비정상 트래픽 전달을 교정하는 트래픽 컨디셔너의 개발

문귀남° 박창윤
중 앙 대 학 교 컴 퓨 터 공 학 과
{knmoon, cypark}@orchid.cse.cau.ac.kr

Development of A Traffic Conditioner Correcting Abnormal Traffic Behavior

Kwi-Nam Moon° ChangYun Park
Dept. of Computer Science & Engineering, ChungAng University

요 약

인터넷의 급속한 증가는 네트워크상의 트래픽 형성에 중요한 변화를 가져오게 되었다. 통신 성능 향상을 지원하는 과정에서 여러 가지 통신상의 이상 현상들이 발생하게 되었으며, 그로 인한 통신상의 성능 문제에 부딪히게 되었다. 본 논문에서는 이러한 네트워크상의 비정상적인 트래픽의 예 중에서 비순서적 도착(Out-of-Order)에 대해서 실험을 하였으며, 트래픽 컨디셔너라는 개념과 이 개념을 도입한 독립형 구현 모델을 이용해서 네트워크상의 통신 성능을 개선하는 방법에 대해서 설명한다.

1. 서론

최근 몇 년 사이에 인터넷이 우리 일상 생활의 한 부분으로 자리잡게 되면서, 이제는 누구나 쉽게 인터넷을 사용해서 자신이 원하는 정보를 이용할 수 있게 되었다. 이러한 인터넷의 급속한 증가는 네트워크 상의 트래픽 형성에 중요한 변화를 가져오게 되었다. 인터넷이 보급된 이후로 사람들은 끊임없이 새로운 서비스의 필요성을 느끼게 되었으며, 현재에는 단순한 텍스트 위주에서 벗어나 오디오, 비디오와 같은 멀티미디어 서비스를 이용할 수 있게 되었다. 이러한 멀티미디어 서비스는 기본적으로 대용량의 트래픽을 필요로 하게 된다. 이러한 대용량의 트래픽을 필요로 하는 통신 서비스들이 증가하게 되면서 네트워크 망에서의 트래픽은 급속하게 증가를 하게 되었다.

네트워크상의 트래픽이 이처럼 기하급수적으로 증가하게 되면서 우리는 이에 따른 여러 가지 통신상의 이상 현상들을 겪게 되었다[4]. 예를 들면, 네트워크의 혼잡으로 인한 통신 단절 현상이나 패킷의 손실, 서버의 과부하 등 여러 가지 현상이 발생하게 된다.

인터넷상의 통신 성능 및 사용자가 느끼는 서비스 성능은 네트워크의 기반이 되는 통신망과 응용 서비스를 제공하는 서버, 그리고 사용자인 클라이언트의 상호작용으로 결정된다. 서버와 클라이언트의 성능이 아무리 뛰어나더라도 그 기반이 되는 네트워크 망의 성능이 떨어지게 된다면 전체적인 통신 성능은 좋을 수가 없게 된다. 반대로, 통신망의 성능이 뛰어나도 서버나 클라이언트의 성능이 떨어지는 경우에도 마찬가지 결과를 가져온다. 따라서 어느 한쪽의 성능이 뛰어나다고 해서 전체적인 통신의 성능이 개선되는 것은 아니다.

본 논문에서는 트래픽 컨디셔너(Traffic Conditioner : TC)라는 개념을 도입하여 이러한 네트워크의 통신 성능을 개선하기 위한 방법을 제시하고자 한다. 2장에서는 관련연구에 대해서 살펴보고, 3장에서는 통신 이상의 예 및 실험 결과를 알아보고, 4장에서는 본 논문에서 제시한 트래픽 컨디셔너에 대해서 살펴볼 것이다.

2. 관련 연구

네트워크의 통신 성능을 개선하고자 하는 노력은 여러 방면에서 이루어지고 있다. 이러한 연구 중 네트워크 자원의 효율적인 활용과 통신 성능의 개선을 목표로 하는 트래픽 엔지니어링(Traffic Engineering : TE) 기술[1][5]이 활발히 연구되고 있다. 트래픽 엔지니어링이란 트래픽을 네트워크상의 링크, 스위치, 라우터들에게 골고루 분산시키도록 트래픽의 경로를 제어함으로써, 네트워크 자원의 사용을 최적화하고 성능을 높이는 기술을 의미한다. 인터넷의 급속한 성장으로 몇몇 ISP 네트워크는 급증하는 IP 트래픽에 대처하기 위해 자신들의 망 내에 라우터와 전송 링크 등을 증설하고 있다. 이로 인해 ISP 백본망은 노드와 링크 측면에서 확장되고 있으며, 따라서 망 내에 수많은 대체 경로가 존재하게 되었다[7]. 특히, 네트워크 내의 두 지점간에 동일한 비용의 복수개의 경로(Equal Cost Multi-Path : ECMP)가 존재하는 경우 트래픽 엔지니어링은 큰 의미를 갖는다[1].

