

DiffServ 네트워크에서 QoS 보장형 트래픽 제어 시스템 설계 및 구현

이 명 섭, 김 환 섭, *류 명 춘, 박 창 현
영남대학교 컴퓨터공학과, *경운대학교 컴퓨터공학과
전화 : 053-810-1504 / 핸드폰 : 011-810-7101

Design and Implementation of Traffic Control System in Differentiated Services Networks

Myung Sub Lee, Hwan Sub Kim, Myung Choon Ryu, Chang Hyeon Park
Dept. of Computer Engineering, Yeungnam University
E-mail : skydream@cse.yu.ac.kr

Abstract

QoS support for the Internet traffic becomes an important future requirement of various network devices such as routers and switches. Differentiated Services(DiffServ) is considered as an efficient feasible solution for providing different service characteristics to different classes of network users. IETF DiffServ working group has already defined a general architecture of DiffServ and is extending its detail features. Network vendors also start implementing DiffServ enabled network devices. However, the management of DiffServ is not fully standardized yet. This paper, we present methods of traffic flow control in DiffServ Networks. we have designed a DiffServ Traffic Control system.

I. 서론

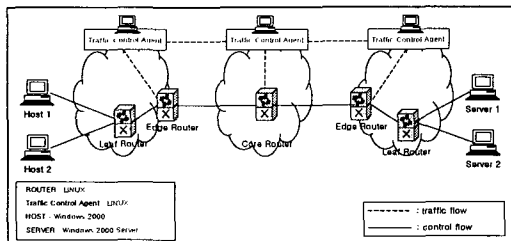
최근 들어 인터넷 사용자가 늘어남에 따라 인터넷에서 구할 수 있는 정보도 빠른 속도로 늘어나고 있으며, WWW (World-Wide-Web)의 대중화와 함께 다양한 인터넷 서비스가 제공되고 있다. 인터넷 서비스를 제공하는 다양한 응용 프로그램들이 네트워크 기술과 관련되어 개발되면서 네트워크 트래픽은 계속적으로 증가하고 있다.

또한, 네트워크 대역폭의 증가율이 네트워크 사용율의 증가에 비해 매우 늦어지기 때문에 네트워크 사용자의 대역폭 요구는 더욱 높아지고 있다. 그러나 TCP/IP 프로토콜을 기반으로 한 현재의 인터넷은 모든 패킷을 동일하게 전달하는 FIFO(First In First Out)방식으로 동작하는 최선형(best effort) 서비스만을 제공하고 있기 때문에 서비스에 따른 패킷의 손실 또는 지연 등의 QoS(Quality of Service)에 대한 요구 사항을 보장해 주지 못하고 있다. 따라서 인터넷에서 서비스의 QoS를 보장해 주기 위해서는 현재의 모델과는 다른 새로운 서비스 모델을 필요로 한다. 실시간 응용 서비스가 요구하는 QoS를 지원하기 위해 새로운 서비스 모델에 기반을 둔 IP(Internet Protocol) 패킷전달 방식에 대한 연구가 최근 수년간 IETF WG(Internet Engineering Task Force Working Group)에서 연구되고 있으며 대표적인 서비스가 IntServ(Integrated Services)^[1]와 DiffServ(Differentiated Services)^{[2][3]}이다. IntServ 모델은 실시간 응용 서비스에서 발생하는 패킷의 흐름을 시그널링 프로토콜인 RSVP(Resource reSerVation Protocol)^[4]를 이용하여 연결수락제어와 자원 예약을 수행하여 전달지연을 보장해 주는 모델이다. 그러나 IntServ 모델은 각 패킷 흐름에 대한 상태 정보를 라우터가 유지하고 있어야 하기 때문에 현실적으로 수용하기에 어려움이 있다. 따라서 확장성 문제를 가지고 있는 IntServ 모델의 한계를 극복하고 인터넷 백본 망에서 적용할 수 있는 서비스 모델로 DiffServ 모델의

구조 및 관련 표준안이 개발되고 있다. 본 논문에서는 선별적인 QoS를 보장하기 위해 IETF에서 제안한 DiffServ지원 라우터를 구현한다. 우선순위에 기반 한 선별적인 포워딩을 구현하기 위해 리눅스의 큐 구조를 조작하였으며, 이러한 차별적 지원 라우터를 효율적으로 제어 관리하기 위한 트래픽 제어 에이전트를 구현한다.

