

TCP를 사용한 위성망에서의 성능 향상 알고리즘

김상희, 이정규
한양대학교 전자계산학과

Algorithm for Performace Enhancement of Satellite Link using TCP

Sang Hee Kim and Jong Kyu Lee
Dept. of Computer Science Engineering, Hanyang University

Abstract

본 논문에서는 TCP를 사용한 위성망에서의 문제점을 도출하고 그에 대한 성능향상 알고리즘을 제안하였다. 긴 전파 지연과 높은 에러율을 지닌 위성링크는 TCP의 성능을 현저히 떨어뜨린다. 특히 TCP에서는 에러에 의한 세그먼트 손실도 Congestion에 의한 손실로 판단하여 Slow Start 나 Congestion Avoidance 알고리즘을 적용시킨다. 따라서 본 논문에서는 에러에 의한 손실과 Congestion의 손실을 구분해 주는 ENA(Error Notification Ack) 알고리즘을 제안했다. 그리고 위성에서 좋은 성능을 나타내는 SACK TCP에 이 알고리즘을 적용하여 성능을 평가했다.

1. 서론

주로 문자 형태의 저 트래픽이 주를 이루었던 인터넷이 멀티미디어 서비스를 제공하기 위해 음성, 데이터, 비디오 등 여러 종류의 트래픽 형태로 변환되고 있다. 이러한 멀티미디어 서비스를 만족하기 위해 위성망을 이용한 인터넷 서비스가 관심의 대상이 되고 있다. 그러나 위성환경은 기존의 지상망과는 다른 특징을 가지고 있다. 지상

망보다 상대적으로 긴 전파지연시간과 위성과 같은 무선환경에서의 높은 에러율은 TCP Congestion Control 알고리즘에 영향을 준다.

본 논문의 구성은 다음과 같다. 2장에서는 TCP Congestion Control에 관하여 간단히 설명하였고

3장에서는 성능향상을 위한 표준기법들을 살펴보았다. 4장에서는 새로운 알고리즘을 제안하고 5장에서는 시뮬레이션을 통해 성능평가를 수행했다. 마지막으로 6장에서는 결론을 맺었다.

2. TCP의 Congestion Control

TCP는 인터넷 서비스(SMTP, HTTP, FTP등)를 전송하기 위해서 사용하는 프로토콜로써 신뢰성과 연결성을 갖는 전송 프로토콜이다. 송신측이 수신측으로부터 바로 이전 세그먼트의 Ack을 받아야 새로운 세그먼트를 보낼 수 있도록 Flow Control를 한다. 그리고 TCP는 효율적인 전송을 위해 Congestion Control 알고리즘을 사용한다. 만일 TCP가 빠른 속도로 세그먼트를 전송하면 네트워크상에 존재하는 라우터에 과부하를 발생시키고 이는 Congestion을 일으킨다. 이를 방지하기 위해서 TCP는 세그먼트의 전송 속도를 Congestion Control 알고리즘을 사용하여 조절한다[1].

2.1 Slow Start and Congestion Avoidance

과거의 TCP에서는 송신측이 다수개의 세그먼트를 수신측의 Advertised Window Size만큼 보내는 것을 기본으로 했다. 하지만 저속의 링크가 연결되어 있을 경우, 세그먼트들이 라우터에 대기되는 현상이 발생할 수 있다. 따라서 새로운 세그먼트를 보내고자 할 때는 수신측에서 보내는 Ack의 비율에 맞추어서 보내고자하는 것이다. 네트워크에서의 Congestion은 입력이 출력보다 많은 라우터에서 다수개의 입력이 들어왔을 때 발생하게 된다. 기본동작은 다음과 같다. 수신측에서 하나의 Ack을 받으면 송신측은 두 개의 세그먼트를 보낸다. 따라서 Slow Start Phase에서 네트워크로 보내는 세그먼트의 양이 지수적으로 증가하게 된다. 세그먼트의 양이 많아지면

Congestion이 발생할 수 있는데 이를 방지하기 위한 알고리즘이 Congestion Avoidance이다. $cwnd$ 와 $ssthresh$ 두 개의 변수를 가지고 송신측의 윈도우 크기를 조절하여 TCP의 전송속도를 낮추는 것이다. Connection이 설정될 때 $cwnd=1$, $ssthresh = Receiver Window Size$ 로 setting이 된다. Slow Strat Phase에서는 하나의 Ack이 도착할 때마다 $cwnd$ 값이 1씩 증가한다. 그리고 $cwnd$ 이 $ssthresh$ 값보다 커지면 즉, Congestion Avoidance Phase로 들어간다. 이 Phase에서는 Ack에 대해서 $cwnd$ 는 $\frac{1}{cwnd}$ 만큼씩 증가하게 된다. 즉, Congestion Avoidance 알고리즘에서의 세그먼트의 양은 선형증가를 나타나게 된다[2].

