

윈도우 사이즈 조절을 통한 내부 네트워크의 성능향상

전철완, 권경희
단국대학교 전자계산학과
pureloveu@msn.com, khkwon@dankook.ac.kr

Improving a Internal Network Performance By Controlling Window Size

Chul-Wan Jeon, Kyung-Hee Kwon
Dept. of Computer Science, Dan-Kook University

요 약

본 연구는 어떤 조직 내에서의 네트워크를 내부 네트워크와 외부 네트워크로 구분하고, 이들에 각각의 서버를 할당하여 분산된 서비스를 제공하는 환경을 대상으로 한다. 각 서버의 특성에 맞게 TCP Parameter 중의 하나인 Window Size 를 설정하여 내부 네트워크의 성능 향상을 시도하였다. 외부 네트워크에서는 Window Size 가 큰 영향을 미치지 못하나 내부 네트워크에서는 Window Size 에 따라서 처리율(Throughput)이 크게 달라진다. 본 연구에서는 Window Size 의 조절을 통해서 내부 네트워크의 처리율(Throughput) 변화를 NS-2 를 사용하여 시뮬레이션해 보고 그 결과를 비교하여 내부 네트워크의 성능 향상을 확인하고자 한다.

1. 서론

지금까지의 많은 실험 및 연구에서 증명되었듯이 TCP Parameter의 변화는 네트워크의 전체 성능에 많은 영향을 미치게 된다. TCP Parameter들 중에서 Window Size도 네트워크의 성능에 영향을 미치게 된다. 네트워크의 전송량을 늘리기 위해서 Window Size를 크게 하면 에러(Error)나 패킷(Packet)을 잃어 버렸을 때 재전송을 하게 되는데, 이때 큰 Window Size 만큼을 다시 전송해야 한다는 문제점을 가지고 있고, 그와 반대로 Window Size를 작게 한다면 작게 한만큼 오버헤드가 커진다는 단점을 가지고 있다. 따라서 가장 적당한 Window Size의 설정이 전체 네트워크의 성능 향상에 영향을 준다고 할 수 있다.

본 연구에서는 일반적으로 TCP 프로토콜을 통해서 전송하는 네트워크를 대상으로 하였다. 클라이언트(Client)의 접속 요청 위치에 따라 외부에서 접속해 오는 서버와 내부에서 접속해 오는 서버를 각각 따로 구성하였다. 외부 네트워크는 일상적으로 말하는 인터넷을 의미하고, 내부

네트워크는 어떤 조직내에서 LAN(Local Area Network) 또는 LAN과 LAN의 연결 구조를 가지는 네트워크를 의미한다. 내부의 네트워크를 통해서 접속해 오는 데이터만을 처리하는 서버는 내부에서 환경을 임의적으로 변경하여도 변경된 값을 적용할 수 있다는 것과 그 변경된 값은 바로 서버 및 네트워크 전체에 영향을 미칠 수 있다는 장점을 가지고 있다. 이에 반해 인터넷과 같은 외부의 접속 요청에 의해 처리되는 서버는 환경을 임의적으로 변경한다 하더라도 서로 상이한 네트워크들의 집합체인 인터넷의 특성상 외적인 요소에 많은 변화를 가져온다. 그렇기 때문에 실험을 통해서 성능 향상의 어려움을 가지고 있고 본 논문에서의 의도와는 다르기 때문에 외부로부터의 접속이 이루어지는 서버의 환경 설정 및 성능분석은 배제하도록 하였다.

서버는 내부 네트워크용 서버와 외부 네트워크용 서버로 분리해서 구성하고 LAN을 통한 접속 요청과 외부에서 이루어지는 접속 요청을 Router를 통해서 내부 서버와 외부 서버로 트래픽(Traffic)을 분산시켜 연구를 진행하였다. 내부의 네트워크 즉,

LAN은 TCP 프로토콜 위에서 동작하는 응용 계층을 대상으로 환경을 설정하고 트래픽을 발생시켜 실험을 진행하였다.

현재와 같은 복잡한 네트워크 환경에서는 네트워크의 구성은 물론, 적절한 분석이 어렵기 때문에 본 연구에서는 NS-2(Network Simulator-2)를 통해서 네트워크를 구성하고 Window Size를 조절 할 경우 처리율(Throughput)의 변화를 통해 네트워크에 어떠한 영향을 미치는지를 조사 및 분석하였다.

