TCP ACK 패킷의 차등처리에 의한 TCP 종류별 성능에 관한 연구

이은상, 채현석, 최명렬
한양대학교 전자전기제어공학부
e-mail : lio513@asic.hanyang.ac.kr

A Study on TCP Classification Performance by Different Management of TCP ACK Packet

Eun-Sang Lee, Hyun-Seok Chae, Myoung-Ryul Choi
ASIC Lab. Dept of EECS, Hanyang University

요 약

양방향 TCP 연결에서는 ack compression 에 의하여 성능이 저하된다. 이를 해소하기 위한 여러 연구가 진행되고 있으나, TCP 및 인터넷의 사용환경이 워낙 다양하여 뚜렷한 해결책은 없는 상태이다. 특히 TCP 의 종류 및 링크의 속도에 따른 ack compression 에 의한 성능 저하는 다룰 수 밖에 없다. 본 논문에서는 ns-2(network simulator 2)라는 툴(tool)을 사용하여 망의 성능을 측정한 결과로, 우선 단방향과 양방향의 TCP 연결한 망의 성능을 망의 속도별, TCP 종류별로 비교한 후 ack compression 을 적절히 처리하는 방법을 제안하고 제안한 방법에 관한 성능을 또한 망의 속도별, TCP 종류별로 비교하였다.

1. 서론

TCP(Transmission Control Protocol)는 IP(Internet Protocol)와 함께 오늘날 인터넷의 기본으로써 전세계에서 가장 널리 사용되는 transport-layer protocol 이다. 일반적으로 TCP 를 사용하여 망을 형성할 때 단방향의 TCP 를 사용하는 것보다 양방향의 TCP 를 사용하는 것이 망의 성능이 더 좋을 것이라 생각할지 모르나 실제로는 성능이 더 떨어진다. 이렇게 양방향 TCP/IP 연결에 있어서 단방향 TCP 연결보다 성능이 떨어지는 결과를 낳게 하는 근본적인 원인은 ack compression 이다. [1][2]

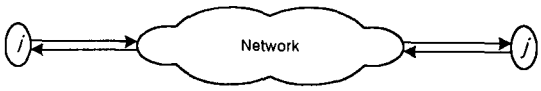
Ack compression 이 무엇인지 간단히 설명해보면 다음과 같다. 대부분의 TCP 에서는 Van Jacobson 이 정의한 혼잡 회피 및 관리 알고리즘(congestion avoidance and control algorithm)을 사용한다. 이는 receiver 에 들어오는 ack 의 속도를 보고, 그에 맞추어 적절한 비율로 데이터를 보내면 bottleneck link 에 넘치지 않게 할 수 있을 것이라는 알고리즘이다. 따라서 TCP sender 는 ack 의 도착에 따라 보낼 데이터의 양을 조절하게 된다. 그런데 만약 데이터를 보내는 도중에 이러한 ack 가 어느 순간 큐에 여러 개가 쌓여 한꺼번에 sender 에 보내지게 되면 sender 는 네트웍이 받을 수 있는 것보다 많은 양의 데이터를 보내게 되어 혼잡상태에 빠지고 망의 성능이 저하되는 것이다. [1]

본 논문에서는 ns-2 를 사용하여, 단방향 TCP 연결 상태일 때와 양방향 TCP 연결 상태일 때의 망의 성능을 측정, 비교하여 양방향 TCP 연결 상태일 때의 망 저하 현상을 보여주고, 이러한 ack compression 에 의한 망 저하 현상이 망의 속도에 따라, 그리고 TCP 의 종류에 따라 얼마만큼의 영향을 미치는 지 알아본 후 ack compression 을 적절히 처리하여 망의 성능을 향상시킬 수 있는 방법을 제안한다.

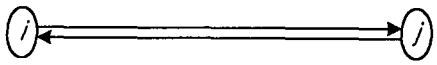
2. 단방향 및 양방향 TCP 연결시의 망의 성능 비교

2-1. 양방향 TCP traffic 모델

본 논문에서 사용한 양방향 TCP 연결 상태일 때의 망의 구성도를 보면 다음의 그림과 같다.



[그림 1] 일반 네트워크에서의 양방향 traffic 모델



[그림 2] 양방향 traffic 의 간략화된 모델

그림에서 [그림 1]은 일반적인 네트워크에서의 양방향 traffic 모델이고, [그림 2]는 양방향 traffic 의 간략화된 모델이다. 또한 i, j 는 각각의 노드를 뜻한다. 양방향 traffic 은 모두 똑같은 조건으로 이루어졌다고 가정한다.

