차별화서비스 네트워크에서 흐름 관리를 위한 트래픽 제어 에이전트

이 명 섭, 박 창 현
영남대학교 컴퓨터공학과
전화 : 053-810-1504 / 핸드폰 : 011-810-7101

A traffic control agent to manage flow usage in Differentiated Service Network

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

Abstract

This paper presents a traffic control agent that can perform the dynamic resource allocation by controlling traffic flows on a DiffServ network. In addition, this paper presents a router that can support DiffServ on Linux to support selective QoS in IP network environment. To implement a method for selective traffic transmission based on priority on a DiffServ router, this paper changes the queuing discipline in Linux, and presents the traffic control agent so that it can efficiently control routers, efficiently allocates network resources according to service requests, and relocate resources in response to state changes of the network.

I. 서론

실시간 응용 서비스가 요구하는 QoS를 지원하기위해 새로운 서비스 모델에 기반을 둔 IP(Internet Protocol) 패킷전달방식에 대한 연구가 최근 수년간 IETF WG에서 연구되고 있으며, 대표적인 서비스가 *IntServ[1]*와 *DiffServ[2]*이다. *IntServ*모델은 단대단 패킷 전송지연시간을 극복할 수 있도록 자원 예약을 위한 RSVP 신호프로토콜을 사용하여 각 흐름별로 자원을 예약한 후 패킷을 전송하도록 하는 방식이다. 이 모델은 흐름별로 각기 다른 QoS가 요구되므로, 각 서비스 유형에서 요구되는 흐름명세(flow spec) 파라메타를 포함하고 있다. 따라서 실시간 응용서비스가 요구하는 전송품질을 보장하기위해 RSVP는 흐름명세 파라메타를 이용하여 종단 호스트와 망 노드 사이에 필요한 대역폭을 요청, 처리한다.

그러나, IntServ모델을 지원하기 위해서는 각 중간라우터에서 흐름에 관련된 정보를 저장해야하기 때문에, 라우터에서 이런 정보를 저장하는 공간을 요구할 뿐만 아니라 라우터의 처리 오버헤드가 커질 수 있다. 특히, 인터넷 백본라우터의 경우 전송속도가 상당히 높고 연결된 흐름의 개수가 매우 많으므로 코어노드에서 IntServ모델을 지원하기가 매우 힘들다[3]. 따라서 확장성 문제를 갖고 있는 IntServ모델의 한계를 극복하고 인터넷 백본망에서 적용할 수 있는 새로운 서비스모델로 DiffServ모델의 구조 및 관련 표준안이 개발되고 있다.

DiffServ모델에서는 우선적으로 QoS를 몇 개의 클래스로 분류하고 분류된 클래스에 따라 서비스의 전송품질을 보장하도록 한다. 이에 따라 DiffServ모델은 IP헤더의 특정 필드(IPv4 : ToS Field, IPv6 : Traffic Class Field)에 QoS정보를 마킹(marking)하여 차별화서비스를 설정하게 되며, 이렇게 설정된 QoS정보들에 의해 적절한 전송(PHB : per hop

behavior)[4]을 수행하는 구조를 가진다. DiffServ모델에서는 차별화서비스 도메인의 경계라우터에서 IP 헤더의 ToS필드에 DSCP(Differentiated Service Code Point)를 설정하여 트래픽 흐름을 분류한다. 그리고 네트워크 내에서는 분류된 트래픽 흐름에 따라 자원할당, 패킷 폴리싱, 스케줄링 등을 수행하도록 한다. 즉, DiffServ 모델에서는 경계노드와 코어 노드의 역할을 분리시켜 IntServ 구조의 확장성 문제를 해결하고자 하는 것이다.

DiffServ모델에서는 클라이언트에서 요구되는 자원과 망 노드에 설정된 SLA(Service Level Agreement)[5]정보를 바탕으로 자원할당방법을 결정하는 장치가 필요하며, 이러한 기능을 하는 장치를 BB(Bandwidth Broker)[6]라 한다. BB는 SLA정보에 따라 할당된 대역폭을 저장하는 데이터베이스를 유지하고 있으며, 데이터베이스에 저장된 정보는 향후 대역폭 할당을 결정하기위한 기초정보로 이용된다.

IntServ모델의 문제점을 극복하기 위해 제안된 DiffServ모델은 현재 이의 구현을 위한 연구개발이 진행 중에 있다.

본 논문에서는 IP네트워크 환경에서 선별적인 QoS를 보장하며 클라이언트의 대역폭 요구에 대한 동적 자원할당을 수행하는 트래픽 제어 에이전트를 제시한다. 본 논문에서 제시하는 트래픽 제어 에이전트는 망 노드의 트래픽 정보를 수집하고 분석하여 동적자원할당 기능을 수행한다.