지금까지 TCP Parameter의 조절을 통해서 네트워크의 성능 향상 방법들은 많이 연구되어져 왔다. 그러나 Window Size 조절을 내부 네트워크용 서버와 외부 네트워크용 서버로 분리되지 않은 통합적인 서버에서 적용시켜 성능을 평가 하기는 어렵다. Window Size를 조절하는 것은 네트워크를 통해서 접속 되어진 수신측 호스트 TCP Parameter를 변경하여 처리율을 상승시키는 것이다. 그러나 Window Size는 수신측 호스트의 운영체제마다 조금씩 다르다는 점과 인터넷과 같은 거대한 네트워크로 연결되어 데이터가 전송되어질 경우 네트워크의 상태에 따라서 수시로 재설정 이 이루어지기 때문에 특정한 곳에서의 성능향상을 이루기는 어렵다고 보는 것이 사실이다.

따라서 본 연구에서는 단순한 Window Size 의 조절 뿐만 아니라 외부 접속용 서버와 내부 접속용 서버를 구분하여 서비스 할 수 있도록 하여 트래픽의 분산과 함께 내부 접속용 서버의 Window Size 를 조절하여 내부 네트워크의 성능 향상을 확인하였다.

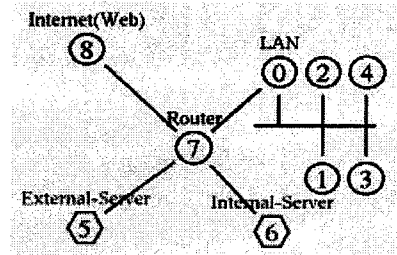
2. 네트워크의 구성

현재와 같은 외부의 접속 요청과 내부의 접속 요청을 동시에 처리하는 통합된 서버는 사용자의 증가로 인한 많은 접속 요구에 따라 서비스가 그에 부응하지 못하고 있다는 것이다. 예로서 느린 접속 속도 및 시간초과로 인한 재접속 등을 들 수 있으며 이는 서버의 과부하를 유발시킨다. 이러한 서버의 과부하는 패킷 손실 및 서비스 지연, 서버 다운 등의 문제로 이어지면서 서버 분산의 필요성이 대두되었다.

하지만, 대용량의 서버의 분산은 관리의 어려움과 비용 증가의 문제점을 야기 시킨다. 관리의 복잡함과 비용의 부담을 줄일 수 있는 방법으로 같은 서비스를 하는 서버를 분산시켜 성능을 향상시키는 방법을 사용한다. 본 연구에서는 많은 서버로 분산시키기 보다는 같은 서비스를 하는 내부 접속용 서버와 외부 접속용 서버로 분산 시켜 실험을 진행하였다. 내부 접속용 서버의 Window Size 조절을 통해서 성능 향상을 확인하기 위해 [그림 1]와 같이 네트워크를 구성하여 실험을 진행하였다.

LAN 의 세부 환경은 [표 1]과 같은 형태로 설정하였다. [표 1]의 Mac 802.3 은 IEEE 802.3 과 유사한 형태로 NS-2 에서 사용하는 표준이다.

내부에서의 모든 노드는 트래픽(Traffic)을 일정한 간격으로 시뮬레이션이 끝나는 시간까지 무한정으로 발생 시킨다.



[그림 1] 서버 분산후 네트워크

Link	Configuration
Bandwidth	10Mbps
Delay	0.3ms
Link Layer	LL Type, DropTail
Mac Layer	Mac 802.3
Physical Layer	Channel
Packet Size	1000Byte

[표 1] LAN 의 세부환경

[그림 2]와 관련된 자세한 설정사항은 [표 2]에 정리하였다.

Parameter	LAN	n8-n7	n7-n5	n0-n7	n7-n6
Bandwidth	10Mb	10Mb	10Mb	10Mb	10Mb
Delay	0.3ms	1ms	1ms	1ms	1ms
Queue	Drop Tail	Drop Tail	Drop Tail	Drop Tail	Drop Tail
Packetsize	1000 Byte				

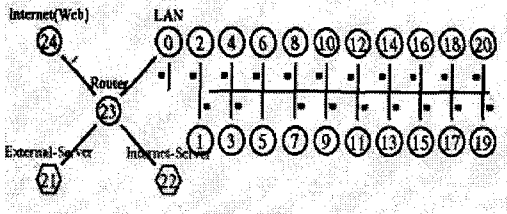
[표 2] Link Parameter

3. 시뮬레이션 및 성능분석

3.1 시뮬레이션

서론에서 언급하였듯이 네트워크를 분산시켜서 내부 접속용 서버의 Window Size 를 조절하여 성능 향상을 확인하기 위해서 NS-2[5,8]을 사용하여 [그림 1]와 같이 LAN 의 노드 개수가 5 개부터 시작해서 [표 1], [표 2]와 같은 환경으로 노드의 개수를 [그림 2]와 같이 증가시킬 경우 처리율(Throughput)의 변화를 확인함으로써 가장 최적의 Window Size 확인하였다.

