

DiffServ 도메인에서의 동적 자원 할당에 관한 연구

김진철*, 이영희*

*한국정보통신대학원대학교 정보공학부
e-mail : jckim@icu.ac.kr, yhlee@icu.ac.kr

A Study on the Dynamic Resource Provisioning in the DiffServ Domain

Jin-Cheol Kim*, Young-Hee Lee*

*Dept. of Information Science, Information & Communication University

요 약

DiffServ 네트워크 모델은 기존의 Best Effort 서비스를 지원하는 IP 네트워크에 차별화된 서비스를 지원할 수 있는 IP QoS를 무리없이 도입하기 위하여 제안되었다. DiffServ 네트워크는 IntServ 네트워크와 같은 Scalability 문제를 해소할 수 있도록 Edge-to-Edge QoS를 지원하기 용이한 단순화된 구조를 가지고 있다. 이러한 단순화시킨 네트워크 구조에서 여러 가지 QoS 요구 특성을 가지는 트래픽에게 서비스를 제공하고, 네트워크 자원의 이용 효율을 극대화 하고자 하는 사업자의 요구에 따라 효율적인 자원 할당 방안이 필요하게 된다. 자원 활용의 효율을 제고하기 위해서는 정밀하고 동적인 자원할당 방식을 이용하는 것이 필요하며, 이를 실현하기 위하여 Intra-domain의 Edge-to-Edge Path 상에 있는 노드에서의 트래픽 부하 측정을 바탕으로, DSCP 제어 코드를 사용한 탐색 패킷을 통하여 Intra-domain의 부하 및 자원 할당 수준을 탐색하고, 이를 기반으로 Admission Control을 중심으로 한 Resource Provisioning을 동적으로 수행할 수 있는 메커니즘을 제안한다.

1. 서론

최근 인터넷의 새로운 조류로서 차세대 인터넷 기술에 대한 연구가 활발하게 진행되어 왔다. 이러한 노력의 일환으로 QoS 지원을 위한 메커니즘들이 연구되고 이를 적용한 인터넷의 네트워크 모델들이 제안되었다. 이러한 범주에는 IntServ, DiffServ 등의 개념을 도입한 IP QoS 지원 네트워크 모델들이 연구되어 왔으며, IntServ 모델은 기존 Best Effort 서비스 기반의 IP 모델에서 큰 변화를 추구한 반면, DiffServ 모델은 이를 단순화하여 기존 IP 모델과의 격차를 줄이고 Scalability를 제공함으로써 대규모의 코어망으로서 활용이 가능하게 되었다.

DiffServ에서는 여러 개의 서비스 클래스를 정의하고 이를 지원하기 위한 Boundary 및 Interior 라우터들의 요구 기능, 그리고 Framework 및 각종 서비스 구성 요소들에 대한 표준화 작업을 IETF에서 진행해 왔으며, IntServ 모델과의 연동에 대해서도 연구가 이루어지고 있다.

IETF의 DiffServ WG에서는 서비스 차별화를 위해, 기존의 Best Effort 서비스 클래스 외에 파라미터 측정치로 검증할 수 있도록 구체적인 보장으로 표현되는 Quantitative 트래픽을 위해 Expedited Forwarding(EF) 서비스 클래스를, 상대적 비교우위로 표현되는 특성을 가지는 Qualitative 트래픽을 위해 Assured Forwarding(AF) 서비스 클래스를 정의하고 있다[1]. 아울러 이러한 패킷 처리 클래스를 서비스 모델에 연결하는 PHB(Per Hop Behavior)를 DSCP(DS Codepoint)에 맵핑시키는 방안을 제시하고 있다. PHB를 기술적으로 살펴보면, 각 Hop에서의 Classification, Dropping, Queueing, Scheduling, 그리고 Forwarding 등의 기능들을 조합한 추상적 수준에서의 요구 Behavior에 대한 기술이다[2].

이러한 PHB에 따라 차별화된 서비스를 구현하여 제공하기 위해서는 여러 단계의 작업이 필요하다. 서비스 Provisioning은 이용자(Customer) 서비스 사업자 사이의 계약(SLA: Service Level Agreement)을 기반으로, Admission Control을 중심으로 한 Resource Provisioning

과 이를 실현시키기 위한 Configuration 을 통해 구현 될 수 있다. 여기서 Resource Provisioning 은 Admission Control 과 Configuration 을 포함하는 용어로 기술된다.