II. 트래픽 제어 에이전트

본 논문에서의 트래픽 제어 에이전트는 클라이언트에서 요구되는 자원과 설정된 SLA(Service Level Agreement)정보를 바탕으로 동적자원관리를 수행한다. 동적자원관리를 수행하기 위해 트래픽 제어 에이전트는 SLA별로 할당된 대역폭을 저장하는 데이터베이스를 가지고 있으며, 향후 자원할당을 결정하기 위한 기초자료로 사용한다. 클라이언트에서 차별화서비스 네트워크의 경계라우터로 서비스를 요청하면, 경계라우터는 서비스의 유형, 목표 전송률, 최대 버스트 크기, 서비스 요구시간 등의 SLA정보를 기반으로 해당 트래픽 제어 에이전트로 서비스요청 정보를 보내게 된다. 트래픽 제어 에이전트가 서비스요청 정보를 받으면 요구대역폭을 할당할 수 있는지를 검사하여, 가능하면 경계라우터로 승낙 정보를 전송하게 된다.

즉, 트래픽 제어 에이전트는 차별화서비스 네트워크 전체의 자원을 관리하며, 또한 차별화서비스 네트워크에서의 정책에 따라 수락제어를 결정한다. 이를 바탕으로 내부 라우터들과의 통신 그리고 인접 차별화서비스 네트워크 사이의 협상기능을 수행하게 된다. 이러한 기능을 수행하기 위하여 본 논문에서의 트래픽 제어 에이전트는 2 계층(two-tier) 구조를 가지게 된다.

첫 번째 계층은 차별화서비스 도메인 내부의 자원할당을 다루는 도메인 내부(Inter-domain) 자원관리구조로, 도메인 내부에서 트래픽 제어 에이전트와 라우터들 간의 통신을 통해 자원관리를 하게 된다. 두 번째 계층은 두 도메인 간(Inter-domain) 네트워크 경계에서 자원을 공급하고 할당하는 도메인 사이의 자원관리구조이다. 이를 위해 두 도메인 간의 트래픽량과 트래픽유형 등의 정보를 트래픽 제어 에이전트 간 수락제어를 통해 교환한다.

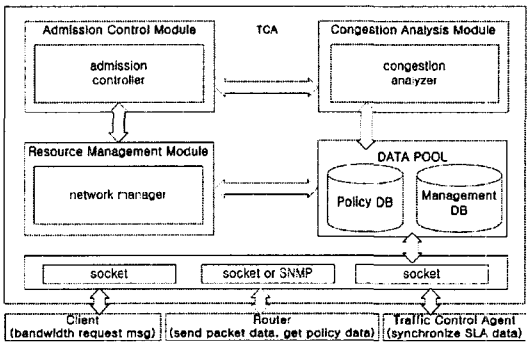

그림 1 흐름 관리를 위한 트래픽 제어 에이전트 구성도

본 논문에서 제시하는 트래픽 제어 에이전트는 그림 1에 보이는 바와 같이 수락 제어 모듈(Admission Control Module), 자원관리 모듈(Rescurce Management Module), 혼잡 분석 모듈(Congestion Analysis Module)로 구성되며 각 모듈들에대한 설명은 세부 절로 구성한다.

2.1 수락제어모듈

수락제어 모듈은 링크 상에 흐름의 예약률 합이 링크의 용량을 초과하지 않도록 새로운 요청의 수락 여부를 결정하는 모듈이다. 그림 2에서와 같이 수락제어 모듈은 정책 데이터베이스를 참조하여 SLA협상을 정적으로 수행할 수도 있으며, 자원관리 모듈을 연계하여 동적으로 수행할 수도 있다.

그림 2에서 초기 정책설정은 네트워크 관리자에 의해 수행되며, 설정된 정책은 명령어 처리기를 통해 정책데이터베이스에 저장된다. 수락제어기는 정책데이터베이스를 참조하여 클라이언트의 요청에 대한 SLA를 수행한 후 서비스 수락 여부를 결정하게 된다. 서비스 수락이 결정되면 그림 3과 같이 트래픽 제어 에이전트는 DiffServ라우터내의 흐름 관리기와의 통신을 통하여 트래픽 제어를 수행한다.