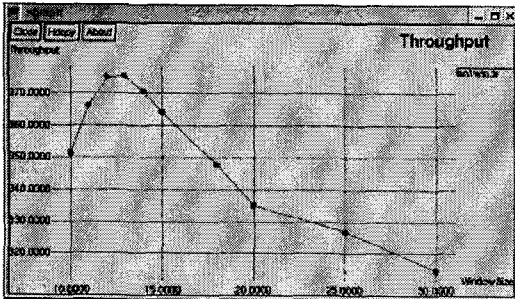
본 실험에서는 처리율의 단위가 Mbit/s 이며, Window Size 는 패킷 수를 의미 하는데, NS-2 는 20 개로 설정이 되어 있다.[7] NS-2 의 디폴트 값을 기준으로 실험을 진행하였다.



[그림 2] LAN Node 증가(노드수 20)

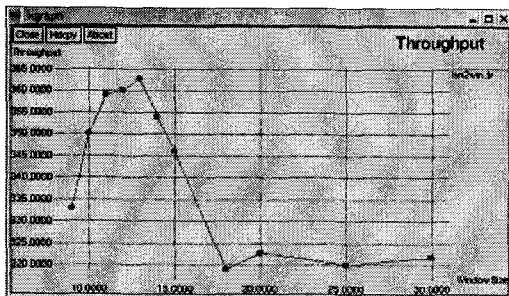
3.2 네트워크 성능분석

[그림 1]과 같은 LAN에서 Node2의 호스트만이 트래픽을 발생시킬 경우 Window Size 변화에 따라 성능 변화를 아래 [그림 3]의 그래프를 통해서 확인할 수 있다.



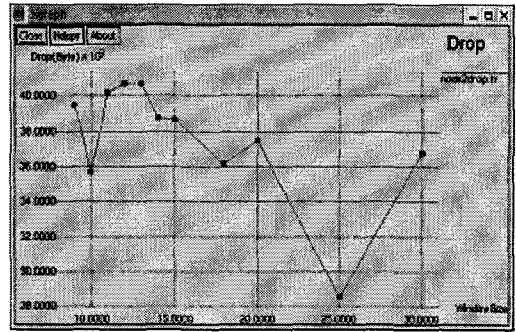
[그림 3] Throughput 변화(트래픽발생 노드수 1)

[그림 3]은 Window Size 를 10 개부터 시작해서 30 개까지 진행하는 동안 처리율의 변화를 나타내 준다. Window Size 가 12, 13 개 일 경우에 처리율이 가장 좋다는 것을 확인할 수 있다. LAN 에서 데이터를 발생하는 노드의 수가 1 개일 경우에는 드롭(Drop) 되는 데이터는 발생하지 않았다. 노드가 1 개인 경우 발생 트래픽의 양이 적다. 하지만, Window Size 를 [그림 3]과 같이 30 개로 설정하였을 경우 재전송 시 Window Size 만큼 큰 데이터를 다시 전송해야 하기 때문에 처리율의 하락을 가져오게 되는 것이다.



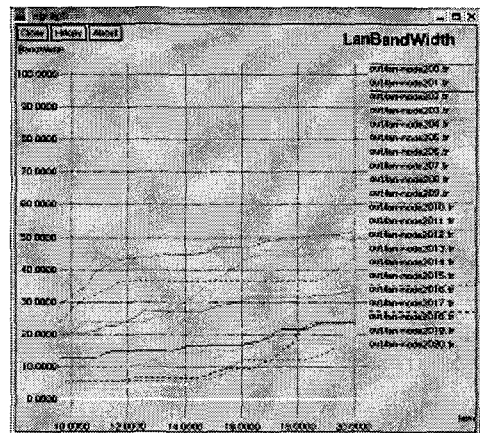
[그림 4] Throughput 변화(트래픽발생 노드수 2)

[그림 4]를 통해서 트래픽 발생 노드 수를 증가시킬 경우도 마찬가지로 처리율의 변화를 확인할 수 있다. 그러나 이때 LAN에서 처리되지 못하고 [그림 5]와 같이 드롭되는 패킷들이 나타난다.