패킷 네트워크에서는 Connection-oriented 네트워크의 경우와 달리 본질적으로 트래픽의 예측이 어려우며, 따라서 네트워크의 Resource Provisioning 에도 상대적으로 어려운 문제점이 존재한다[2]. 사업자의 요구는 네트워크의 자원을 최적으로 활용하여 통계적 Multiplexing 이득을 추구하는 반면, 이용자는 네트워크를 최대한 이용하려는 요구를 가진다.

DiffServ 에서 구분하는 기본적인 세 가지 등급의 트래픽은 그 특성에 따라 트래픽 Profile 의 표현 양식도 다르다. DiffServ 트래픽은 Best Effort 트래픽 위에 Quantitative 트래픽과 Qualitative 트래픽이 우선되며, Quantitative 트래픽은 주로 SLA/SLS(Service Level Specification)에 준한 자원 예약을 통해 서비스를 지원 받게 된다. SLS 는 이용자의 동적인 변경도 가능하지만 기본적으로는 Static 한 특성을 가진다. SLS 가 Static 한 특성을 가지는 Quantitative 트래픽의 경우에는 트래픽의 예측이 비교적 용이하므로 Static provisioning 으로도 충분히 지원 가능하다. 그러나 Qualitative 트래픽의 경우에는 SLS 가 static 하여도 트래픽 패턴이 서로 다른 노드에서 동적으로 변하므로 예측이 어렵다. 게다가, Dynamic Provisioning 은 동적으로 변하는 SLS 를 지원하고, Quantitative 서비스에 대해서도 보다 자유로운 Provisioning 으로 통계적 이득을 얻을 수 있게 한다. 따라서, 네트워크 자원을 보다 효율적으로 활용하기 위해서는 Qualitative 트래픽에 대해서는 Dynamic Resource provisioning 방식이 필요하게 된다[1].

본 연구는 IP QoS 를 지원하고자 하는 DiffServ 도메인에서 트래픽 측정을 기반으로 하는 Admission Control 을 동적으로 지원할 수 있는 Resource Provisioning 메커니즘을 제안하고, 이의 특징 및 동작을 분석한다.

2. 관련 연구 동향

Diffserv 도메인에서의 자원 할당 방식은 기본적으로 세 가지 형태로 구분될 수 있다. 집중형과 분산형, 그리고 이 두 가지를 혼합한 방식으로 구분할 수 있으며, 집중형은 Bandwidth Broker(BB)와 같이 하나의 중앙 서버가 도메인 전체의 자원 할당을 관리하는 구조이다. 이에 비해 분산형은 Edge 노드들이 자원 할당을 관장하는 구조이고, 혼합형은 두 가지 방식을 결합한 구조로 정의할 수 있다.

IETF 의 RAP WG 에서는 IntServ/RSVP 네트워크 모델을 위한 자원 할당 프로토콜을 연구하고 있으며, 클라이언트/서버 구조에 기반을 둔 COPS(Common Open Policy Service) 프로토콜을 제안하고 있다[3]. COPS 는 Policy 기반의 Admission Control 을 위해 Policy 서버와 PEP(Policy Enforcement Point)들 사이의 REQ/DEC 메시지를 정의하고, outsourcing 모델과 provisioning 모델을 지원할 수 있다. Local PDP(LPDP)를 PEP 에 내장하는 경우도 있으나 부분적이며 분산 DB 를 지원해야 되는 부담이 있다. DiffServ 를 위한 COPS 의 outsourcing 모델도 제안되고 있다.