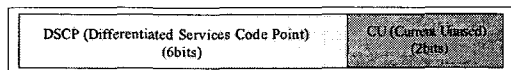
II. 요구사항

IntServ모델과는 달리 DiffServ에서는 IP 헤더의 TOS(Type Of Service) 부분 중 6비트를 QoS를 정하는 부분으로 바꾸어 사용하여 모든 트래픽 요구를 QoS에 따라 분류하고 이에 따라 트래픽을 통합(aggregation)함으로써 스케줄링을 수행한다. 또한, 목적지와 출발지간의 어떠한 QoS정보도 교환되지 않으므로 RSVP가 가지고 있던 연결설정 비용에 대한 문제를 제거하였다. 그러나 DiffServ에서는 IntServ에서 제공하는 자원예약 등의 기능이 없기 때문에 특정 수준의 서비스를 보장하는 것은 매우 어려운 일이다. 따라서 DiffServ에서는 어떤 특정 수준의 서비스 품질을 보장하기 보다는 각각의 흐름 집합에 대한 규칙에 근거하여 상대적 서비스 품질보장을 제공하도록 한다. 위와 같은 단점을 보완하기 위해 본 논문에서는 트래픽 제어 에이전트를 두어 망의 상황을 모니터링하고, 이에 따라 DiffServ지원 라우터를 제어하게 한다.



[그림 1] DiffServ 네트워크 구성도

[그림 1]에서 트래픽 제어 에이전트는 각 도메인 내에 위치하며 도메인의 유효자원의 모니터링과 트래픽에 대한 모니터링을 통한 수락제어와 정책 제어 역할을 담당한다. 각 트래픽 제어 에이전트 간에는 서비스 단계의 협약을 체결하고 있으며 일정한 회선에 대한 대역폭 보장을 지원한다. DiffServ 구조에서 차별화 서비스를 지원하기 위해 라우터는 각각의 패킷의 흐름(flow)을 분류할 수 있는 분류화(classification)기능을 지원해야 한다. 흐름을 분류하고 이를 표시하는데 있어서 DiffServ에서는 IP헤더의 TOS 필드를 이용한다.



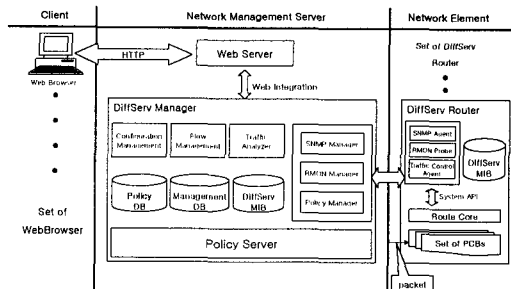
[그림 2] TOS 필드의 구조

TOS 필드에서 첫 번째 필드의 6비트는 DSCP (Differentiated Services Code Point)로 구성되어 있으며, 일부는 종단간 처리를 위해 사용되며 나머지는 도메인내의 서비스를 제공하기 위해 사용되어 진다. 두 번째 필드의 2비트는 현재 사용되지 않으며 미래를 위해 예약되어져 있다. 본 논문에서는 분류화를 위해 패킷의 DSCP필드를 근거로 패킷의 집합이 어디에 속하는가에 대한 분류와 마킹(marking)을 수행한다. 분류화에 의해 패킷의 흐름이 분류된 다음 그 패킷이 소비하는 대역폭을 알아야 한다. 해당 흐름이 대역폭의 임계치를 초과하는지를 판단하기 위해 라우터는 일정시간동안의 흐름을 검사하게 되지만 사용자가 보내는 데이터의 양은 매우 유동적이기 때문에 이러한 방법은 충분하지 않다. 따라서 라우터는 흐름 율을 측정할 수 있어야 할 뿐만 아니라 트래픽의 집중(burst)정도도 파악하는 미터링(metering) 기능을 수행할 수 있어야 한다. 본 논문에서는 이러한 기능을 제공하기 위해 일정 시간마다 라우터의 자원 할당 상황을 덤프 하여 트래픽 제어 에이전트에게 보내주는 루틴을 구현한다. 미터링 결과 흐름이 집중 데이터를 포함하고 있을 때 라우터는 이를 처리하기 위해 여러 가지 방법을 선택할 수 있다. 집중 데이터를 처리하거나 트래픽의 속도를 조절하는 것을 셰이핑(shaping)이라고 하며, 본 논문에서는 셰이핑 과정에서 흐름이 임계치를 초과할 경우 라우터는 흐름에서 하나 이상의 패킷을 폐기하는 드롭핑 방법을 구현하여 흐름이 일정한 대역폭을 보장 받을 수 있도록 하였다. 또한 네트워크에서 혼잡이 발생했을 때 특정 패킷의 지연을 최소화하기 위해 DSCP 필드에 패킷이 가장 좋은 링크를 통해 전달될 수 있도록 그 값을 설정해 주도록 하는 EF(Expedited Forwarding) 서비스를 구현한다.

