

2-way 세그먼트 방식에서 Snoop 게이트웨이를 이용한

동적 TCP 흐름 제어 기법

이원정^o 안준철 임경식

경북대학교 컴퓨터학과

{wjlee^o, jcahn}@ccmc.knu.ac.kr kslim@knu.ac.kr

Dynamic TCP flow control scheme into 2-way segment by snoop gateway

Wonjung Lee^o Joonchul Ahn Kyungshik Lim

Dept. of Computer Science, Kyungpook National University

요 약

광대역 망에서 범용 TCP를 사용하는 경우 최대 전송 윈도우의 크기가 제한되어 가용한 망 자원이 낭비된다. 이를 해결하는 일반적인 방법은 윈도우 스케일 옵션을 사용하는 것인데 이 경우 송신측의 응용을 수정해야 하는 단점이 있다. 본 논문에서는 이를 해결하기 위하여 송신측을 수정하지 않고 전송 윈도우의 크기를 증가시키는 방법을 제안한다. 이 방법에서는 송수신측 사이에 있는 snoop 게이트웨이가 망의 상태와 수신측의 처리 능력을 동적으로 파악하여 최적의 윈도우 확대 계수를 결정하고, 이 계수에 상응하는 수의 응답 패킷을 송신측에 전송하여 전송 윈도우 크기가 확대된 효과를 얻는다. 특히, 이 방법은 서로 다른 특성을 갖는 망이 snoop 게이트웨이를 사용하여 2-way 세그먼트 방식으로 연동된 환경에서 부가적인 비용없이 단대단 시맨틱을 유지하면서 성능을 향상시킬 수 있다.

1. 서 론

광대역 망 환경에서 범용 TCP를 사용하는 경우 최대 전송 윈도우 크기의 제한으로 인해 망의 성능 저하가 발생하게 된다. 지연이 큰 망에서 범용 TCP를 이용하여 데이터를 전송할 경우 제한된 전송 윈도우만큼의 데이터를 전송한 후 긴 지연 시간이 지난 후에 수신측의 응답 패킷에 의해 추가적인 데이터 전송이 가능하게 된다. 송신측은 지연 시간동안 추가적인 데이터 전송을 하지 못하고 수신측의 응답 패킷을 기다리는 휴면 상태로 머무르게 되고 이용 가능한 망의 자원이 낭비된다.

범용 TCP의 최대 전송 윈도우 크기 제한을 극복하기 위한 연구가 제안되어 있다. 이러한 연구는 송신측의 윈도우 스케일 옵션을 이용하여 최대 전송 윈도우의 크기를 증가시키는 기법과 성능 향상 게이트웨이에서 전송 윈도우의 크기를 조절하는 기법이 있다. 본 논문에서는 송신측에서 윈도우 스케일 옵션을 사용하는 효과를 얻기 위해 최대 전송 윈도우의 크기를 확대시키는 새로운 기법을 제안한다. 그리고 다양한 망 환경에서 최적의 성능 향상을 얻을 수 있는 최대 전송 윈도우의 크기를 결정하는 기법을 제안한다.

본 논문은 다음과 같이 구성되어 있다. 제 2 장에서는 광대역 망의 성능을 향상시키기 위한 기존의 연구를 소개한다. 제 3 장에서는 전송 윈도우 확대 기법의 수행원리에 대해 소개하고, 제 4 장에서 최적 윈도우 확대 계수를 결정하는 방법을 시뮬레이션을 통해 확인한다. 마지막으로 제 5 장에서는 본 논문의 결론과 향후 과제에 대해 기술한다.

2. 관련 연구

광대역 망에서의 성능저하 요인인 범용 TCP의 전송 윈도우

크기 제한을 극복하기 위한 기존의 연구는 윈도우 스케일 옵션을 사용하는 것과 성능향상 게이트웨이를 두어 성능을 향상시키는 기법이 있다[1,2].

윈도우 스케일 옵션의 사용은 송신측에서 전송할 수 있는 최대 전송 윈도우의 크기를 확대하는 것이다. 연결 설정과정에서 SYN 패킷에 전송 윈도우의 확대 정도를 설정하면 수신측은 설정된 값을 이용하여 수신 윈도우의 크기를 확대하여 전송 윈도우만큼의 데이터를 수용할 수 있는 자원을 할당하게 된다. 전송 윈도우 크기의 증가로 같은 시간동안 전송할 수 있는 데이터의 양의 증가하게 되고 망의 성능 향상을 얻을 수 있다. 하지만 이 기법은 송신측을 수정하여야 하므로 많은 비용이 드는 문제점을 가지고 있다.

