

차등화 서비스(Diffserv)를 이용한 Web Server Content의 효율적 QoS 보장에 관한 연구

김정윤, 유인태, 박재성

경희대학교 정보통신대학원

e-mail:inokyuni@hanmail.net

A Study on Differentiated Service for Web Server Contents QoS

Jung-Yun Kim, In-Tae Ryu, Jae-Sung Park

The Graduate School Of Information and Communication,
Kyung Hee University

요 약

BE(Best-Effort)방식의 인터넷에서 제공되던 Text위주의 서비스 제공 방식이 Multimedia 위주의 서비스로 점차 바뀌어 감에 따라, 기존의 BE(Best-Effort)방식으로는 Multimedia QoS(Quality of Service)를 보장하기가 어려워졌다. 이에 IETF(Internet Engineering Task Force)에서는 변화하는 인터넷 환경에서 QoS 보장을 위해 Diffserv(Differentiated Service)방식을 제안하게 되었다.

본 논문에서는 Content를 제공하는 Linux Web Server에서 패킷을 클래스 분류기준(Classification Method)에 의해 분류하고 Scheduling Algorithm을 적용하여 DSCP(Differentiated Service CodePoint) 값을 Web Server 자체에서 결정하여 경계라우터(Edge Router)로 전송하는 방식을 취하였으며, 이를 토대로 하여 경계라우터의 Traffic 부하를 줄이고, 경계라우터의 코어라우터(Core Router)화를 통해 더욱더 향상된 Differentiated Service를 제공하는 것이 목적이다. 이를 본 논문에서는 ns2 를 통해 IETF에서 제안된 Diffserv방식과 본 논문에서 제안한 방식의 Diffserv 방식과 현재의 BE방식을 비교하여 어느 정도의 성능 향상이 있었는지 비교 분석하였다.

1. 서 론

인터넷은 불과 몇 년 만에 우리의 생활과 산업 전반적인 면에 걸쳐서 없으면 안될 것으로 성장하였다. 이러한 인터넷을 통해서 서비스를 제공하는 자와(Server)와 그 서버에 접속하여 서비스를 받는자(Client)가 존재하게 되어 QoS(Quality of Service)가 중요시 되게 되었다.

현재 세계적으로 QoS(Quality Of Service)의 향상을 위한 연구가 계속적으로 이루어지고 있는 상황에서 IETF(Internet Engineering Task Force)는 변화하는 인터넷 환경에서 QoS 충족을 위해 IntServ 와 Diffserv를 제안하였다.

기존의 Internet Service는 모든 User가 동일시 취

급되었으며, 동일한 Service를 받는 것처럼 여겨져 모든 Traffic이 FIFO(First-in, First-Out)방식으로 처리되었다. 그러나 VoIP, VPN(Virtual Private Network)등의 서비스가 등장하여 QoS를 요구하게 되었고 기존의 선입선출(FIFO : First-In First-Out)방식을 통해 Service를 하고 있는 인터넷은 위와 같은 요건을 충족시키기에는 역부족이다.

2. IntServ and Diffserv

2-1 IntServ 방식

IntServ(Integrated Service)는 Signalling Protocol인 RSVP(Resource ReSerVation Protocol)을 이용하여 자원을 할당하여 패킷을 전송하는 방법으로써 3

가지 방법으로 제공되어지고 있는데 1번째가 BE(Best-Effort)방식, 2번째가 GS(Guaranteed Service), 3번째가 PS(Predictive Service)이다.

GS방식은 최대한 수용 가능한 서비스를 제공해주는 것이고 PS는 통계적으로 수용 가능한 서비스를 제공해주는 서비스이다.

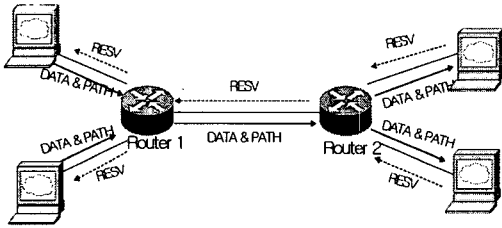


그림 1. IntServ의 구조

그러나 이러한 IntServ는 각각의 경로를 저장하고 있어야 하는 단점으로 인해 확장성이 떨어지는 문제가 가장 큰 단점이다. 이러한 단점은 곧 코어라우터(Core Router)에서는 적용이 어렵다는 것을 의미하는 것이다.

