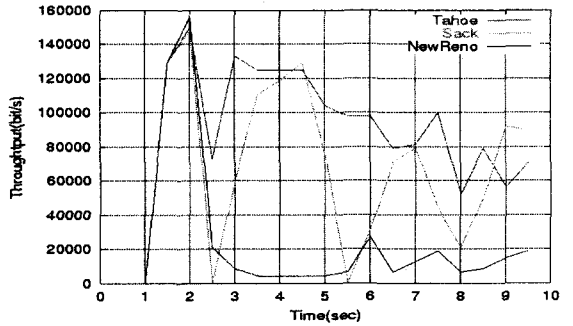
(그림 1)과 같이 구현된 망에 사용된 파라미터는 <표 2>와 같다.



(그림 1) Simulation topology

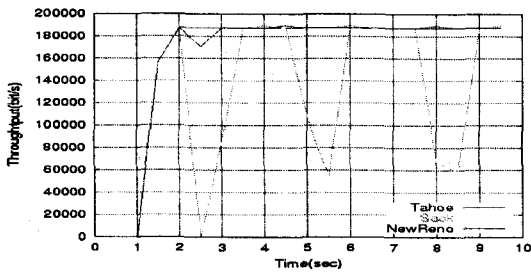
<표 2> Simulation parameter

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



(그림 4) Fairness of TCP version

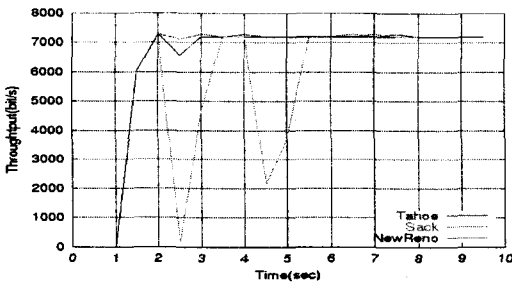
4. 결과 및 분석



(그림 2) Throughput of tcp packet in TCP version

(그림 2) 는 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 효율성이 있다는 것을 알 수 있다.



(그림 3) Throughput of ACK packet in TCP version

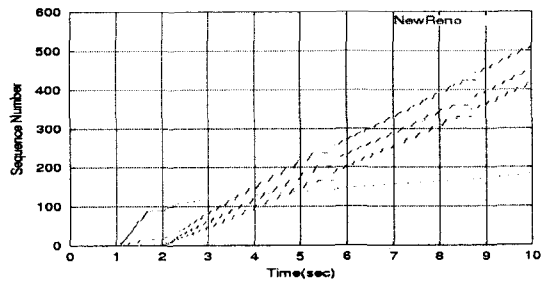
(그림 3)는 TCP 버전별 Ack packet에 대한 전체 처리율이다. 역시 Tahoe가 가장 처리율이 높게 나타났고 Sack가 가장 처리율이 낮게 나타났다.

(그림 4)은 노드 7에서 패킷의 처리율을 TCP 버전별로 비교한 것이다. 노드 7에서는 Tahoe가 다른 버전보다 대역폭을 공정하게 사용하고 있으나 다른 노드에서 실험해 볼 결과를 종합한다면 노드별로 약간의 차이는 있으나 전체적으로 큰 차이는 볼 수 없었다.

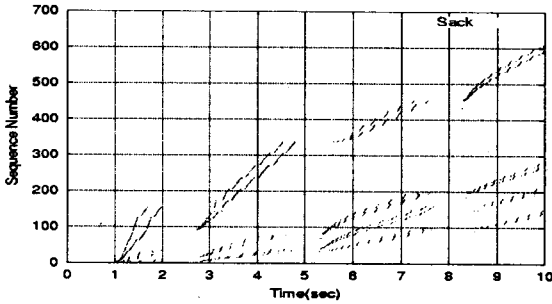
(그림 2), (그림 3)에서는 전체 처리율은 알아본 반면 (그림 5), (그림 6), (그림 7)에서는 실제로 처리된 패킷의 처리율을 패킷 순서 번호를 통하여 알아본 것이다. 전체 처리율은 Tahoe, NewReno, SACK 순으로 나타났으나 패킷 순서 번호로 본 실제 처리율은 <표 3>과 같이 Tahoe, Sack, NewReno 순으로 나타났다.

<표 3> Packet sequence number of TCP version

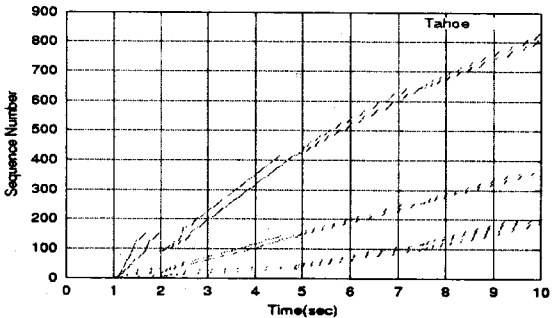
Version	Sequence number	Throughput (byte)
NewReno	512	512,000
Sack	612	612,000
Tahoe	830	830,000



(그림 5) Packet sequence number of NewReno



(그림 6) Packet sequence number of Sack



(그림 7) Packet sequence number of Tahoe

5. 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 사용함으로써 손실된 패킷을 보다 효율적이고 안정적으로 처리할 수 있을 것이다. 그러나 이 같은 경우 역방향에서 패킷의 양

이 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]이행남(2003) "TCP Sack와 NewReno 프로토콜의 성능비교에 관한 연구", 한국해양항만학회지.
- [4] Farooq Anjum and Ravi Jain(2000) "Performance of TCP over Lossy Upstream and Downstream Links with Link-level Retransmission", IEEE
- [5] <http://isis.nic.or.kr>
- [6] <http://www.isi.edu/naman/ns>
- [7] K. Fall, S. Floyd(September, 1996) "Simulation based comparisons of Tahoe, Reno, and SACK TCP", Computer Communication Review.
- [8] M. Mathis, J. Mahdavi, S. Floyd, A. Ronanow(1996) "TCP select acknowledgment option", RFC 2018.
- [9] Mark A. Smith and Ramakrishnan, Member(2002) "Formal Specification and Verification of Safety and Performance of TCP Selective Acknowledgment", IEEE
- [10] Postel. J(September, 1981) "Transmission Control Protocol", RFC793.
- [11] S. Floyd, T. Henderson(April, 1999) "The NewReno Modification to TCP's Fast Recovery Algorithm, ", RFC 2582.
- [12] Stevens. W(January,1997) "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", RFC2001.
- [13] Stevens. W, Allman. M and V. Paxson(April, 1999) "TCP Congestion Control", RFC 2581.
- [14] Yuan-Cheng Lai and Chang-Li Yao(2001) "TCP Congestion control algorithms and a performance comparison", Computer Communication and Networks.
- [15] Zhu Jing, Li Zhengbin, Niu Zhisheng(Oct, 1999) "A Modified TCP-NewReno Retransmission Scheme for Lossy Network", APCC/OECC'99, p204-208.