지연이 큰 망의 한 종류인 위성 망에서 범용 TCP의 성능을 향상시키는 기법으로 성능향상 게이트웨이를 사용하는 것이다. 이는 송신측과 수신측의 연결을 두 개로 분리하여 중간의 성능향상 게이트웨이에서 두 개의 연결을 관리한다. 이 기법은 전송 윈도우의 크기를 망의 상황에 따라 동적으로 변화시키면서 망의 환경에 따라 성능을 향상시킬 수 있는 크기로 변화시킨다. 하지만 TCP의 단대단 시맨틱을 지키지 못하게 되는 문제점을 가지고 있다.

본 논문에서는 이러한 문제점을 해결하기 위해서 송신측 혹은 수신측의 수정 없이 최대 전송 윈도우 크기의 확대를 담당하는 snoop 게이트웨이를 통해서 최대 전송 윈도우 크기를 확대시킬 수 있다. 그리고 이 게이트웨이에서 패킷 스누핑을 통해 데이터 패킷을 저장하고 다시 전송하므로 송신측과 수신측의 연결이 나누어지지 않고 TCP의 단대단 시맨틱을 유지할 수 있다[3].

3. 전송 윈도우 확대 기법

전송 윈도우 확대 기법은 망과 수신측의 상태에 따라 최적 윈도우 확대 계수를 결정하여 해당되는 수의 조작된 응답 패킷의

본 연구는 한국과학재단 목적기초연구(R01-2003-000-10562-0)와 정보통신연구진흥원의 대학 IT연구센터 지원으로 수행되었음.

전송을 통해 송신측의 전송 윈도우의 크기를 확대시킨다.

이 기법은 송신측의 전송 윈도우 확대 및 수신측의 수신 윈도우 확대 과정과 전송 과정에서 발생한 패킷 손실에 대한 복구 과정으로 구성된다. 그림 1은 간단한 전송 윈도우 확대 기법이 적용된 망 환경이다.

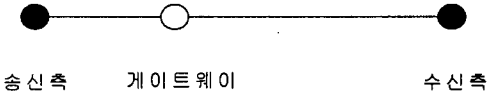


그림 2. 망 환경

3.1 전송 윈도우 크기 확대

송신측의 전송 윈도우의 크기는 범용 TCP를 사용하는 경우에 64KB로 제한된다. 이러한 단점을 극복하기 위해서 조작된 응답 패킷을 snoop 게이트웨이에서 생성하여 송신측으로 전송함으로써 송신측의 전송 윈도우 크기를 확대시킨다. 이 과정은 snoop 게이트웨이에서 조작된 응답 패킷의 생성과 전송 시기의 결정을 통해 수행된다.

조작된 응답 패킷을 생성하는 과정은 송신측에서 전송된 전송 윈도우만큼의 데이터 패킷의 양을 snoop 게이트웨이에서 판단하여 마지막 패킷에 대한 응답 패킷을 생성하여 송신측으로 전송한다. 송신측에서는 조작된 응답 패킷을 수신하고 최대 전송 윈도우 크기에 해당하는 데이터 패킷이 수신측에 정상적으로 전송되었다고 판단하고 전송 큐에서 응답 패킷을 기다리는 데이터 패킷들을 삭제한다. 삭제된 패킷의 양만큼 전송 윈도우의 크기가 증가되고 송신측에서는 추가적인 데이터 전송이 가능하게 된다.

조작된 응답 패킷의 전송 시기는 snoop 게이트웨이에 저장된 데이터의 양을 통해 판단한다. 저장된 양이 범용 TCP의 최대 전송 윈도우 크기의 상수 배가 되는 경우에 전송하게 된다. 전송 회수는 망과 수신측의 상태를 확인하여 결정된 윈도우 확대 계수에 해당하는 조작된 응답 패킷을 생성하여 송신측으로 전송하게 된다.