2-2. 시뮬레이션 환경

ns-2 로 위의 모델을 바탕으로 시뮬레이션 할 때의 환경은 다음과 같이 하였다.

- window size : 64
- packet size : 1000byte
- sender 의 application 종류 : FTP
- mss : 10ms
- queue : droptail 방식
- 선의 종류 : duplex-link
- 망의 속도(bps) : 100M, 20M, 10M, 2M, 512K, 56K
- TCP sender 의 종류 : Tahoe, Reno, Sack1(TCP with selective repeat – RFC 2018), Vegas(TCP Vegas), Fack(Reno TCP with "forward acknowledgment")
- TCP receiver 의 종류 : TCPSink(TCP sink with one ACK per packet), DelAck(TCP sink with configurable delay per ACK), Sack1(selective ACK sink – RFC 2018), Sack1/DelAck(Sack1 with DelAck)

2-3. 단방향 및 양방향 TCP 연결시의 시뮬레이션 결과

단방향 및 양방향 TCP 연결시의 라인 속도별, TCP 종류별 망의 성능을 시뮬레이션한 결과는 [표 1]과 같이 나타났다.

[표 1]의 결과에 나타났듯이, 양방향 TCP traffic 의 성능이 단방향 TCP traffic 성능보다 떨어짐을 알 수 있다. 물론 라인 속도가 100M 정도로 아주 빠른 환경에서는 ack compression 의 영향을 받을 조건이 되지 않으므로 성능의 차이가 없다. 그러나 20M~512K 속도에서는 양방향 TCP traffic 의 성능이 떨어지는 것이 확연히 보인다. 이 결과에서 특이했던 것이 라인 속도가 56K 인 조건에서의 성능 측정 결과인데, 이 속도에서는 단방향 TCP traffic 의 성능이 라인의 성능을 따라오지 못했다.

또한 [표 1]의 결과를 보면 단방향 TCP traffic 환경일 때는 TCP sender 의 종류가 TCP Vegas 일 때의 성능이 가장 좋았고, 양방향 TCP traffic 환경일 때는 대체적으로 망의 속도가 100M~2M 일 때는 TCP sender 의 종류가 Fack(Reno TCP with "forward acknowledgment")이고 TCP receiver 의 종류가 Sack1(selective ACK sink – RFC 2018)일 때가 성능이 가장 좋았고, 망의 속도가 512K, 56K 일 때는 TCP sender 의 종류가 Tahoe 나 Fack 이고 TCP receiver 의 종류가 Sack1/DelAck(Sack1 with DelAck)일 때가 성능이 가장 좋았다.

3. TCP ACK 패킷의 차등처리시의 망의 성능 비교

2 의 결과에서도 볼 수 있듯이, 양방향 TCP traffic 의 성능은 ack compression 의 영향에 의하여 단방향 TCP traffic 의 성능보다 떨어진다. 이를 해결하기 위하여 본 논문에서는 TCP ack 패킷의 차등처리방법을 제안한다.

TCP ack 패킷의 차등처리방법은 다음과 같다.

일반적으로 ack 패킷은 다른 data 패킷과 똑같이 들어오는 순서대로 큐에 쌓였다가 보내지는데 이렇게 ack 를 일반 data 패킷과 동등하게 처리하는 것 때문에 ack compression 이 생기는 것이다. 이를 최대한 방지하기 위하여 ack 패킷에 우선순위를 주어 큐에 들어오는 즉시 먼저 보내지도록 한다면, 여러 개의 ack 가 큐에 쌓여 data 패킷이 다 보내질 때까지 기다렸다가 한꺼번에 보내지는 일은 없을 것이다.

바로 이러한 방법을 이용하여 본 논문에서는 2 의 시뮬레이션 환경에 queue 만 ack 에 우선순위를 주는 것으로 바꾸어 다시 시뮬레이션 해보았다.

이것의 결과는 다음의 [표 2]에 나와있다.

이 표를 보면 알 수 있듯이, 역시 100M 의 속도정도로 아주 빠른 환경에서는 변화가 없었으나 [표 1]의 결과에서처럼 ack compression 의 영향을 많이 받은 2M~516K 의 속도에서는 ack 차등처리한 쪽의 성능이 더 좋아졌음을 볼 수 있다. 그러나 여기에서도 [표 1]의 결과에서처럼 56K 의 속도에서는 특이하게도 성능이 더 떨어지는 결과를 얻게 되었다.