2-2 Diffserv 방식

이러한 IntServ의 단점을 보완하여 코어라우터에서 확장성에 보다 효율적으로 대처한 것이 바로 Diffserv 이다.

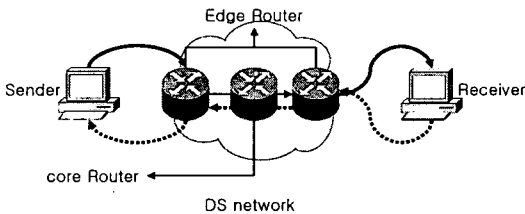


그림 2. DiffServ의 구조

Diffserv는 경계라우터(Edge Router)에서 3개의 Service Class 즉, EF(Expected Forwarding), AF(Asured Forwarding), DE(Default)로 구분하여 DSCP값을 설정한 후 코어라우터(Core Router)로 보내면 코어라우터는 이 DSCP(Differentiated Service Code Point)값을 보고 PHB(Per Hop Behavior)를 수행하는 것이다.

DSCP 값은 IP Header의 TOS (Type Of Service) byte 8bit중 6bit를 사용하여 DSCP값을 설정하므로

써 Packet 별로 차등화된 서비스를 제공하는 것이 가능하다.

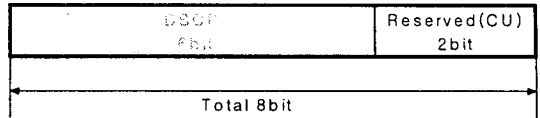


그림 3. Type of Service

3. Linux Web Server에서의 DSCP 설정

Diffserv 환경에서 Traffic이 폭주하면 이 또한 QoS 보장에 어려운 상황이 발생한다. 그리하여 경계라우터에서 행해지고 있는 DSCP값의 설정을 Web Server에서 패킷의 클래스를 분류기준(Classification Method)에 의해 분류하고 Scheduling Algorithm을 적용하여 DSCP값을 Web Server 자체에서 정하여 경계라우터로 전송하는 방식을 채택하게 되면 경계 라우터의 Traffic 부하감소는 물론 경계라우터가 코어라우터의 역할을 수행할 수 있으므로 더욱더 향상된 환경을 제공할 수 있을 것으로 생각되었다.

Linux Web Server에서 Content를 처리하는 방법은 다음과 같다.

3-1 User Level Scheduling(Front-End Processor)

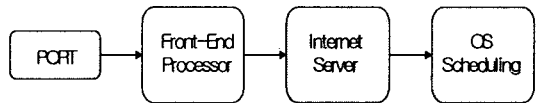


그림 4. Front-End Processor

인터넷 서버와 별개의 모듈(Front-end Processor)를 만들어서 인터넷 서버로 가는 서비스의 요청이 이 모듈을 통과하도록 하는 방식이나, 서비스의 요청이 폭발적으로 증가하게 되면, 이 모듈과 서버와의 통신에 들어가는 과부하를 무시할 수 없게 되어 오히려 효율이 떨어지는 결과를 낳게 된다.

3-2 User Level Scheduling(Revised Internet server)

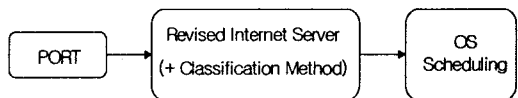


그림 5. Revised Internet Server

인터넷 서버 자체(ex; FTP, Web server)를 수정하여, 내부에서 서비스 품질 정의에 의해 우선순위를 부여하고 분류할 수 있도록 하는 방식이나 인터넷 서버 자체를 수정해야 하는 어려움이 따른다.

3-3 Kernel Level Scheduling

User level의 방법은 서비스를 제공한다 하더라도 그 서비스를 제공하는 프로세스는 결국 OS에 의하여 Scheduling된다. 결국 완벽한 신뢰를 위해서는 OS의 커널 자체를 수정하여, 인터넷 서버가 지정한 우선순위가 OS가 관리하는 우선순위로의 Mapping이 될 수 있도록 해야 하고, 인터넷 서비스는 Time-critical한 작업의 연속이므로 Real-Time Scheduling이 가능한 모듈을 삽입하여 적절한 Scheduling이 이루어지도록 배려할 수 있어야 한다.