그림 1은 윈도우 확대 계수가 1 인 경우에 조작된 응답 패킷이 한번 전송되는 과정을 나타낸다. 송신측에서 최대 전송 윈도우 크기만큼의 데이터를 전송되면 snoop 게이트웨이에서 해당 데이터의 마지막 패킷을 수신하는 경우에 조작된 응답 패킷을 생성하여 송신측으로 전송하게 된다. 이를 수신한 송신측은 전송 윈도우의 크기가 다시 증가되고 추가적으로 최대 전송 윈도우 크기만큼의 데이터를 전송할 수 있게 된다. 이 과정을 통해 송신측의 전송 윈도우의 크기가 두 배로 확대된 효과를 확인할 수 있다.

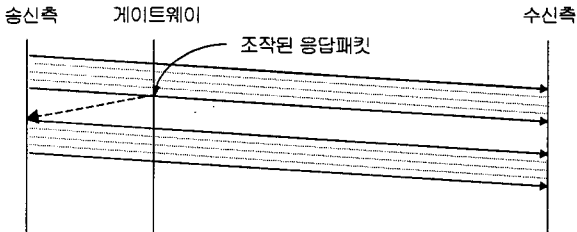


그림 3. 윈도우 확대 계수 1 인 경우 수행 과정

3.2 수신 윈도우 크기 확대

송신측의 전송 윈도우 크기가 확대됨에 따라 증가된 데이터의 양을 수용할 수 있는 수신측의 수신 윈도우의 크기가 또한 확대되어야 한다. 송신측에서 윈도우 스케일 옵션을 사용하는

경우 TCP 연결 설정과정에서 SYN 패킷을 이용하여 수신측의 수신 윈도우 크기 확대를 요청하게 된다. 요청에 의하여 수신측의 자원이 증가된 데이터의 수신을 위해 할당된다. 이 프로토콜을 이용하여 TCP의 연결 설정과정에서 전송되는 SYN 패킷을 snoop 게이트웨이에서 조작하여 송신측에 윈도우 스케일 옵션을 사용하는 것처럼 명시하여 전송하게 된다. 수신측은 SYN 패킷에 의해서 확대된 전송 윈도우의 크기를 수용할 수 있는 자원을 할당하게 된다.

하지만 수신측에서 윈도우 스케일 옵션을 지원하지 않는 경우, 전송 윈도우 확대 기법을 사용하지 않는 것이 효과적이다. 망에서 패킷 손실이 발생하게 되면 수신측에서 데이터 패킷을 상위 계층으로 전달하지 못하고 복구되기까지 손실된 패킷의 재 전송을 기다리게 된다. 또한 전송 윈도우 확대 기법을 이용하여 데이터 전송 양의 증가로 한 전송 윈도우 내에 패킷 손실이 발생할 확률이 높아지게 된다. 이로 인해 정상적으로 수신된 데이터를 상위 계층으로 전달하지 못하고 수신 큐에 저장하고 있어야 하므로 수신 윈도우 크기의 부족 현상이 발생하고 망 전체의 성능저하를 초래하게 된다.

3.3 데이터 패킷의 손실 복구

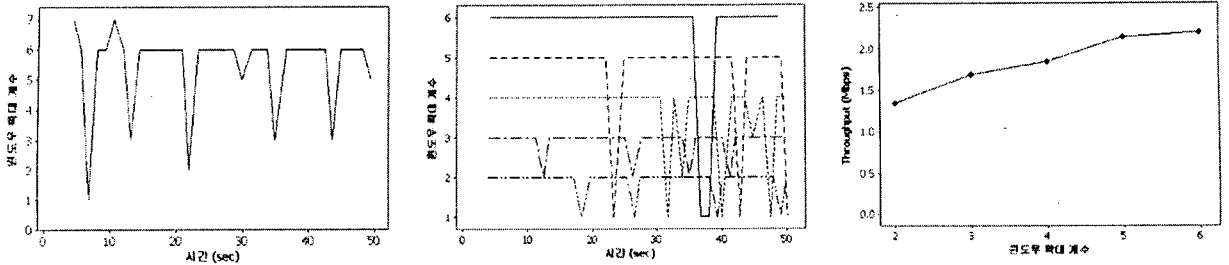
조작된 응답 패킷에 의해 전송 윈도우의 크기가 증가되면 한 번에 전송할 수 있는 데이터 패킷의 증가로 인해 한 전송 윈도우 내의 패킷 손실이 발생할 확률이 증가한다. 한 전송 윈도우 내에 한 개의 패킷이 손실되었을 경우에는 중복 응답을 사용하여 복구가 가능하다. 하지만 한 전송 윈도우 내에 두 개 이상의 패킷에서 손실이 발생하게 되면 중복 응답을 이용하여 복구할 경우 복구 과정에서 송신측의 RTO가 만료되는 문제점이 발생할 수 있다.