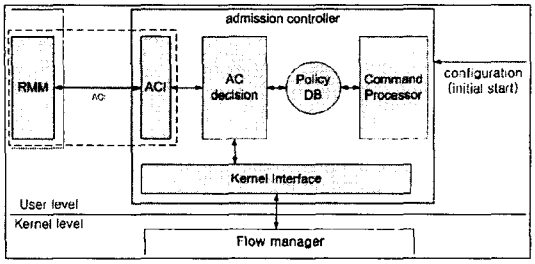

그림 2. 수락제어 모듈

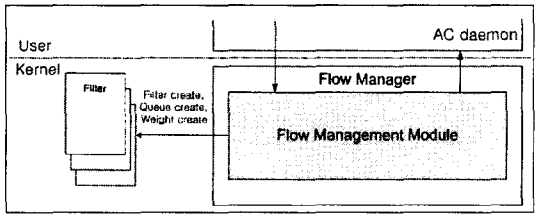

그림 3. 트래픽 제어부 구성도

그림 3에서 트래픽 제어부내의 흐름 관리기는 동적 필터생성 기능, 큐 관리 정책설정, 큐 가중치할당 등의 기능을 수행한다. 동적 필터생성 기능은 경계 라우터의 Ingress 인터페이스 부분에서 u32필터를 생성하여 패킷의 미터링(metering) 기능을 수행하며, Egress 인터페이스 부분에서 tc_index필터를 통해 분류된 트래픽의 DSCP마킹을 수행한다. 예약된 자원을 할당하기 위한 큐 관리 정책과 가중치설정은 경계 라우터와 코어 라우터에 동일하게 적용된다.

2.2 자원관리모듈

자원관리모듈에서는 클라이언트에서 요청한 SLA정보를 처리하기위해 라우터로부터 망 자원에 관한 정보를 가져오는 역할과 수집된 데이터를 기반으로 현재 망에서 서비스가 가능한지를 계산하게 된다. 이때 라우터로부터 수집된 정보는 관리데이터베이스에 저장되고, 서비스 가능여부를 계산한 정보는 정책데이터베이스에 저장된다. 정책데이터베이스에 저장된 정보는 수락제어 모듈에서 클라이언트의 서비스요청 메시지에 대한 수락여부판단에 이용된다. 또한 차별화서비스 도메인내의 네트워크이용률을 계산하여 클라이언트와 트래픽 제어 에이전트 간의 SLA를 위한 정보로 이용된다.

본 논문의 자원관리모듈은 그림 4와같이 패킷획득방법과 SNMP모듈을 사용한다. 패킷획득방법은 차별화서비스 네트워크에서 전송되고 있는 트래픽을 서비스단위로 검사하는데 사용되며, LibPcap라이브러리를 이용하여 구현한다. 자원관리 모듈에서 패킷수집과 데이터베이스저장 부분은 별도의 프로세스로 동작을 하며 프로세스사이의 데이터 공유를 위하여 공유메모리기법을 사용한다.

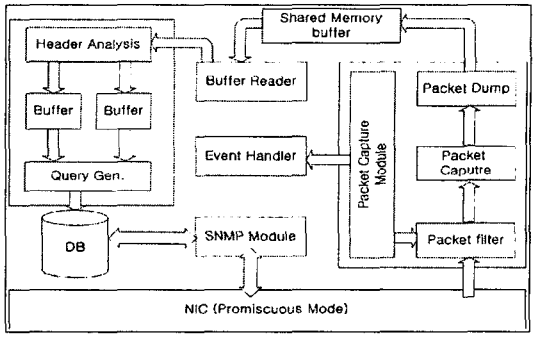

그림 4. 자원관리 모듈의 구성도

그림 4에서 패킷획득방식은 네트워크 디바이스에서 패킷필터링 과정을 거쳐 패킷을 획득하고 일련의 루틴에 따라 헤더를 풀어낸 후 공유메모리버퍼에 저장한다. 이때, 데이터베이스 변환모듈에서 이 데이터를 읽어서 데이터베이스 테이블에 맞는 형태로 일련의 변환과정을 거쳐 일정간격으로 저장한다.

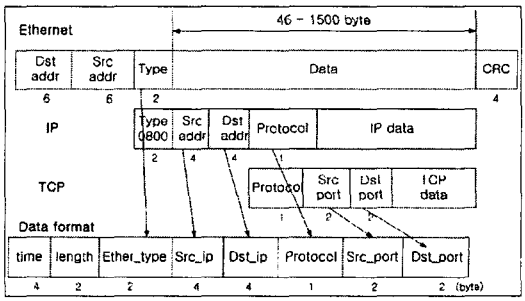

그림 5. 패킷 획득의 데이터 형태

그림 5는 이더넷 프레임에서 정보를 추출하여 데이터베이스에 저장되는 데이터형태를 설명한다. 데이터형태에서 time은 패킷을 수집해서 데이터베이스에 저장할 때의 시간이며, length는 패킷 전체의 크기에 CRC값(4byte)을 더한 값이다. time과 length를 제외한 나머지 정보는 모두 패킷에서 가져오는데 ether_type은 이더넷 헤더에서 정보를 가져오고, src_ip, dst_ip, protocol 등은 IP프로토콜 헤더에서, src_port, dst_port 등은 TCP/UDP헤더에서 정보를 가져온다. 데이터 포맷의 일정 필드를 사용하지 않는 프로토콜은 0으로 채운다.