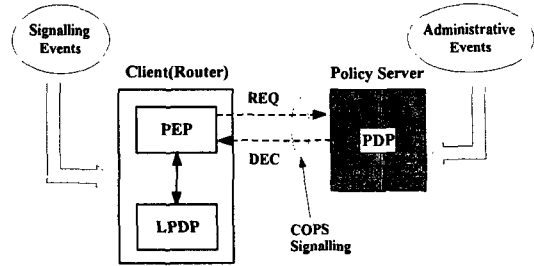


그림 1. COPS 를 이용한 Policy 기반의 모델

이러한 집중형은 관리가 용이하나 도메인 전체의 토폴로지와 자원 할당 Status 를 유지, 업그레이드해야 하고 Signalling 프로토콜을 지원해야 되는 부담이 있으며, 고장시 도메인의 전반적인 문제를 야기할 수 있다. 또한 Policy 서버와의 연결 링크 토폴로지 형상이 트래픽 라우트와 분리하여 연결하는 경우와 중첩하여 연결하는 경우에 따라 링크의 Over-provisioning 이나 링크 bottleneck 이 생길 수 있다. 집중형의 또 다른 문제는 Scalability 로서 서버가 대규모의 네트워크 도메인을 관장하는 것은 어느 정도 트래픽의 예측이 가능하여야 실효성이 있으므로, 자원의 활용도를 높이기 어렵고 동적인 Admission Control 을 지원하기가 어렵다. 이에 따라 이 방식은 트래픽 예측이 비교적 용이한 Quantitative 서비스에 적용할 수 있다.

분산형은 자원 활용의 효율성을 높이고 동적인 Admission Control 을 지원할 수 있으나, 구현이 어렵고 각 노드에 제어 오버헤드가 부가되는 단점이 있다. 그러나 트래픽의 변동성을 예측하기 어려운 Qualitative 서비스에 적용하기 좋은 방식이다. 분산형 가운데 DSCP 필드중 2 비트의 CU 필드를 사용한 부하 제어 방식[4]은 구현이 단순하고 부하가 적은 장점이 있는 반면, 기존 Explicit Congestion Notification 과의 호환성 문제와 최대 Flow 추정의 어려움, 그리고 Marking 유니트의 유연한 설정이 곤란하여 제어 대상 용량의 표현이나 검출을 위해 많은 수의 패킷을 카운트해야 하는 오버헤드를 가지고 있다.

본 논문에서는 DiffServ 도메인의 동적인 Resource provisioning 을 위해 Edge-to-Edge Path 상에 있는 각 노드의 트래픽 측정치를 기반으로, 트래픽 루트의 가용 자원을 탐색하고 이 결과를 가지고 Edge 노드에서 Admission Control 을 결정하는 메커니즘을 제안한다. 제안 메커니즘은 Intra-domain 을 대상으로 Edge 노드에 분산된 Admission Control 기능을 갖추고 실제 트래픽의 상황에 맞추어 동적이며 Fine-grained 수준의 자원 할당 기능을 수행한다. 이와 같이 Intra-domain 의 Edge 노드들이 정밀하고 실시간적인 상태 및 제어 정보를 탐색하고 유지하게 되면, Inter-domain 간의 자원 이용 정보 교환에도 좋은 기반을 제공할 수 있으며, 전체적인 성능 향상에 기여할 수 있는 메커니즘이 된다. 본 논문에서는 Inter-domain 간의 자원 제어 문제는 다루지 않는다.

3. Dynamic Resource Provisioning 메커니즘

AF(Assured Forwarding) 클래스와 같은 Bursty 특성을 가지는 트래픽의 Admission은 사업자가 Statistical multiplexing 이득을 통해 자원 활용을 극대화하기가 용이하지 않다. 본 논문의 제안 메커니즘은 대규모의 Transit 트래픽 Flow를 수용할 수 있는 네트워크에서 동적 특성을 예측하기 어려운 서비스 클래스를 대상으로 다양한 기능과 정밀한 자원 할당 및 제어 기능을 지원하기 위해 설계되었다. 각 링크의 Capacity는 항상 노드의 처리 용량보다 큰 것으로 가정한다.

제안 메커니즘은 기존의 PHB(Per Hop Behavior)를 맵핑하기 위해 정의된 DSCP 코드가 Data Plane 용도의 서비스 코드에 한정된 것을 확장하여, Control 및 Management Plane을 위한 DSCP 코드를 정의하고 이를 Intra-domain 내부에서 이용한다. 이를 사용함으로써 가용 자원의 표현을 정밀하게 할 수 있고, 실시간적인 트래픽 수준을 감지하여 적절한 Admission Control을 처리할 수 있다. 아래 그림의 DS 필드에서 Control 및 Management Plane을 위한 DSCP 코드는 제어 Class로 구분하고 EF 클래스와 같은 등급을 부여한다.

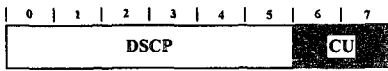


그림 2. DiffServ DS Field

이러한 제어용 DSCP 코드는 다음과 같은 3가지 Pool 가운데 Pool 2의 EXP/Local 코드를 사용하여 동적 자원 제어 메커니즘을 설계한다.