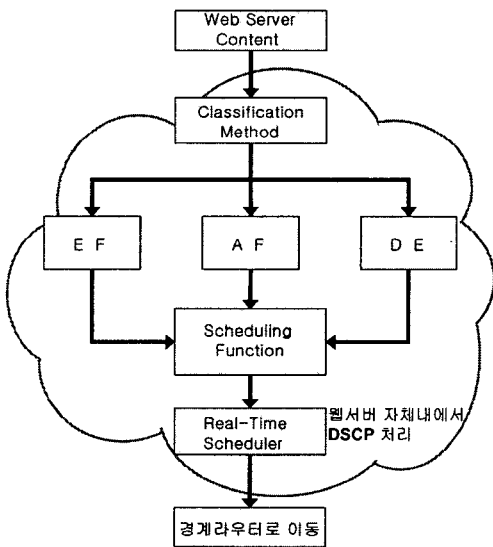


그림 6. Kernel Level Scheduling

3-4 Web Server에서의 Class 분류

웹 서버에서의 클래스 분류기준은 Server와 Client, 또는 서비스 제공자와 서비스 소비자가 어떠한 기준에 따라 우선순위를 정할 것인지에 대한 분류 기준이 필요한데 이러한 분류기준은 서버와 클라이언트 기반의 방법으로 나눌 수 있겠다.

<표1> Web Service의 클래스 분류기준

Server	Web Service 접근 권한 설정
Client	ID, Pass를 통해 사용자 인증, 차등서 비스

3-5 Web Server Scheduling Algorithms

웹 서버에서의 Scheduling Algorithms은 PQ(Priority Queue)방식과 WRR(Weighted Round Robin)방식과 MWRRM2 (Modified WRR Mode 2) 방식이 있지만, 각각의 클래스가 할당된 Weight에 따라 Link의 Capacity를 지니고 있어, 모든 클래스들이 공정하게 서비스를 받을 수 있는 방식인 WRR 방식을 시뮬레이션에서 사용하였다.

4. Simulation 구조 및 환경

4-1 기존의 Diffserv망 구조

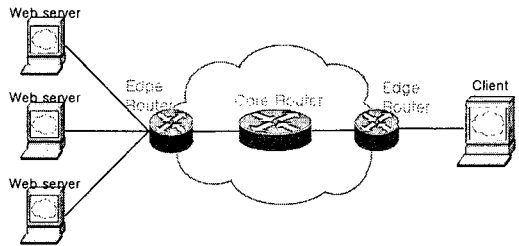


그림 7. 기존의 Diffserv망 구조

DSCP값을 경계라우터(Core Router)설정하여 코어라우터로 보내면 이 정보를 받은 코어 라우터는 이 DSCP(Differentiated Service CodePoint)값을 보고 PHB(Per Hop Behavior)를 수행하는 방식이다.

4-2 제안된 Diffserv망 구조

Linux Web Server에서 패킷을 클래스 분류기준에 의해 분류하고 Scheduling Algorithm을 적용하여 DSCP값을 설정하여 경계라우터로 전송하고, 경계라우터는 DSCP를 설정할 필요 없이 PHB Process를 실행하여 Packet을 전송한다.

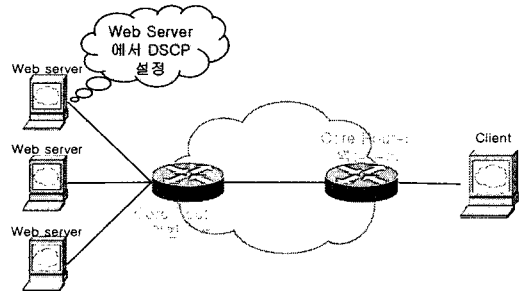


그림 8. 제안된 Diffserv망 구조

4-3 시뮬레이션 환경

Scheduling Algorithm으로는 WRR방식을 사용하였고, 리눅스 Kernel 2.5.17에서 시뮬레이터는 ns2를 사용하였으며, Traffic은 CBR Traffic을 설정하였고, 패킷의 크기는 1024byte로 설정하였다.