III. 트래픽 제어 시스템

본 논문에서 구현한 트래픽 제어 시스템의 구조는 [그림 3]에서와 같이 3계층 구조를 가지며 클라이언트, 네트워크 관리서버, 네트워크 엘리먼트로 구성되어져 있다. 클라이언트 프로그램은 웹 브라우저 내에서 동작하며 네트워크 관리 서버에 의해 처리된 네트워크 관리 정보를 GUI 형태로 관리자에게 제공하며 DiffServ라우터의 트래픽 제어를 위한 GUI기능을 담당한다. 네트워크 관리 서버는 웹 서버와 DiffServ 관리자를 포함하고 있으며 DiffServ 관리자는 구성관리와 트래픽 분석을 수행하기 위해 DiffServ 라우터와 기타 네트워크장비에서 수집된 트래픽 정보와 DiffServ MIB정보를 데이터베이스에 저장한다. 또한 흐름 관리를 수행하기 위해 정책 관리자, 정책 서버 그리고 트래픽 제어 에이전트

를 이용한다. SNMP^[5]매니저와 RMON매니저는 DiffServ라우터내의 SNMP 에이전트, RMON 프로브와 관리 정보들을 교환하여 네트워크 관리를 수행한다. 웹 서버는 웹 기반의 관리 인터페이스를 제공하는 역할을 수행한다. 그리고 네트워크 장비로는 차별화 서비스를 지원하는 라우터로 구성되어 있다. [그림 3]에서 전체 시스템 구성도를 보이며 각 구성요소에 대한 세부 설명은 각 세부 절에서 설명된다.

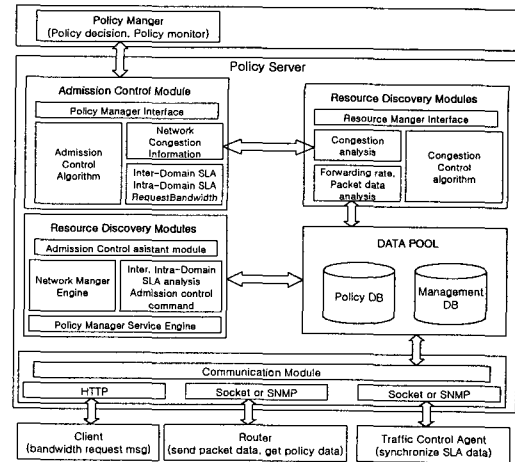


[그림 3] 트래픽 제어 시스템 전체 구성도

DiffServ 트래픽의 흐름관리를 수행하기 위한 시스템 구성도를 [그림 4]에서 보이며 구체적인 동작방식을 아래에서 보인다. 먼저, 클라이언트로부터 통신모듈내의 HTTP를 사용하여 정책 서버에 대역폭 요구를 받게 되며 이 정보를 데이터베이스에 저장하고, 데이터베이스내의 SLA 테이블의 정책과 일치하는지 확인을 하게 된다. 만약 동일한 서비스 정책을 이전에 맺었던 적이 있다면 서비스 종료 일자를 확인하고 그 서비스가 유효한 기간이면 서비스 가능 메시지를 다시 통신모듈을 통하여 클라이언트에게 전송한다. 새로 갱신된 정책인 경우에는 트래픽 제어 에이전트 내의 인증제어 모듈에서 데이터베이스에 있는 정보를 바탕으로 현재 망의 상태를 감시하는 자원검색모듈을 거쳐 서비스가 가능한 상황이면 트래픽 제어 에이전트 간의 통신을 통하여 서비스 수락 여부를 판단하게 된다. <표 1>에서 클라이언트와의 정책 결정과정과 대역폭 할당 수락 절차를 보인다.