2.2 Fast Retransmit and Fast Recovery

Fast Retransmit의 목적은 한 개의 손실된 세그먼트를 RTO(Retransmit Timeout)전에 발견하고 Retransmit함으로써 성능을 향상시키고자 하는 것이다. 즉, 송신측이 중복되는 Ack을 3번 받으면 그 세그먼트는 손실된 것으로 판단하고 재전송을 한다. Fast Retransmit 알고리즘을 사용하면 Fast Recovery 알고리즘을 같이 사용하게 된다. Fast Recovery 알고리즘은 3개의 중복 Ack을 받으면 $cwnd$ 를 1로 줄이지 않고 $ssthresh$ 의 반값으로 setting한다[1]

2.3 위성망에서 TCP성능의 문제점

위성은 수십 Mbps에서 수 Gbps의 링크속도를 제공할 수 있으나 동시에 위성구간은 긴 전파지연시간을 갖는다. 원래의 TCP 표준은 TCP 수신 윈도우를 64 Kbyte로 제한한다[3,4,5]. TCP의 수신윈도우 크기는 TCP연결의 최대 전송률이 왕복시간에 의해서 제한되는 위성환경에서는 특히 중요하다. 즉, 최대 전송률은 다음과 같다.

$$throughput_{max} = \frac{receive\ buffer\ size}{round\ trip\ time}$$

정지위성의 경우 다음과 같이 성능이 제한된다.

$$\text{throughput}_{\max}(\text{satellite}) = \frac{64\text{Kbytes}}{590\text{ms}}$$

$$\approx 112,000 \frac{\text{bytes}}{\text{second}} \approx 896,000 \frac{\text{bits}}{\text{second}}$$

즉, T1(1.544Mbps)의 대역폭을 갖는 위성링크를 사용해도 기존의 TCP에서는 896Kbps의 전송률 밖에 얻을 수 없다.

그리고 위성망에서 제공하는 무선 링크는 지상망에 비해 높은 BER(Bit Error Rate)을 나타낸다[6]. 이런 높은 에러율은 TCP성능을 제한한다. TCP는 에러에 의한 손실을 Congestion으로 인한 손실로 해석하여 전송률을 낮춘다. 즉, Slow Start Phase부터 다시 시작하거나 Congestion Avoidance Phase로 동작하는 Fast Recovery 알고리즘이 수행된다.

3. TCP성능향상을 위한 표준 기법

TCPSAT WG(Working Group)에서는 기존의 표준들을 이용하여 위성망에서의 TCP 성능향상이 가능한 기법들을 정리하여 문서화하였다[5]. 이들 중 TCP 계층에서는 Window Scale Option(RFC 1323)[7], SACK(RFC 2018)[8]이 있다. Window Scale Option은 높은 DB(Delay X Bandwidth)를 갖는 환경에서 윈도우 크기를 증가시켜 TCP가 고대역폭 링크를 모두 활용할 수 있도록 한다. 그리고 SACK은 한 Window내에서 여러개의 세그먼트 손실이 발생했을 때 블록화된 Ack정보를 송신측이 수신할 수 있으므로 재전송이 용이하도록 한다.

4. ENA(Error Notification Ack) 알고리즘 제안

위의 2, 3을 통해서 위성망에서의 TCP 성능에 대해 자세히 알아보았다. 지상망의 경우 에러에 의한 세그먼트 손실은 전체 손실의 1% 정도밖에 차지하지 않기 때문에 에러에 의한 세그먼트 손