SNMP모듈은 차별화서비스 네트워크에서 입출력되는 총 패킷 양을 모니터링하고 네트워크 이용률 및 에러율을 계산하는데 사용된다.

2.3 혼잡분석모듈

혼잡분석모듈에서는 자원관리모듈에서 수집된 관리 정

보를 기반으로 패킷의 흐름별 전송률, 흐름별 이용률, 흐름별 에러율 등을 검색하여 인증제어 모듈의 인증절차의 기초 자료로 이용된다. 본 논문에서 제시하는 이용률과 에러율을 분석하는 식에서 사용된 변수를 표 1에 정리한다.

표 1. 식에서 사용된 변수

변 수	설 명
x	폴링 주기에서 이전 폴링시간
t	폴링 주기
ifInOctets	인터페이스에 수신된 옥텟의 총 수
ifOutOctets	인터페이스 외부로 전송되는 옥텟의 총 수
sysUpTime	시스템이 마지막으로 재 초기화된 이후의 시간
ifSpeed	인터페이스의 현재 대역폭, bps로 표시
ifInError	오류 때문에 상위 계층에 전달되지 못한 패킷의 수
totalPktIn	전체 도착 패킷의 수

네트워크 트래픽의 이용률 계산을 위해 혼잡 제어 모듈에서는 각각의 컴포넌트 네트워크에 따라 분석 식을 다르게 사용한다. 본 논문에서는 이더넷 이용률과 에러율 계산식만 제시한다. 모든 라우터의 입·출력 트래픽 량은 식 (1)과같이 송신측에서 전송한 패킷의 총 비트 수와 수신측에서 받은 총 비트 수를 합하여 네트워크 링크의 전체 대역폭으로 나누면 된다. 기본적으로 이더넷은 브로드캐스팅 전송방식을 사용한다. 따라서 이더넷의 이용률을 측정하기 위해서는 모든 입출력 트래픽을 더한 다음, 트래픽의 전체 합을 최대대역폭으로 나눠주면 된다. 식 (2)에서 이더넷 트래픽의 이용률 분석 식을 보인다. 에러율의 경우 전체대역폭에 1%이상이 에러로 판명되면 네트워크 관리자는 네트워크 디바이스를 점검해야만 한다. 식 (3)에서 에러율 분석 식을 보이며 *totalPktIn*은 전체 입력 패킷의 수를 의미한다. 식 (4)와 같이 전체 입력 패킷의 수는 입력 유니캐스트 패킷의 수, 입력 브로드캐스트 패킷의 수 그리고 입력 멀티캐스트 패킷의 수를 합하여 측정할 수 있다.

$$Input/Output\ traffic\ of\ router: \quad (total\_bit\_sent + total\_bit\_received)/bandwidth. \tag{1}$$

$$Ethernet\ Utilization: \quad \frac{(ifInOctets_{(x+t)} - ifInOctets_{(x)} + ifOutOctets_{(x+t)} - ifOutOctets_{(x)}) \times 8}{(sysUpTime_{(x+t)} - sysUpTime_{x}) \times ifSpeed \times 100}. \tag{2}$$

$$Traffic\ error\ rate: \quad \frac{ifInError_{(x+t)} - ifInError_{(x)}}{totalPktIn_{(x+t)} - totalPktIn_{(x)}}. \tag{3}$$

$$Total\ Input\ Packet\ count: \quad totalPktIn = (ifInUcastPkts + ifInBroadcastPkts + ifInMulticastPkts). \tag{4}$$

V. 결론

본 논문에서는 IP네트워크 환경에서 선별적인 QoS를 보장하고 클라이언트의 대역폭 요구에 대한 동적 자원 할당을 수행하는 트래픽 제어 에이전트를 제시하였다. 본 논문에서 제시하는 트래픽 제어 에이전트는 망 노드의 트래픽 정보를 수집하고 분석하여 동적자원할당 기능을 수행하며, 이러한 동적자원할당을 위해 트래픽 제어 에이전트와 클라이언트, 트래픽 제어 에이전트와 라우터 그리고 트래픽 제어 에이전트와 트래픽 제어 에이전트간의 신호체계를 정의는 지면 관계상 생략한다.

참고문헌

[1] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architec- ture: an Overview," *IETF RFC 1633*, June 1994.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," *IETF RFC 2475*, December 1998.

[3] W. Almesberger, "Linux Traffic Control Implementation Overview," *Technical Report, EPFL*, November 1998.

[4] K. Nicholas, S. Blake, F. Barker and D. Blake, "Definition of the Differentiated Services Field(DS Field) in the IPv4 and IPv6 Herders," *IETF RFC 2474*, December 1998.

[5] D. Verma, "Supporting Service Level Agreement in IP Networks," *Macmillan Technical Publishing*, 1999.

[6] B. Evans, "Differentiated Service Implementation," http://www.ittc.ku.edu/~kdrao/BB/.