

Diffserv Network에서 Host 간의 QoS

보장에 관한 연구

*김정운 *박재성 *유인태 **변옥환

*경희대학교 정보통신대학원

**한국과학기술정보연구원

inokyuni@hotmail.com

A Study on a QoS Provisioning for Host-to-Host in Diffserv Network

*Jung-Yun Kim, *Jae-Sung Park, *In-Tae Ryu, **Ok-Hwan Byeon

* The Graduate School of Information and Communication,
Kyung Hee University

** Korea Institute of Science and Technology Information

요 약

변화하는 인터넷 환경에서의 QoS 보장은 빼놓을 수 없는 중요한 과제로 각종 제안들이 나오고 있다. 본 논문에서는 IETF(Internet Engineering Task Force)에서 제안한 Diffserv 네트워크에서 Host-to-Host의 QoS 보장에 관하여 다루었으며, DSCP(Differentiated Service CodePoint)값의 설정을 Edge Router가 아닌 Host에서 수행하여, 경계라우터의 Traffic 부하를 줄이고, MRRR2(Modified Weighted Round Robin Mode2) 방식의 스케줄링을 통해 더욱더 향상된 Diffserv network에서의 Host-to-Host QoS를 제공하는 것이 목적이다.

본 논문은 IETF에서 제안된 Diffserv 방식과 본 논문에서 제안된 Diffserv 방식의 효율성 입증을 위하여, ns2 시뮬레이션을 사용하여, 비교 분석하였다.

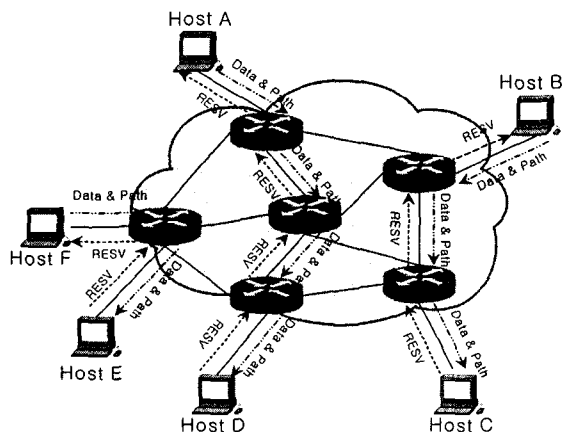
1. 서 론

인터넷 환경의 발달로 인해 IP망에서의 QoS(Quality of Service)적용은 피할 수 없는 당면과제로 등장하고 있다. 현재의 FIFO(First-In-First-Out)방식의 인터넷 서비스는 모든 사용자와 트래픽을 동일시 취급하여, 먼저 들어온 패킷은 먼저 처리하고 나중에 들어온 패킷은 나중에 처리하는 방식을 취한다. 만약 네트워크 혼잡 상태가 발생하여, 먼저 들어온 패킷이 처리가 되지 못하는 상황에서 계속적으로 패킷이 들어오면, 그 패킷은 드롭되어 진다. 그러나, 데이터 전송을 서비스하던 기존의 네트워크 환경이 멀티미디어로 변해감에 따라서, QoS의 중요성이 더욱더 부각되게 되었다. 이에 IETF에서는 IntServ와 Diffserv를 제안하였다.

2. Intserv and Diffserv

신호프로토콜(Signalling Protocol)인 RSVP(Resource ReSerVation Protocol)을 사용하는 IntServ의 RSVP는 단일 모드로 동작이 되며, 호스트는 서로 다른 방향에서 자원을 요청하게 된다. RSVP가 ATM 신호 Protocol과

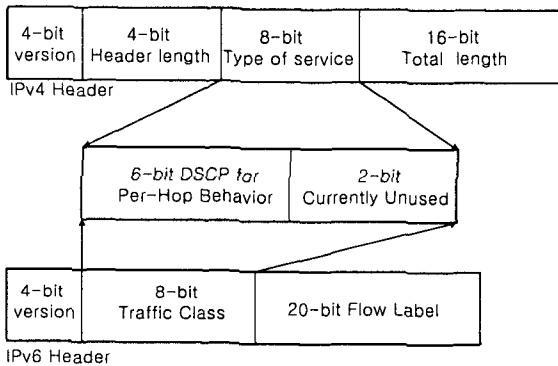
크게 다른 점은 수신자가 자원 요청을 수행하는 것이라 하겠다.