실을 Congestion에 의한 손실이라고 가정하고 알고리즘을 적용시켰다[6]. 하지만 위성의 경우는 Error에 의한 손실들이 많아짐으로써 이러한 손실을 Congestion으로 여기고 Congestion Control 알고리즘을 적용하는 것은 성능을 저하시킨다[6]. 이 논문에서는 이러한 문제점을 해결하여 성능향상을 시킬 수 있는 알고리즘을 제안한다. 기본적으로는 SACK을 적용하는 TCP를 고려한다. SACK은 다수의 세그먼트 손실이 발생하는 경우에 좋은 성능을 나타낸다. SACK Option Format에 제대로 받은 세그먼트 정보와 손실된 정보를 포함하고 있기 때문에 하나의 RTT동안 여러개의 손실된 세그먼트를 재전송함으로써 성능을 향상시킬 수 있는 것이다. 만약 그런 정보를 표현할 수 있는 크기가 40 Bytes면 N blocks의 길이 = $8 \times N + 2$ Bytes의 식에 의해 최고 4개의 block까지 포함할 수 있다[8].

SACK option Format은 표1과 같다.

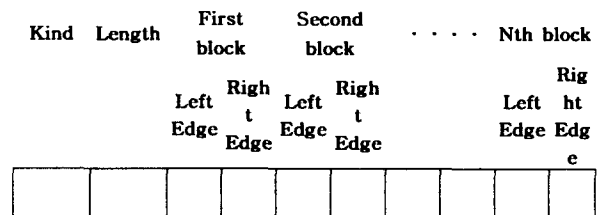


표 1. SACK Option Format

따라서 Ack은 Expected Number와 N block의 정보를 갖고 있다. 따라서 새로운 알고리즘에서는 Ack의 Format에 다음과 같은 정보를 포함한다. Expected Number와 손실된 세그먼트와 제대로 수신된 세그먼트의 정보를 포함하고 있는 3개의 Block과 세그먼트가 도착했을 때 Error인가 아닌가를 표현해 주는 Bit와 에러인 해당 세그먼트번호를 갖고 있도록 구성했다. 따라서 이 Ack의 정보를 받은 송신측은 에러를 표현해 주는 Bit를 우선 조사한다. 수신측이 에러 세그먼트를 받았다는 정보를 송신측이 확인하면 Ack 정보에 있는 Error인 세그먼트만을 재전송한다.

대신 Congestion이 발생했을 때처럼 전송률을 줄이지 않는다. 따라서 그에 따른 성능저하를 막을 수가 있는 것이다.

5. 시뮬레이션을 통한 성능평가

위성망에서의 TCP 성능을 분석하기 위해 시뮬레이션 전용언어인 Simscript II.5를 사용했다. [그림1]은 성능평가를 위한 Model이다.

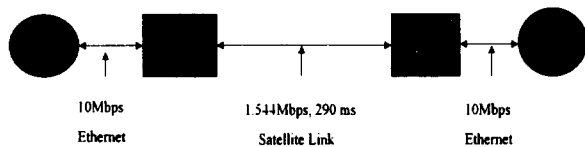


그림 1. 시뮬레이션 모델

지상망으로는 10Mbps의 Ethernet으로 잡고 위성링크는 T1의 대역폭으로 설정하였다. 그리고 위성의 전파지연시간은 290 ms로 잡고 Ethernet의 전파지연시간은 위성전파지연시간에 비해 아주 작기 때문에 없는 것으로 가정했다.

파라미터	값
TCP Segment	512 Bytes
Satellite Router Buffer Size	100
Maximum window Size	64 KBytes, 110 KBytes
Minimum Slow-Start threshold	1

표 2. 시뮬레이션 파라미터

표2는 기본적인 TCP의 파라미터들이다. 위성의 링크손실은 10^{-5} , 10^{-6} , 10^{-7} 의 세 가지 경우를 시뮬레이션 했다. Test File Size는 200K, 500KB, 1MB, 5MB, 10MB이다[9]. 그리고 TCP Version은 SACK을 선택했다. 그리고 Maximum Window Size은 표준 TCP에서 사용되는

64Kbytes와 Window Scale Option을 사용한 110Kbytes의 두 가지 경우에 대해 실험했다. 그리고 에러인 세그먼트는 반드시 수신측에 도달한다는 가정한다. 그리고 송신측에는 Ack을 받았을 때 전송해야 될 세그먼트가 항상 존재한다고 가정한다. 송신측에서 위성으로 가는 라우터를 기준으로 받는 대역폭과 보내는 대역폭을 다르게 설정하고 유한개의 Buffer를 설정함으로써 Congestion이 발생하도록 Modeling하였다.