<표 1> 클라이언트와 에이전트 간 제어 정보

- | |
|--|
| <ol style="list-style-type: none"> ① 클라이언트는 트래픽 제어 에이전트로 서비스 가능 여부를 묻는다. ② 현재 도메인에서 가능한 서비스이면 다시 패킷 송출 대상 도메인으로 서비스 가능 여부를 묻는 메시지를 전송한다. ③ 마지막 도메인의 트래픽 제어 에이전트까지 메시지 전송이 되고 최종적으로 서비스가 가능하다고 판단이 내려지면 서비스 수락 메시지가 돌아오게 된다. |
|--|



[그림 4] 트래픽 흐름 관리를 위한 시스템 구성도

<표 1>과 같은 방법으로 각각의 트래픽 제어 에이전트에서는 데이터베이스내의 정보와 인증 제어 모듈의 정책 처리 절차를 따라 서비스 가능여부를 결정하게 된다. DiffServ 관리자는 통신 모듈을 통하여 라우터로부터 망 자원에 관한 정보를 가져오는 역할과 라우터의 QoS를 설정 역할을 담당한다. 이때, 라우터로부터 수집된 정보는 데이터베이스에 저장되고, 수집된 데이터를 기반으로 자원 검색 모듈에서는 현재 망에서 서비스가 가능한지를 계산하게 된다. 계산된 정보는 인증제어 모듈에서 클라이언트로부터의 요청에 대한 수락 여부를 판단하는 기준으로 삼게 되며 관리자가 효율적으로 망 자원을 모니터링 할 수 있도록 도와주는 정보로도 사용되게 된다. 인증제어 모듈에서 서비스가 가능하다는 판단이 내려지면 라우터로 QoS 설정을 위한 구성메시지를 보낸다.

본 논문에서 사용된 트래픽 제어 에이전트와 라우터 간의 SLA요청 동작알고리즘은 다음과 같다.

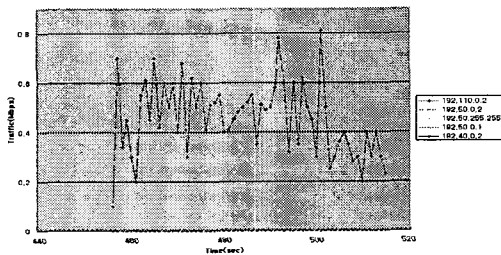
- ① 트래픽 제어에이전트는 데이터베이스에서 SLA 정책을 유지하고 있다가 정책 기간이 완료되면 정책을 지운다.
- ② 새로운 SLA 요구가 생성되었을 때 트래픽 제어 에이전트는 요구된 요청 타입, 전송율을 비교한다.
 - ㉔ 만약 요청에 맞는 정책이 없으면 트래픽 제어 에이전트는 Edge라우터로 파라미터가 분류된 새로운 클래스의 라우터 구성 메시지를 보내고 클래스에 DSCP를 할당한다.
 - ㉕ 비슷한 요청이 있으면, 구성 메시지를 보내지 않는다
 - ㉖ 대역폭 할당 요청에서 사용되는 SLA ID가 사용자에게 반환이 된다.

클라이언트와 트래픽 제어 에이전트, 트래픽 제어 에이전트와 트래픽 제어 에이전트간의 BAR 동작 알고리즘은 다음과 같다.

- ① 트래픽 제어 에이전트에서는 데이터베이스 안에서 SLA정책을 유지하고 있다가 정책 기간이 완료되면 정책을 지운다.
- ② 트래픽 제어 에이전트가 대역폭할당 요청을 받으면 SLA ID를 사용해서 대역폭 할당 요청이 서비스 단계 협약에 일치하는지를 식별하고 요청을 허용할지 결정한다.
- ③ 요청이 수락되면 트래픽 제어에이전트는 대역폭 할당 요구시에 분류된 라우터 ID에 맞는 Leaf 라우터정책 수행과 마킹을 위해 라우터 구성 메시지와 흐름에 할당된 DSCP를 보낸다.