이를 해결하기 위하여 SACK 패킷을 사용하여 수신된 응답 패킷에 명시된 손실 정보를 이용하여 손실된 패킷의 재 전송을 수행하게 된다[4]. SACK 패킷은 한 전송 윈도우 내의 최대 세 개의 영역의 손실 패킷에 대한 복구가 가능하므로 snoop 게이트웨이에 저장된 데이터 패킷을 이용하여 손실된 패킷에 대한 일괄적인 재 전송을 수행할 수 있다. 이를 통해 복구 과정에서 발생하는 지연을 줄일 수 있고 송신측의 RTO가 만료되는 문제를 줄일 수 있다. 하지만 한 전송 윈도우 내에 비연속적으로 세 개 이상의 패킷 손실이 발생할 경우에는 SACK을 통해 모두 복구하지 못하고 망의 성능저하를 초래하게 된다.

4. 최적 윈도우 확대 계수

최적 윈도우 확대 계수는 특정한 망 환경에서 최적의 성능을 내는 윈도우 확대 계수를 의미한다. 광대역 망에서 최적 윈도우 확대 계수를 이용하여 특정 망 환경에서의 최적 윈도우 확대 계수를 결정하여 망에 적용시킴으로써 망의 자원을 효율적으로 사용하고 망의 성능을 향상시킬 수 있다.

패킷 손실이 발생하는 망 환경에서는 윈도우 확대 계수를 크게 설정해 놓더라도 값이 유동적으로 변하기 때문에 시간에 따라 최대 확대될 수 있는 전송 윈도우의 크기가 변하게 된다. 변화되는 전송 윈도우의 크기 가운데 가장 큰 윈도우 확대 계수를 구하게 된다. 그 값을 기준으로 최대로 확대될 수 있는 윈도우 확대 계수를 감소시켜 가면서 변화되는 윈도우 확대 계수의 변화와 각 경우에 망 전체의 성능을 측정하게 된다. 측정된 값을 판단하여 윈도우 확대 계수의 변화가 적고 망의 성능이 좋은 경우의 윈도우 확대 계수를 선택하게 된다. 이렇게 선택된 윈도우 확대 계수는 해당 망에서 최적의 성능을 낼 수 있는 최적 윈도우 확대 계수라고 할 수 있다. 이러한 판단 과정을 통해 다양한 망에서 적합한 최적 윈도우 확대 계수를 결정하게 된다.



a. 확대 계수가 10 인 경우

b. 확대 계수가 7 미만인 경우

c. 윈도우 확대 계수에 따른 성능

그림 3. 최적 윈도우 확대 계수 결정

본 논문에서는 특정 망 환경에서 최적 윈도우 확대 계수를 결정하는 과정을 시뮬레이션을 통해 확인하였다.

시뮬레이션 환경은 광대역 망의 한 종류인 DVBS(Digital Video Broadcast Satellite) 망 환경에서 성능을 측정하고 분석하였다. 망 시뮬레이터는 ns-2(버전 2.28)를 사용하였고, 레드햇 9.0 버전을 운영체제로 하는 펜티엄 4의 3.0 Ghz 시스템을 사용하였다[5].

4.1. 시뮬레이션 모델

그림 4는 DVBS 망의 특징을 가진 간단한 시뮬레이션 환경을 나타낸 것이다. 위성 구간의 RTT 값은 500ms로 설정하고, 비트 에러를 10^{-7} 으로 설정하였다. 시뮬레이션 시간은 50초로 하여 측정하였다. ns-2의 수신측이 기본적으로 무한한 수신 윈도우를 가지고 있다고 가정한다.