5. Simulation 결과

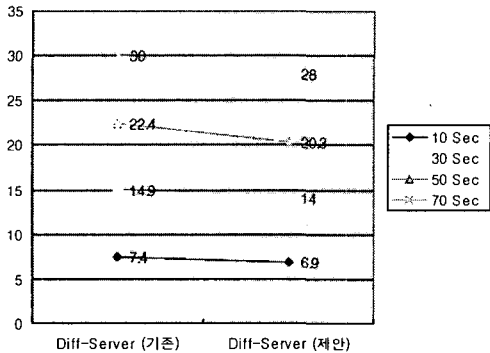


그림 9. 시간 변화에 따른 패킷 손실(Sec/Kbyte)

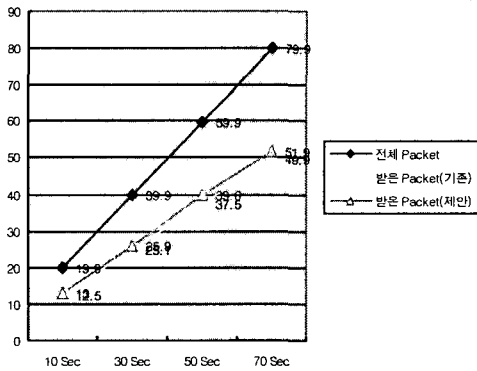


그림 10. 패킷 수신성공 결과(Sec/kbyte)

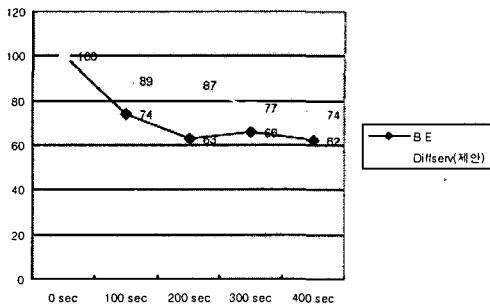


그림 11. BE 방식과 Diffserv(제안)비교(Sec/kbyte)

기존의 Diffserv 보다 제안된 Diffserv의 패킷 손실이 더 적은 것을 알 수 있었으며, 수신 성공률에 있어서도 보다 나은 결과를 보여주었고, 또한 기존의 BE방식과 비교 했을 때 훨씬 나은 성능을 제공하고 있음을 알 수 있었다.

6. 결론 및 향후 연구방향

Edge Router의 역할을 Web Server에서 처리 하는 것으로 Edge Router의 부하를 줄여 줌으로써 더욱더 원활한 Packet 전송이 이루어 졌다는 것을 검증하고 평가했으며, 시간의 흐름이 길어지면 길어질 수록 제안된 Diffserv방식이 BE방식과 Diffserv(기존) 방식보다 다소 효율적인 방식이라는 사실을 검증하고 평가했다는 측면에서 매우 의미 있는 연구라고 하겠다.

향후 연구내용으로는 OS Scheduling별로 최적의 Web Server Content 전송방식을 찾아내 이를 적용하여, 더욱더 효율적인 Content 전송방식을 찾아내는데 있다.

참고문헌

- [1] Providing packet-loss guarantees in DiffServ Architectures Hossam S. Hassanein May, 2002
- [2] 조유근, 리눅스 커널 Programming, January, 2002
- [3] 김점분, 김정호, 서인옥, Linux Kernel Scheduling, May, 2002
- [4] 홍원기, 김재영, 김명섭, 류숙현, Development of Internet Server Technology supporting Differentiated QoS services. Postech, June, 2001.
- [5] Stevens, W.R. TCP/IP Illustrated Volume.2. Addison Wesley. November, 2001.
- [6] William Stallings, Operating Systems. Addison Wesley. November, 2000.
- [7] Daniel P. Bovet, Marco Cesati, Understanding the Linux Kernel, O'Reilly. November, 2000.
- [8] Maurice J. Bach, The Design of the UNIX Operating System. Prentice Hall.
- [9] S. Blake et al., "An Architecture for Differentiated Services", Internet Draft<draft-ietf-Diff-serv-framework-00.txt>, May, 1998.
- [10] K. Nichols et al., "Definition of the Differentiated Services Field(DS Field) in the IPv4 and IPv6 Headers". RFC2474, December, 1998.