5.1 RENO와 SACK의 성능평가

T1의 위성링크속도, 110 KBytes의 Large Window의 환경에서 RENO와 SACK의 성능을 비교해 봤다. 그림 2의 결과처럼 위성환경에서는 SACK의 성능이 더 좋은 것을 알 수 있다.

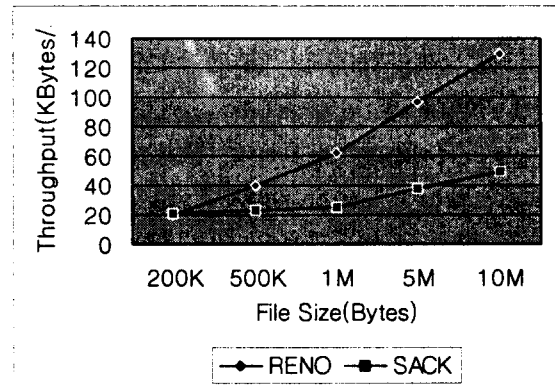


그림 2. RENO와 SACK의 성능평가

5.2 File Transfer의 성능 평가

T1/2의 위성링크속도, 64 Kbytes의 Maximum Window Size에서 200K, 500KB, 1 M, 5 M, 10 M의 File들을 실험했다. 그리고 에러없는 경우와 10^{-7} 의 에러율을 갖는 두 가지 경우에 대해 비교했다[그림 3].

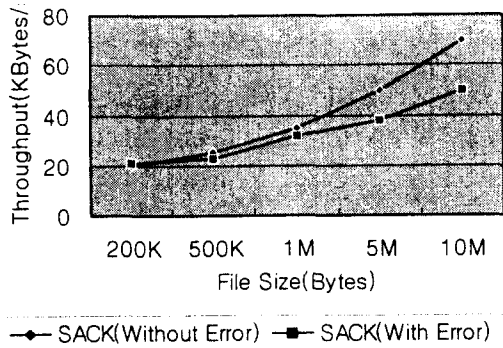


그림 3. File Size에 따른 성능평가

5.3 Window Scaling option 적용에 따른 성능평가

T1의 위성링크속도, Window Scaling Option의 적용유무에 따른 성능 평가이다[그림 4]. BER은 10^{-7} 이다. 위성과 같이 RTT가 큰 경우에는 Window Size를 크게 해줌으로써 성능을 향상시킬 수 있다. TCP 표준에 규정된 64KBytes일때와 T1의 최대 전송률을 사용할 수 있는 110KByte의 두 가지 경우에 대해 실험했다. 그 결과 110KBytes의 경우 좋은 성능을 나타내고 있다.

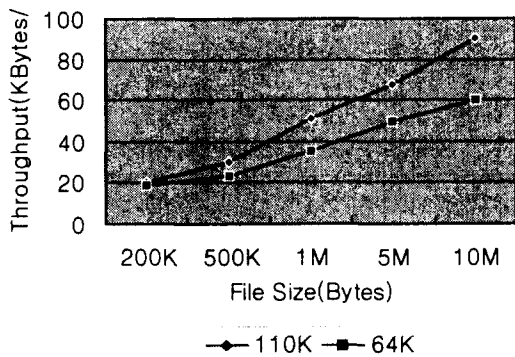


그림 4. Window Scale Option 적용유무에 따른 성능평가

5.4 BER에 따른 SACK과 ENA 알고리즘의 성능평가

T1의 위성링크속도, 그리고 T1의 전송률에

맞출 수 있는 110 KBytes의 Window Size를 선택했다. 그리고 5MB Sizes의 환경에서 $10^{-5}, 10^{-6}, 10^{-7}$ 의 에러율에 따른 성능을 평가했다[그림 5]. 각각의 Error rate에서 발생하는 세그먼트수는 약 300, 30, 3개 정도이다. 따라서 ENA 알고리즘은 에러발생률이 높은 경우에 성능을 더 많이 향상시킬 수가 있다.