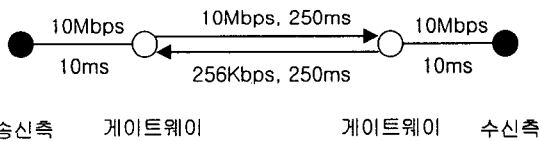


그림 4. 시뮬레이션 환경

4.2. 최적의 전송 윈도우 크기

특정 망 환경에서 시간에 따른 윈도우 확대 계수의 변화를 이용하여 최적의 성능을 내는 최적 윈도우 확대 계수를 결정하게 된다.

그림 3. a는 비트 에러가 10^{-7} 인 망 환경에서 최대 윈도우 확대 계수를 10으로 설정을 하여 최대 10배까지 전송 윈도우의 크기가 확대될 수 있는 조건 하에서의 윈도우 확대 계수의 변화이다. 비트 에러로 인해 윈도우 확대 계수의 변화가 생기게 된다. 이로 인해 윈도우 확대 계수를 크게 설정해 놓더라도 확대될 수 있는 최대 전송 윈도우의 크기가 7로 제한됨을 확인할 수 있다. 시뮬레이션 결과를 통해 가장 큰 윈도우 확대 계수를 얻을 수 있다.

가장 큰 윈도우 확대 계수를 구한 다음 7 미만의 윈도우 확대 계수를 최대 값으로 가지도록 설정하여 동일한 망 환경에서 각 경우의 윈도우의 변화를 측정하게 된다. 시뮬레이션 결과인 그림 3. b는 서로 다른 최대 윈도우 확대 계수를 적용한 환경에서 윈도우 확대 계수 변화를 나타낸다. 그리고 그림 3. c는 각각의 윈도우 확대 계수가 적용된 환경에서 측정된 망의 성능을 나타낸다. 그림 3. b, c를 이용하여 변화량이 적고, 성능이 높은 윈도우 확대 계수를 구하면 그림 3. b에서 윈도우 확대 계수의 변화가 6인 경우에 가장 적음을 확인할 수 있다. 그림 3. c에서 측정된 망의 성능 또한 6인 경우에 더 향상된 것을

확인할 수 있다. 위의 결과를 이용하여 최적 윈도우 확대 계수를 6으로 판단하게 된다. 따라서 시뮬레이션 환경에서 6을 최적 윈도우 확대 계수로 설정하게 되면 망에서 최적의 성능 향상을 얻을 수 있다.

시뮬레이션을 통해 특정 망 환경에서 최적의 성능을 내는 최적 윈도우 확대 계수를 판단할 수 있었다. 다양한 광대역 망에서 망 환경에 따라 앞의 제안된 방법을 이용하여 최적의 성능을 낼 수 있는 최적 윈도우 확대 계수를 결정하게 된다.

5. 결론 및 향후 과제

광대역 망에서 범용 TCP의 전송 윈도우 제한으로 인한 성능저하 현상을 송신측 혹은 수신측에 수정을 가하지 않고 snoop 게이트웨이를 이용하여 망 전체의 성능향상을 얻을 수 있다. snoop 게이트웨이에서 조작된 응답 패킷을 만들어 전송함으로써 송신측 전송 윈도우의 크기가 확대된 효과를 얻을 수 있다. 그리고 시뮬레이션을 통해 특정한 망 환경에서 최적의 성능을 낼 수 있는 윈도우 확대 계수를 확인하는 방법을 제안한다.

향후 과제로서 시스템에서 SACK, 타임 스탬프 옵션 및 윈도우 스케일 옵션의 지원 여부와 망에서 발생하는 에러 혹은 패킷 손실 등의 요소를 가지는 다양한 망의 환경에서 제 4 장에 소개된 방법을 이용하여 최적의 성능을 낼 수 있는 윈도우 확대 계수를 결정하는 방안에 관한 연구가 필요하다.

참고 문헌

[1] V. Jacobson, R. Braden, and D. Borman, "TCP Extensions for High Performance," RFC 1323, May 1992.
 [2] SkyX Technology, White Paper, <http://www.mentat.com/skyy/whitepaper.html>, 2003.
 [3] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks," ACM Mobile Computing and Networking Conference, Vol. 1, Issue 4, November 1995.
 [4] M. Mathis, J. Mahdavi, S. Floyd and A. Romanow, "TCP Selective Acknowledgment Options," RFC 2018, October 1996.
 [5] K. Fall and K. Varadhan, ns Notes and Documentaion, LBNL, <http://www.isi.edu/nsnam/ns/>, April 2005.