또한 [표 2]를 보면, 100M 의 속도에서는 TCP receiver 의 종류가 TCPSink 일 때와 Sack1 일 때 가장 성능이 좋았고, 20M 의 속도에서는 TCP sender 의 종류가 Fack 일 때, TCP receiver 의 종류가 TCPSink 일 때와 Sack1 일 때가 가장 성능이 좋았다. 10M 의 속도에서는 TCP receiver 의 종류가 DelAck 혹은 Sack1/DelAck 일 때 대부분 성능이 좋았고, 2M 의 속도에서는 TCP sender 의 종류가 Vegas 일 때, TCP receiver 의 종류가 DelAck 혹은 sack1/DelAck 일 때 가장 성능이 좋았다. 512K 의 속도에서는 TCP receiver 가 Sack1/DelAck 일 때 대부분 성능이 좋았고, 56K 일 때는 TCP ack 패킷을 차등처리하지 않았을 때가 성능이 더 좋았다.

4. 결론 및 향후과제

본 논문에서는 양방향 traffic 환경에서의 ack compression 의 영향으로 인한 성능 저하가 다양한 환경 요소에 의하여 다르게 나타난다는 것을 ns-2 를 이용한 시뮬레이션으로 확인하고 그 성능을 TCP 종류별, 그리고 망의 속도별로 모두 확인해 보았다.

또한 이러한 ack compression 의 영향을 줄이기 위하여, 큐에서 ack 를 차등처리하는 방법을 제안하고 이 방법을 사용하여 TCP 종류별, 망의 속도별로 시뮬레이션하여 그 성능을 모두 확인해 보았다. 그리하여 양방향 TCP traffic 환경일 때는 망의 속도별로 sender 와 receiver 를 어떤 종류를 사용해야 가장 높은 성능을 얻을 수 있는지를 알아볼 수 있었다.

이번 시뮬레이션에서 가장 특이했던 점이 바로 망의 속도가 56K 일 때의 성능이었는데, 다른 망의 속도에서 볼 수 있었던 일반적인 예상과는 달리 특이한 결과가 나온 이유가 무엇인지 알아보는 것은 향후과제로 하겠다. 또한 이렇게 ack 패킷의 차등처리방법을 사용했을 때 각 인터넷 서비스의 성능은 어떻게 변하게 될 것인지도 향후 알아볼 것이다.

참고문헌

[1] Jeffrey C. Mogul. "Observing TCP Dynamics in Real Networks", ACM SIGCOMM'92, Aug. 1992, pp. 305-317

[2] Lampros Kalampoukas, Anujan Varma. "Two-Way TCP Traffic over Rate Controlled Channels: Effects and Analysis", IEEE ACM TRANSACTION ON NETWORKING, Vol 6, December 1998, pp.729-743

[3] ns Notes and Documentation –The VINT Project, Feb. 25 2000

[4] http://www.isi.edu/nsnam/ns/

(단위 : Kb/s)

		TCPSink		DelAck		Sack1		Sack1/DelAck	
		단방향	양방향	단방향	양방향	단방향	양방향	단방향	양방향
Tahoe	100M	25497.6	25497.6	25395.2	25395.2	25497.6	25497.6	25395.2	25395.2
	20M	20000	14387.2	20000	14372	20000	14387.2	20000	14372
	10M	10000	7585.6	10000	7592.8	10000	7571.2	10000	7952.8
	2M	1981.6	1857.6	1984.8	1858.4	1981.6	1857.6	1985.6	1858.4
	512K	510.4	492.8	512	501.6	510.4	492.8	512	502.4
	56K	24.8	53.6	24.8	54.4	24.8	53.6	24.8	54.4
Reno	100M	25497.6	25497.6	25395.2	25395.2	25497.6	25497.6	25395.2	25395.2
	20M	20000	14387.2	20000	14372	20000	14387.2	20000	14372
	10M	10000	7602.4	10000	7852.8	10000	7604	10000	7852.8
	2M	2000	1487.6	2000	1877.6	2000	1415.2	2000	1878.4
	512K	512	492.8	512	481.2	512	492.8	512	481.6
	56K	24.8	53.6	24.8	54.4	24.8	53.6	24.8	54.4
Sack1	100M	25497.6	25497.6	25395.2	25395.2	25497.6	25497.6	25395.2	25395.2
	20M	20000	14387.2	20000	14372	20000	14387.2	20000	14372
	10M	10000	6639.2	10000	7845.6	10000	7840	10000	7848
	2M	2000	1353.6	2000	1876	2000	1876	2000	1873.6
	512K	512	492.8	512	483.2	512	492.8	512	502.4
	56K	24.8	53.6	24.8	54.4	24.8	53.6	24.8	54.4
Vegas	100M	25497.6	25497.6	25395.2	25395.2	25497.6	25497.6	25395.2	25395.2
	20M	20000	10565.6	20000	10564.8	20000	10565.6	20000	10564.8
	10M	10000	5606.4	10000	5606.4	10000	5606.4	10000	5606.4
	2M	2000	1467.2	2000	1405.6	2000	1467.2	2000	1405.6
	512K	512	417.6	512	437.6	512	417.6	512	437.6
	56K	56	46	56	53.6	56	46	56	53.6
Fack	100M	25497.6	25497.6	25395.2	25395.2	25497.6	25497.6	25395.2	25395.2
	20M	20000	14387.2	20000	14372	20000	14387.2	20000	14372
	10M	10000	6377.6	10000	6867.2	10000	7897.6	10000	7855.2
	2M	1773.6	1364.4	1880.8	1556.8	2000	1879.2	2000	1876
	512K	427.2	492.8	512	501.6	512	492.8	512	502.4
	56K	24.8	53.6	24.8	54.4	24.8	53.6	24.8	54.4