트래픽 쉐이퍼(Traffic Shaper)란 애플리케이션이 생성한 트래픽이 네트워크 망으로 전달되기 바로 전에 자신의 트래픽에 대해서 일정한 특성을 형성(shaping)시켜 주는 기능을 하는 것을 말한다[6]. 예를 들면, 버스트한 트래픽의 흐름을 적정 속도를 만족하도록 트래픽의 흐름을 조절하는 것이 될 수 있다. 리눅스에서는 모듈 형태로 트래픽 쉐이퍼 기능을 하는 드라이버를 제공하고 있다. 현재 여러 네트워크 장비 벤더들에서는 서로 앞다투어 트래픽 쉐이퍼 기능 외에도 여러 가지 기술을 접목한 제품을 출시하고 있다.

3. 트래픽의 이상 현상

3.1. 비정상 트래픽 전달 동작의 예 - 비순서적 도착

네트워크 망 사이에서는 트래픽의 혼잡(Congestion)을 미리 예방하기 위한 조처로 RED(Random Early Detection)에 의해서 패킷이 망 사이에서 버려지는 경우가 발생한다. 이때 패킷을 보낸 송신측에서는 보낸 패킷에 대한 ACK가 되돌아오지 않기 때문에 타임아웃에 의한 패킷의 재전송이 이루어지게 된다. 이 경우 수신측에서는 받은 패킷의 일부 중에 비순서적 패킷이 발생할 수 있게 된다. 이러한 비순서적 패킷이 많이 발생하게 되면 그만큼 수신측에서는 순서를 맞추는 작업을 해주어야만 하는데, 이러한 불필요한 작업을 처리하는데 시간이 소모되기 때문에 성능 저하를 가져오게 된다.

또 다른 측면으로는 송신측과 수신측 사이의 네트워크 망 사이에는 망의 성능을 높이기 위해 통신망 사업자들이 주요 노드 사이에 복수 개의 링크를 설치, 운영하고 있다[7]. 따라서 동일 비용의 여러 개의 경로(Equal Cost Multi-Path : ECMP)가 망 사이에 존재하게 된다[1][4]. 이 경우에 연속되는 패킷이 전송될 때 동일한 경로를 거치는 것이 아니라 다른 경로를 통해서도 전달될 수 있기 때문에 수신측에서 패킷이 비순서적으로 도착하게 되는 문제가 발생하게 된다. 따라서 단말 호스트에서는 이러한 패킷에 대한 처리를 모두 해주어야 하기 때문에 통신 성능에 문제가 발생하게 된다.

3.2 비순서적 도착에 대한 실험

네트워크상의 비정상적인 트래픽의 실제적인 예로, 본 논문에서는 ECMP로 인한 패킷의 비순서적 도착에 대한 실험을 테스트하였다. 본 실험에 사용된 응용 프로토콜로는 FTP와 HTTP를 이용하였으며, 테스트 환경은 리눅스 머신을 이용하였다. 실험 방법은 리눅스상의 네트워킹 관련 커널 소스(*tcp_input.c*)를 수정해서, 처음 순서 바뀜이 발생한 시점의 시간과 커널 상에서 다시 순서를 맞추는 시간을 측정하여 두 시간 사이의 차이를 분석하였다. 실험 결과는 실험을 시행한 시간대에 따라서, 그리고 그 당시 네트워크의 상태에 따라서 결과는 달라질 수 있지만, 주로 트래픽의 사용량이 적은 시간대를 이용해서 테스트를 하였다.

1) Ping을 이용한 지연시간 분석

	Min	Avg	Max
1	2.5	4.3	12.9
2	2.6	3.8	7.3
3	2.6	3.9	6.8
4	2.7	4.1	8.3
5	2.9	5.8	14.3
평균	2.66	4.38	9.92

표 1 RTT 측정값 (단위, ms)

먼저 간단한 Ping 테스트를 통해 송신측과 수신측 사이의 RTT(Round Trip Time)를 측정하였다. 모두 5번의 실험으로 각 실험마다 100번의 패킷 교환 시간의 RTT값의 최소/평균/최대 시간을 측정한 결과값은 표와 같다. 결과표의 측정치에서 볼 수 있듯이 평균적인 RTT의 최소값은 2.66ms정도이다. 따라서 송신측에서 수신측으로 보내는 패킷이 도달하는데는 대략 1.33ms정도의 시간이 소요될 수 있다는 것을 알 수 있다. 실험은 중앙대와 보라넷(ftp.bora.net) 사이에서 이루어졌다.