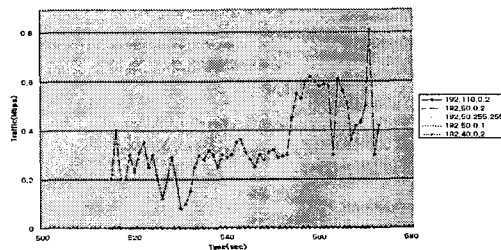
IV. 실험

본 논문에서는 높은 대역폭을 요구하는 동영상 스트림을 생성하기 위해 Windows Media Encoder 7.0을 사용하여 8M의 advanced stream format인 파일을 전송하였고, 각 도메인의 대역폭 브로커와 라우터에서 패킷 생성기를 사용하여 임의로 트래픽을 발생시켰다.



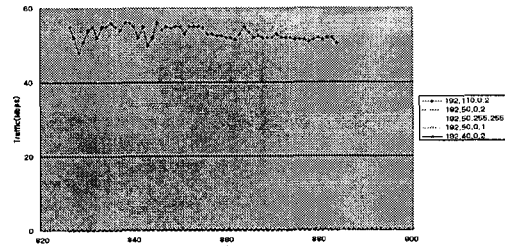
[그림 5] 일반적인 트래픽 상태

[그림 5]은 일반적인 트래픽 상태에서의 그래프로 도메인 1의 서버 1과 서버 2에서 1.5Mbps로 동영상 스트림을 동시에 전송할 때 그래프와 같이 망 내에서 각 IP별 패킷의 상태는 트래픽의 차이가 많이 나지만 어느 정도 빠른 전송률을 유지하고 있다. 그러나 각 도메인에서 패킷 생성기를 사용하여 호스트에 연결된 라우터1을 목적으로 트래픽을 발생시켰을 경우 [그림 6]와 같이 전송률이 현저히 줄어드는 것을 볼 수 있다. 호스트 1과 호스트 2에서는 동영상 파일의 재생이 트래픽으로 인해 정지된 상태이다.



[그림 6] 트래픽 발생 후의 트래픽 상태

[그림 7]은 호스트 2로 가는 동영상 스트림에 정책을 적용하고 호스트1에는 정책을 적용하지 않은 Best Effort상태이며 트래픽 발생 후에도 정책을 적용하기 전보다는 훨씬 높은 전송률을 유지하면서 패킷이 흘러가고 있다. 호스트 1은 여전히 트래픽으로 인해 스트림의 재생이 중지된 상태이다.



[그림 7] 정책 적용 후의 트래픽 상태

V. 결론

본 논문에서는 선별적인 QoS를 보장하기 위해 IETF에서 제안한 DiffServ를 구현하였다. 우선순위에 기반한 선별적인 Forwarding을 구현하기 위해 리눅스의 큐 구조를 조작하였으며, 이러한 차별적 지원 라우터를 효율적으로 제어 관리하기 위한 트래픽 제어 에이전트를 구현하였다.

참고문헌

- [1] IETF, Integrated Services(IntServ) Working Group. <http://www.ietf.org/html.charters/intserv-charter.html>
- [2] IETF Differentiated Services(diffserv) Working Group. <http://www.ietf.org/html.charters/diffserv-charter.html>
- [3] RFC2998: Bernat, Yoram; Ford, Peter; Yavatkar, Raj; Baker, Fred; Zhang, Lixia; Speer, Michael; Braden, Bob; Davie, Bruce; Felstaine, Eyal; Wroclawski, John. Aframework for Integrated Services Operation over Diffserv Networks, IETF, November 2000.
- [4] R.Braden, L.Zhang "Resource ReSerVation Protocol(RSVP)- Version 1 Functional Specification", RFC2205, Sep.1997.
- [5] S.Handelman, S. Stibler, N. Brownlee and G. Ruth, "RTFM: New Attributes for Traffic Flow Measurement", IETF RFC 2724, October 1999.