- ◊ Pool 1 : 'xxxx0' : IANA/Std
- ◊ Pool 2 : 'xxxx11' : Exp/Local
- ◊ Pool 3 : 'xxxx01' : Exp/Local/IANA

새로운 제어 코드의 제안 목적은 DiffServ 네트워크 모델의 Edge-to-Edge 특성을 활용하여 Resource Probing을 수행하기 위한 것으로서, Ingress 노드에서 제어 코드 패킷을 사용하여 원하는 대상 트래픽의 라우트를 Egress 노드까지 탐색하면, 그 라우트에 있는 노드들은 실시간적으로 업데이트하는 평균 트래픽 수준을 바탕으로 해당 클래스의 가용 자원을 DSCP 필드에 인코딩한다. 그 결과는 Egress 노드가 Ingress 노드에 응답하여 Admission Control의 근거 정보로 사용한다. 또한 Per-Flow 처리를 하지 않는 DiffServ 도메인에서의 내부 노드의 특성상 Aggregated Flow를 제어하기 위해서는 모든 노드들이 검사하는 DS 필드를 활용하는 것이 오버헤드를 줄일 수 있는 효과적인 방안이다. Resource Probing을 위한 방법에는 Signalling 방식에서부터 Mobile Agent를 이용하는 방식에 이르기까지 여러 가지 방법이 있지만 오버헤드가 크거나 처리 속도가 느린 단점들이 있다.

제어 패킷의 생성은 Event-driven 방식으로 Ingress 노드에서 생성하고, 내부 노드는 DSCP Encoding을, Egress 노드는 응답 패킷의 생성 및 전송을 처리한다. 본 메커니즘은 각 노드에서의 실제 트래픽 측정을 전제로 동작하며, Probing하는 내부 라우트는 데이터 트래픽의 라우트와 동일하다고 가정한다. 이러한 본 메커니즘의 주요 기능 및 동작 시나리오는 다음과 같다.

1) 트래픽 측정 및 단위

도메인 내부 라우트의 트래픽 측정은 내부 노드에

서 RED(Random Early Detection) 계열의 방식과 같은 기존 메커니즘을 활용하며, 각 노드는 각 트래픽 클래스에서의 평균 Queue 점유 수준을 실시간으로 업데이트한다. 이 정보는 제어 패킷이 수신되었을 때 DSCP에 Encoding할 정보가 되며, 자원 인코딩 단위는 트래픽 상황에 맞추어 선택적으로 Configuration이 가능하도록 몇 가지 체계를 동작 파라미터로 지정하여 사용한다. 예로서, 전체 Capacity에서 RED Threshold를 뺀 부분을 1단위로 사용하거나, 전체에 대한 퍼센트를 단위로 하여 표시한 체계를 사용할 수도 있다. 단위 체계를 Configuration할 수 있는 파라미터로 두면, 동적인 트래픽 수준을 다양하게 반영할 수 있는 체계를 개발할 수 있는 여지가 있으며, 보다 정밀한 제어에 활용할 수 있다.

2) Resource Reservation & Refreshment

Quantitative Traffic과 같은 경우 트래픽의 예측이 어느 정도 가능하고 가장 우선 순위의 서비스 수준을 지원하여야 하므로, Reservation에 의한 서비스를 제공하는 것이 일반적이다. 이 경우 새로운 예약 이벤트가 발생하면, Ingress 노드는 기존에 파악된 정보를 바탕으로 예약 패킷을 전송한다. 예약 용량은 Aggregated Flow에 대한 기본 단위 이상의 증감분이 발생했을 때 하나 이상의 제어 패킷에 기본 단위의 배수로 DSCP에 인코딩하여 전송하고, 내부 노드는 이를 보고 허용 가능하면 필요한 동작을 수행하고 허용 불가이면 CU 필드에 마킹을 한다. Egress 노드는 예약 수행 결과를 Ingress 노드에게 응답 패킷으로 보낸다. 예약 관련 코드는 '0xxx11'을 사용하며, 다음과 같이 증감을 구분하여 'xx'에 기본 단위의 배수로 인코딩한다.

- '01xx11' : Increment Reservation
- '00xx11' : Decrement Reservation

Reservation Refreshment는 예약 자원의 Aggregated Flow를 대상으로 Soft State 방식으로 처리한다. 주기적인 Refreshment를 위한 코드는 '000011'을 사용하고, 처리 대상의 용량 표시는 패킷의 Payload에 기본 단위의 배수로 표시한다. Refreshment 패킷의 주기는 RSVP의 주기 이하로 하며, 주기적 특성을 활용하기 위하여 2-비트 CU 필드에는 AF 클래스의 최소 가용 자원을 마킹하도록 하여 백그라운드 Resource Probing 효과를 얻도록 한다.