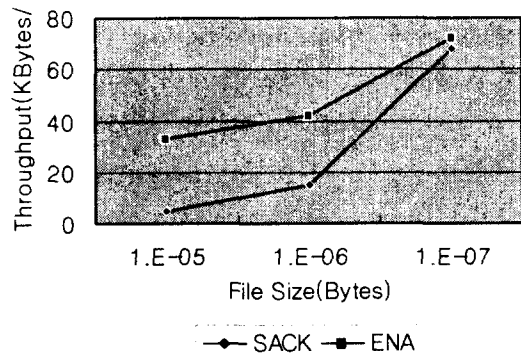


그림 5. BER에 따른 성능평가

5.5 ENA 알고리즘의 성능평가

다음은 이 논문에서 제안한 ENA 알고리즘에서 BER과 File Size간의 관계를 알아보기 위한 성능평가이다[그림6].

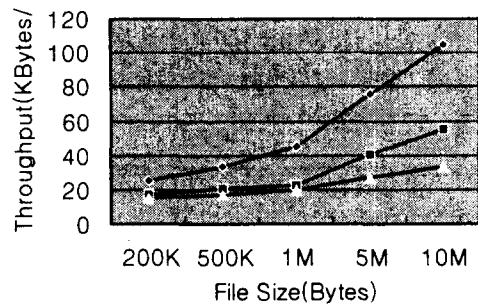


그림 6. BER과 File Size간의 성능평가

6. 결론

위성링크를 사용한 인터넷의 확장으로 위성망

에서의 TCP 성능향상을 위한 방안들이 중요한 문제로 떠오르기 시작했다. T.V.Lakshman, Aldar C.F.Chan는 Tahoe와 Reno TCP에 대해 수학적 분석을 통하여 성능을 분석하기도 했다 [10,11]. 우선 위성망의 특징 및 TCP의 성능을 저하시키는 요인에 대해서 알아보았다. 그리고 현재까지 연구된 TCP성능향상을 위해 제시된 기법들을 알아보았다. 그러나 에러에 의한 손실을 Congestion에 의한 손실로 보고 Congestion Control을 적용시킴으로서 성능을 저하시키는 방법에 대한 해결책으로 제시된 바가 없다. 이 논문은 이 부분에 대한 해결책으로 SACK Option Format에 수정했다. 즉 3개의 Block 정보를 포함하고 나머지 Bytes에는 에러를 알리는 bit와 해당 세그먼트의 Sequence Number를 나타내는 Bytes로 구성했다. ACK을 받은 송신측은 에러에 의한 손실인지 아닌지를 판단하여 에러에 의한 세그먼트만 재전송하는 ENA 알고리즘을 제안했다. 이 경우 Congestion Control알고리즘이 적용되지 않기 때문에 윈도우 사이즈에는 감소하지 않는다. 즉 네트워크로 전송되는 데이터 양을 줄이지 않는 것이다. 시뮬레이션 결과 ENA 알고리즘을 적용시킨 TCP가 성능이 기존의 SACK보다 더 향상되어 나타남을 알 수 있다. 그리고 에러율이 높은 경우일수록 성능이 더 향상되었다.

7. 참고 문헌

[1] Van Jacobson, "Congestion avoidance and Control", ACM SIGCOMM 88, 1998.
 [2] Stevens, W.Richard: "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," RFC 2001, January 1997.
 [3] Han Kruse, "TCP Performance in a Geostation-ary Satellite Environment".

[4] Craig Partridge and Tim Shepard: "TCP Performance over Satellite Links." IEEE Network, 11(5),pp44-49,September /October 1997.
 [5] Allman, Mark, Chris Hayes, Hans Kruse, and Shawn Ostermann:"TCP Performance Over Satellite Links." In Proceedings of the 5th International Conference on Telecommunication Systems, March 1997.
 [6] Aldar C.F.Chan, Danny.H.K. Tsang, Sanjay Gupta, "TCP Over Wireless Links", 97 VTC
 [7] Jacobson, V.R.Fraden and D.Borman: " TCP Extensions for high Performance", RFC 1323, February 1991.
 [8] Mathis, J. Mahdavi, Sally Floyd, and Allyn Romanow: "TCP Selective Acknowledgement Options", RFC 2018, October 1996.
 [9] Empirically-Derived Analytic Model of Wide Area TCP Connections, IEEE/ACM Transaction on Networking, pp 316-336, August 1994
 [10] T.V.Lakshman, Upamanyu Madhow, "The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss", IEEE/ACM TRANSACTIONS ON Networking. Vol.5, No.3, June 1997.
 [11] Aldar C.F.Chan, Danny.H.K.Tsang, Sanjay Gupta, "Performance Analysis of TCP in the Presence of Random Losses/Errors", Globecom 98.