[그림 1] IntServ Service Structure

보장형 서비스(Guaranteed Service)나 부하제어형 서비

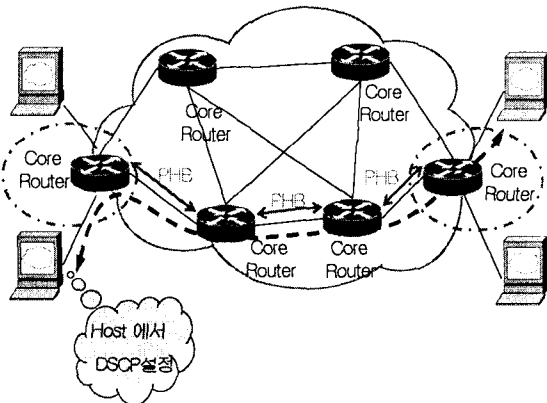
스(Controlled Load Service)를 요구하는 패킷들은 데이터 전송 전에 경로 설정과 자원 예약을 필히 하여야 하는데, 경로상의 모든 라우터에서 발생하는 자원예약 프로세스는 전체 시스템에 큰 영향을 줄 수 있을 만큼의 부하를 가져오게 되는 결점을 가지고 있다. 특히 위의 [그림 1]과 같이 Host F에서 Host E로 패킷을 전송하는 'Short-Lived Flow' 같은 경우, 자원을 예약하는 과정 중 발생하는 부하가 실제 데이터 흐름보다 많기 때문에 비효율적이다. 이러한 단점을 보완하기 위하여, Diffserv가 제안되었다. Diffserv는 이러한 신호프로토콜을 사용하지도 않고, 자원 예약의 번거로움도 없으며, 이로 인한 네트워크 부하도 적다. 또한 자동화 된 서비스제공이 가능하게 되었다.



[그림 2] TOS Byte 설정

경계 라우터(Edge Router)에서 IP Header부분의 TOS byte에 DSCP(DiffServ Code Point)값을 크게 3가지로 구분하여 설정하고, 중심 라우터(Core Router)는 DSCP 설정에 의해서 우선 순위가 높은 패킷을 선 처리하는 방식을 통해 자동화된 서비스를 제공하게 된다.

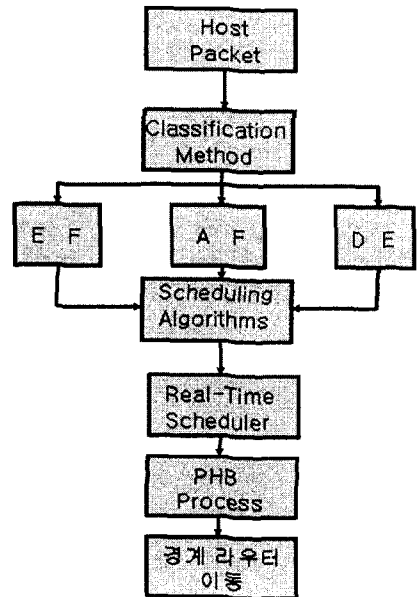
3. 제안한 Host 구조



[그림 3] 제안된 Diffserv 방식 및 Simulation 구조

위의 [그림 3]을 보면 본래의 Diffserv 방식은 Edge Router에서 DSCP 값을 설정해, Core Router로 보내고, Core Router는 DSCP 값을 보고 PHB(Per Hop Behaviors)를 수행하게 되는 방식이다. 그러나 본 논문에서 제안된 방식은 DSCP 설정을 linux Kernel 2.5.0 부터 지원되는 Diffserv kernel의 수정을 통해 직접 Host에서 DSCP를 설정하여, Edge Router로 보내고, Edge Router는 바로 PHB를 수행하는 방식을 제안하였다. Host Scheduling Algorithms는 MRRR2(Modified WRR Mode2)방식을 사용하였으며, linux를 바탕으로 하는 Host에서의 DSCP 설정을 위해 User lever Scheduling 방법과 Kernel 방식의 Scheduling중 Kernel Scheduling 방식을 사용하였다.

만약 User level Scheduling을 사용하여 DSCP를 설정한다고 하더라도, 결국에는 OS에 의하여 Scheduling이 되므로, 완벽한 신뢰를 얻기 위해서 OS의 Kernel 자체를 수정하여, Host가 지정한 우선 순위가 OS에서도 우선 순위로 Mapping이 될 수 있도록 해야 하고, 이러한 DSCP설정은 Time-critical한 작업의 연속이 될 수 있으므로 Real time Scheduling이 가능한 Module을 삽입하여야 한다. 그러므로 Kernel에서 Scheduling이 이루어지도록 했다.



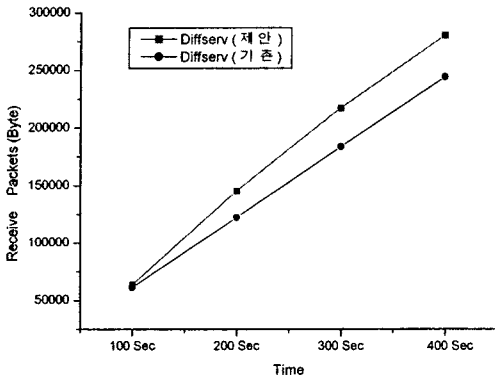
[그림 4] linux Kernel에서의 DSCP 설정과정

제안된 Diffserv 방식과 기존의 IETF에서 제안된 Diffserv와의 성능평가를 위해서 ns2 Simulator를 이용하였으며, [그림 3]을 참고로 하여, 종단의 Host가 종단의 Host로 패킷을 보내며, 4개의 라우터를 거치게 된다.