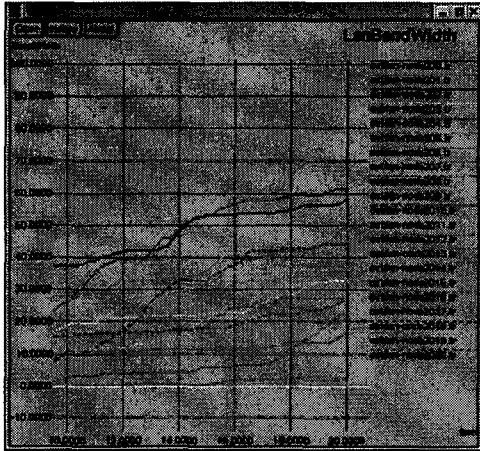


[그림 5] 드롭 데이터(트래픽발생 노드수 2)

[그림 4]와 [그림 5]를 비교를 통해서 드롭되는 양이 처리율에 영향을 주지 않는다는 점도 함께 확인할 수 있다. 트래픽 발생 양이 적을 경우에는 [그림 3], [그림 4]를 통해서 확인한 것과 같이 12, 13 정도의 Window Size에서 최적의 처리율을 보이고 있다. 그러나, LAN은 [그림 1]보다는 많은 수의 노드로 구성되어 있다. 따라서 더 많은 노드를 생성하고 실험을 하기 위해서 [그림 2]와 같이 노드의 수를 20개로 설정하고 LAN의 환경은 [표 2]와 같이 설정하여 실험을 진행하였다. 트래픽의 발생 양이 증가한 만큼 [그림 3]과 [그림 4]에서 도출된 결과보다는 Window Size를 2개정도 더 높게 정하여 [그림 6]과 [그림 7]과 같은 결과를 도출해 내었다.



[그림 6] Window Size 15 개



[그림 7] Window Size 20 개

4. 결론

TCP Parameter의 조절을 통해서 네트워크의 성능을 향상 시킬 수 있다는 것은 다른 연구에서도 증명이 된 사실이다. 네트워크의 성능을 분석하여 가시적으로 확인 할 수 있는 방법은 RTT, 처리율, 장애율, 인터페이스 이용률 등이 있다.

본 연구 논문에서는 내부 접속 요청을 수행하는 서버와 외부의 접속 요청을 수행하는 서버를 분산시켜 내부의 접속 요청을 수행하는 서버의 Window Size 변화에 따른 처리율을 확인해 보았다.

내부 요청을 처리하는 서버의 처리율을 확인 해 본 결과 내부 네트워크에서 발생하는 트래픽 양이 적을 경우 [그림 4]와 같이 향상된 성능을 가지는 Window Size를 도출해 낼 수 있다는 것을 실험으로서 확인 할 수가 있었다.

[그림 2]와 같이 발생하는 트래픽 양이 증가 할 경우 Window Size의 조절을 통해서 [그림 4]처럼 특정 Window Size가 도출되기 보다는 [그림 6], [그림 7]과 같은 결과를 확인 할 수 있다.

[그림 7]에서 Window Size가 20개일 경우에는 전혀 처리를 하지 못하는 LAN 노드가 발생한 반면, Window Size가 15개일 경우 [그림 6]에서 보는 것과 같이 LAN 노드의 전체 분포가 일정수치 사이에 평균적으로 분포하게 되므로 LAN 전체 노드의 성능이 균형적으로 향상된다는 것을 확인 할 수 있다.

이로써 내부와 외부로 접속 요청을 나누어서 서비스 할 경우 내부 접속 요청을 처리하는 서버의 Window Size의 조절이 내부 네트워크의 성능 향상을 가져 다 준다는 결론을 얻을 수 있었다.

향후 연구방향은 정확한 트래픽의 양과 Window Size 변화의 정량적 관계를 명확히 하는 것과 LAN에서 트래픽 발생 노드가 대단위로 증가 할 경우의 Window Size 변화에 따른 처리율을 연구할 계획이다.

참고문헌

- [1] Larry L.Peterson/Bruce S.Davie "Computer Networks" A systems approach. Morgan Kaufmann
- [2] 김진희, 권경희 "재전송 타임아웃 간격의 범위 조절에 의한 Web 서버의 성능향상", 정보처리학회 추계학술발표논문집, 제 9권 제 2호, 2002
- [3] "The Network Simulator ns-2", <http://www.isi.edu/nsnam/ns/>
- [4] "NS by Example", <http://nile.wpi.edu/NS/>
- [5] 권경희, "Web Engineering", 배움터
- [10] W. Richard Stevens, "TCP/IP Illustrated, Volume 1 The Protocols", Addison Wesley, 2001
- [6] 박도용, "무선망에서의 TCP 성능향상 기법에 관한 연구", 건국대학교대학원 석사학위 논문, 2002
- [7] Kevin Fall, Kannan Varadhan, "The ns Manual", <http://www.isi.edu/nsnam/ns/doc/Index.html>
- [8] L.Wood, "introducing ns", <http://www.ee.surrey.ac.uk/Personal/L.Wood/ns/>
- [9] Marc Greis, "ns tutorial", <http://www.isi.edu/nsnam/ns/tutorial>