[표 1] 단방향 및 양방향 TCP 연결시 망의 성능 비교

(단위 : Kb/s)

		TCPSink		DelAck		Sack1		Sack1/DelAck	
		일반	우선순위	일반	우선순위	일반	우선순위	일반	우선순위
Tahoe	100M	25497.6	25497.6	25395.2	25395.2	25497.6	25497.6	25395.2	25395.2
	20M	14387.2	19231.2	14372	19607.2	14387.2	19231.2	14372	19607.2
	10M	7585.6	9615.2	7592.8	9804	7571.2	9615.2	7952.8	9804
	2M	1857.6	1905.6	1858.4	1947.2	1857.6	1905.6	1858.4	1947.2
	512K	492.8	490.4	501.6	501.6	492.8	490.4	502.4	502.4
	56K	53.6	26.4	54.4	27.2	53.6	26.4	54.4	27.2
Reno	100M	25497.6	25497.6	25395.2	25395.2	25497.6	25497.6	25395.2	25395.2
	20M	14387.2	19231.2	14372	19607.2	14387.2	19231.2	14372	19607.2
	10M	7602.4	9615.2	7852.8	9804	7604	9615.2	7852.8	9804
	2M	1487.6	1923.2	1877.6	1959.2	1415.2	1922.4	1878.4	1959.2
	512K	492.8	492.8	481.2	502	492.8	492	481.6	501.6
	56K	53.6	26.4	54.4	27.2	53.6	26.4	54.4	27.2
Sack1	100M	25497.6	25497.6	25395.2	25395.2	25497.6	25497.6	25395.2	25395.2
	20M	14387.2	19231.2	14372	19607.2	14387.2	14387.2	14372	19607.2
	10M	6639.2	9615.2	7845.6	9804	7840	7840	7848	9804
	2M	1353.6	1923.2	1876	1869.6	1876	1876	1873.6	1965.6
	512K	492.8	492.8	483.2	501.6	492.8	492.8	502.4	502.4
	56K	53.6	26.4	54.4	27.2	53.6	53.6	54.4	27.2
Vegas	100M	25497.6	25497.6	25395.2	25395.2	25497.6	25497.6	25395.2	25395.2
	20M	10565.6	16252	10564.8	19607.2	10565.6	16252	10564.8	19607.2
	10M	5606.4	7467.2	5606.4	8240	5606.4	7467.2	5606.4	8240
	2M	1467.2	1923	1405.6	1960.8	1467.2	1923.2	1405.6	1960.8
	512K	417.6	492	437.6	502.4	417.6	492	437.6	502.4
	56K	46	54.4	53.6	53.6	46	54.4	53.6	53.6
Fack	100M	25497.6	25497.6	25395.2	25395.2	25497.6	25497.6	25395.2	25395.2
	20M	14387.2	19231.2	14372	19607.2	14387.2	19231.2	14372	19607.2
	10M	6377.6	9615.2	6867.2	9804	7897.6	9615.2	7855.2	9804
	2M	1364.4	1710.8	1556.8	1833.6	1879.2	1922.4	1876	1959.2
	512K	492.8	402.4	501.6	502.4	492.8	492	502.4	502.4
	56K	53.6	26.4	54.4	27.2	53.6	26.4	54.4	27.2

[표 2] 일반 및 TCP ACK 패킷의 차등처리시 망의 성능 비교