2) FTP

FTP를 이용해서 송신측에서 수신측으로 보내는 패킷 중 비순서적 도착이 발생한 후에 순서를 이루는 시간과의 차이값을 측정한 결과는 그림과 같다. 그림 1에서 볼 수 있듯이 보통의 경우에는 1ms이상이 소요되지만, 경우에 따라서는 1ms에 훨씬 못미치는 경우(예, 0.029ms)도 발생하게 된다. 이것은 RTT 시간보다 훨씬 작은 것으로써, 재전송에 의한 패킷의 비순서적 도착보다는 네트워크 상의 ECMP로 인한 순수한 비순서적 도착이라는 것을 알 수 있다.

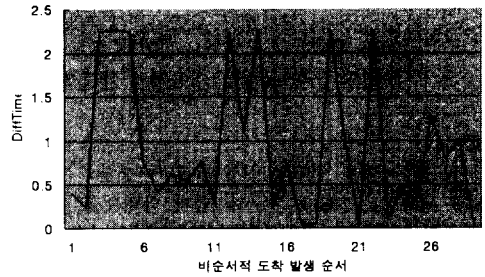

그림 1 순서 복구 시간 (단위, ms)

3) HTTP

HTTP의 경우에는 FTP 실험보다 비순서적 세그먼트가 빈번히 발생하였다. 이것은 HTTP의 특성상 단발성 연결이 복수개 발생하는 결과로 볼 수 있다[2]. 국내, 국외 웹사이트를 이용해서 실험을 한 결과, 일반적인 경우 순서 복구에 소요되는 시간은 1s 이상인 경우가 대부분이고, 짧을 경우에는 FTP 테스트 결과와 마찬가지로 1ms에 훨씬 못미치는 경우(예, 0.053ms)도 발생하였다.

위의 두 실험 결과 재전송에 의한 비순서적 도착도 발생을 하게 되지만, 순수한 의미의 비순서적 세그먼트도 생각보다는 많이 발생한다는 사실을 알 수 있었다.

4. 트래픽 컨디셔너(Traffic Conditioner)

4.1 트래픽 컨디셔너 개념

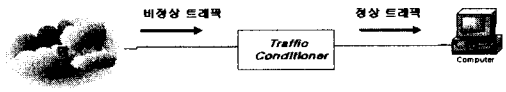

그림 2 트래픽 컨디셔너의 개념

트래픽 컨디셔너란 네트워크를 통해 전달된 트래픽을 종단 호스트가 처리하기 전에 호스트 쪽에서 처리가 용이하도록 조정을 하여, 전체적인 통신 성능의 향상을 얻고자 하는 통신망 요소라고 정의한다. 예를 들면, 비정상적으로 들어오는 트래픽

의 경우에는 정상적인 트래픽으로 바꾸어주거나, 트래픽의 차별화 혹은 부가적인 기능 등을 미리 처리하여 트래픽의 성능이 개선되는 효과를 얻고자 하는 것이다. 트래픽 컨디셔너 개념을 적용하기 위해서는 여러 가지 필요한 조건들이 있다. 그 중 중요한 조건들을 살펴보면 다음과 같다.

1) 투명성(Transparency)

기존 서버 시스템과 네트워크 상의 다른 시스템 사이에 투명성 유지가 필요하게 된다. 다른 시스템 입장에서는 중간에 트래픽 컨디셔너가 있는지 없는지에 관계없이 작동을 하여야 한다.

2) 성능

트래픽 컨디셔너를 적용함으로써 적용하기 전보다 처리량의 증가를 성능 효과로 얻을 수 있어야 한다. 이를 위해서는 TCP 레벨의 빠른 포워딩 방법이 필요하다[3]. 또한 트래픽 컨디셔너로 인한 지연시간 증가의 최소화도 반드시 필요한 조건이다.

4.2. 구현 모델

트래픽 컨디셔너 개념을 실제적으로 적용할 수 있는 구현 모델은 크게 독립형 모델과 내장형 모델의 두 가지로 나누어 볼 수 있다. 먼저 각각의 특징에 대해서 살펴보면 다음과 같다.

1) 독립형 모델

독립형 모델의 경우는 기존의 통신 컴포넌트(즉, 호스트 및 라우터)와는 별도로 독립적으로 작동하는 모델이다. 이 경우 서버나 기존 네트워크 구성은 그대로 유지하면서 독립적으로 트래픽 컨디셔너의 기능을 수행하는 모델이다. 따라서 독립형 모델의 경우에는 기존 네트워크 구성과의 투명성을 유지해야만 한다는 조건이 있다.