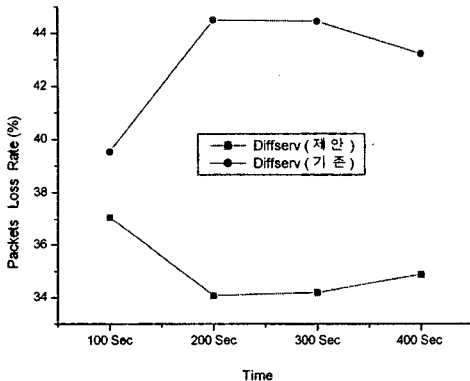
본 논문에서의 모든 실험 환경은 Diffserv 네트워크가 이루어졌고, 제안된 Diffserv모듈이 Host상에 올라와 있다는 것을 가정 하에 Simulation이 진행되었다. Traffic은 CBR(Constant bit rate)을 사용하였고, 패킷의 크기는

1024byte이며, MRRR2(Modified WRR Mode2) Scheduling Algorithms을 사용하였다.

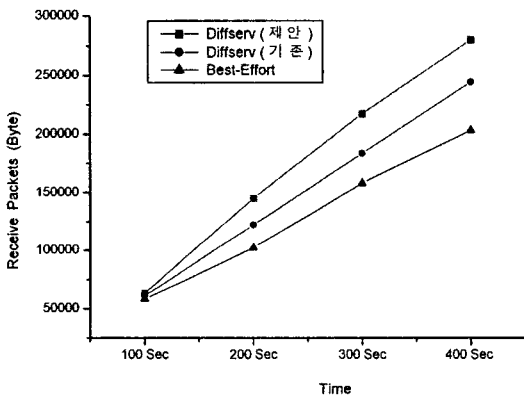
4. 시뮬레이션 결과



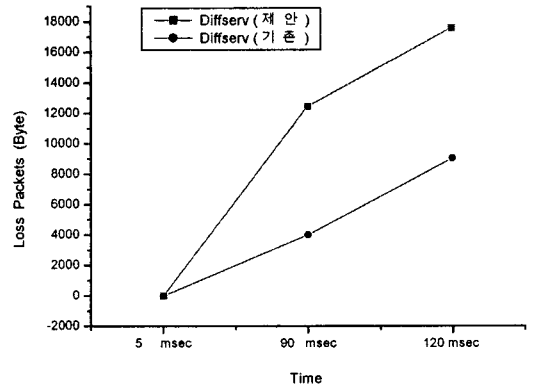
[그림 5] 시간변화에 따른 패킷 수신량



[그림 6] 시간변화에 따른 패킷 손실률(%)



[그림 7] BE, Diffserv(제안), Diffserv(기존) 비교



[그림 8] 지연시간 증가에 따른 패킷 손실량

5. 결론

만약 Diffserv 방식에서 EF 트래픽이 폭주하게 되면, 다른 하위 트래픽은 QoS를 보장받지 못하는 어려움이 있을 수 있으므로, 본 논문은 Diffserv 환경에서 트래픽이 폭주하였을 경우에 보다 원활한 패킷 전송을 위해 호스트에서 DSCP 값을 결정해 Edge Router로 전송하도록 하고, Edge Router는 DSCP값을 설정하는 것이 아니라 바로 Core Router로 PHB를 실행하여, Edge Router의 트래픽 부하 및 DSCP 설정으로 인한 부하를 줄여 Host에서 패킷을 전송하였을 때, 기존에 제안된 Diffserv보다 좀더 효율적으로 패킷을 전송할 수 있는 것에 대해 알 수 있었고, 지연 시간증가 등등에도 본 논문에서 제안된 방식이 좀더 효과적이라는 사실을 검증하고 평가하였다.

참 고 문 헌

- [1] s. Blake et al., "An Architecture for Differentiated Services". Internet Draft <draft-ietf-diff-serv-framework-00.txt>, May, 1998
- [2] J. Wroclawski, "The Use of RSVP with IETF Intergrated Services", RFC 2210, September 1997
- [3] Development of Internet Server Technology supporting Differentiated QoS Services. June, 2001
- [4] K. Jonson et al. "A queue scheduling method for differentiated services", RFC2989, December 1999
- [5] Linux Kernel Scheduling, May, 2002 [1] Providing packet-loss guarantees in Diffserv Architectures S. Hassanein May, 2002