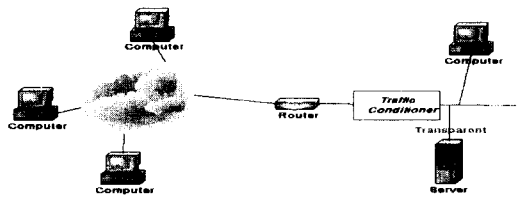

그림 3 트래픽 컨디셔너 구현 모델 - 독립형

2) 내장형 모델

내장형 모델의 경우는 라우터나 Web 서버에 트래픽 컨디셔너의 기능을 추가로 내장하는 모델이다. 따라서 기존의 라우터나 Web 서버 시스템의 확장 및 변형이 필요하게 된다. 라우터의 경우에는 미리 트래픽에 대한 처리를 수행하기 때문에 단말 호스트에서의 성능 개선 효과를 볼 수 있다. Web 서버의 경우에는 트래픽 컨디셔너를 이용해서 Web 서버의 부하를 줄임으로써 전체적인 성능을 효율적으로 개선할 수 있게 된다.

본 논문에서는 위의 두 가지 구현 모델 중 첫째인 독립형 모델을 기본 모델로 정했다. 구현 방법은 호스트에게 비순서적으로 전달되는 패킷들을 따로 저장하였다가, 순서를 맞추어 빠르게 포워딩 하여주는 기능[3]을 가지는 트래픽 컨디셔너(TC)를 개발하여 테스트를 시행하였다. TC의 개념적인 내부 구조는 그림5와 같다. 포워딩 방법은 투명한 브릿지(Transparent Bridge) 모델을 참조하여, TCP 레벨에서의 빠른 포워딩 방법을 이용하

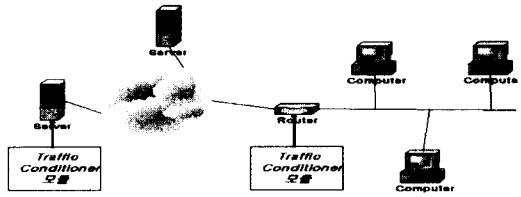

그림 4 트래픽 컨디셔너 구현 모델 - 내장형(라우터)

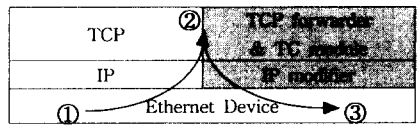

그림 5 TC의 개괄적인 내부구조

① IP module을 수정하여 모든 패킷을 TCP로 전송

② 전송된 패킷의 처리(Out-of-Order인 경우 순서를 맞춤)

③ fast forwarding

였다. 호스트 쪽에서는 정상적인 패킷 처리가 가능하므로 부하가 감소하게 된다. 결국, 통신 서비스 전체의 성능 개선 효과를 얻을 수 있다.

본 연구에서는 트래픽의 비순서적 전달을 교정해주는 트래픽 컨디셔너 모델에 대해서 연구했고, 향후에는 기존의 라우터에 트래픽 컨디셔너를 내장한 모델에 대한 연구를 진행할 예정이다.

5. 기대 효과

본 논문에서 제시한 트래픽 컨디셔너를 이용해서 얻을 수 있는 기대 효과는 통신 성능의 전체적인 개선이다. 기존의 통신 구성 요소와 독립적으로 트래픽 컨디셔너가 위치함으로써, 비정상적인 트래픽을 효율적으로 처리해 주기 때문에 서버와 호스트에서의 성능 개선 효과를 얻을 수 있다. 특히 단말 호스트가 내장형 시스템(Embedded System)처럼 프로세싱 능력이 떨어지는 낮은 시스템인 경우에는 성능 효과가 더욱 클 것으로 예상된다.

6. 참고 문헌

[1] P. Trimintzios, G. Pavlou, I. Andrikopoulos, "Providing Traffic Engineering Capabilities in IP Networks Using Logical Paths," in *Proc. 8th IFIP Workshop*, July 2000

[2] H. Balakrishnan, V. Padmanabhan, S. Seshan, M. Stemm, R. Katz, "TCP Behavior of a Busy Internet Server : Analysis and Improvements," in *Technical Report UCB/ CSD-97-966*, Univ. California, Berkeley, CA, August 1997

[3] O. Spatscheck, J. Hansen, J. Hartman, L. Peterson, "Optimizing TCP Forwarder Performance," in *IEEE/ACM Transaction on Networking*, pp. 146-157, April 2000

[4] J. Bennett, C. Patridge and N. Shectman, "Packet Reordering is Not Pathological Network Behavior," in *IEEE/ACM Transactions on Networking*, pp.789-798, December 1999

[5] X. Xiao, A. Hannan, B. Bailey, S. Carter, L. M. Ni, "Traffic Engineering with MPLS in the Internet," in *IEEE Network magazine*, pp. 28-33, March 2000

[6] Dah Ming Chiu, Miriam Kadansky, "Experiences in Programming a Traffic Shaper," in *SMLI TR-99-77*, 1999

[7] 손장우, "MPLS의 등장 배경과 장점", http://www. netmanias.com