그림 3. Resource Probing Concept

3) Resource Probing

AF 클래스의 Qualitative Traffic과 같이 동적 특성이 큰 경우에는, Admission Control을 위한 이벤트 발생시 백그라운드 Resource Probing으로 얻은 자원 사용 정보가 용량이나 타이밍이 부적절할 경우, Edge-to-Edge Path 상의 가용 자원 정보를 얻기 위하여 그림 3의 개념과 같이 Explicit Resource Probing 패킷을 전송한다. 이 경우 제어 패킷의 Payload에는 허용 대상의 자원 용량을 탑재하여 도메인의 내부 노드들이 확인할 수 있도록 한다.

이 기능을 위한 DSCP 제어 코드는 '1xxx11'를 사용하며, 각 노드는 'xxx'에 자신의 가용자원을 기본 단위의 배수로 표시하되, 자신의 용량이 적은 경우에만 표시한다. Egress 노드는 이 결과를 Encapsulation 한 응답 패킷을 생성하여 Ingress 노드로 전송하며, Reverse Path Probing 도 가능하도록 한다.

4) Congestion Notification

Explicit Congestion Notification 기능을 이용하기 위하여 ECT(ECN-Capable Transport)가 데이터 패킷에 마킹되어 있으면 end-to-end 간의 Congestion Notification 기능을 지원하며, 내부 노드는 RED 기능에서 패킷 Discard 를 할 경우라도 그 개수만큼의 데이터 패킷에 Congestion Indication 을 마킹하도록 한다. Egress 노드에서는 Congestion Indication 마킹 현상을 검출하면 이를 Ingress 노드로 알리고, 해당 트래픽 클래스의 자원 점유 정보로 이용한다. 즉, Congestion Indication 이 표시된 클래스의 Resource Probing 에 대해서는 Egress 노드가 일정 시간 동안 가용 자원의 존재를 거부하는 응답을 할 수 있다.

5) IntServ/RSVP 와의 연동 지원

IntServ/RSVP 와 DiffServ 와의 연동 구조에는 여러 가지 시나리오들이 있으나 가장 일반적인 구조는 DiffServ 망을 코어로 사용하고 IntServ/RSVP 를 end-to-end 서비스로 이용하는 구조이다. 이 경우, RSVP 예약 서비스 트래픽은 DiffServ 의 EF 클래스로 통합하여 처리하는 구조를 설정한다. 이는 약간의 차이를 가지고 있지만 비슷한 트래픽 특성을 가진 두 종류의 트래픽을 위해 서로 다른 M&C 시스템 기능을 지원하는 것은 오버헤드가 크며, 또한 통계적 Multiplexing 에서 추구하는 이득을 위해 하나의 PHB 로 통합하여 지원하는 것이 유리하기 때문이다[5].

RSVP 의 Path 메시지와 Resv 메시지를 DiffServ 도메인에서 처리하는 과정은 다음과 같은 단계로 진행되며, Edge Node 에서는 RSVP 지원 기능을 가지고 Admission Control 과 Flow Aggregation 을 수행한다.

- Sender 가 Path 메시지를 전송하면, DiffServ 내부 도메인에서는 이를 transparent 하게 전달
- Receiver 는 Resv 메시지 전송
- DiffServ Edge 는 Resv 메시지에 대해 Admission Control 을 수행
- Aggregation 의 기본 예약 변동 단위보다 작거나 내부 자원예약 절차로 Capacity 가 허용되면, Resv 메시지를 Sender 에게 전송

여기서 모든 Resv 메시지에 대해 Admission Control 을 수행할 것인가 하는 문제에 대해서는, RSVP 의 주기와 내부 도메인의 Refreshment 주기의 비율을 고려한 Flow Aggregation 을 통하여 처리한다.

6) Intra-domain 의 Round-Trip Delay 정보 수집

트래픽 프로파일이 Delay Metric 을 주요 파라미터로 설정되는 서비스를 지원하기 위해서는 도메인의 Round-Trip Delay 정보를 수집하여야 하며, 이 기능은 본 메커니즘의 제어 패킷을 이용하면 용이하게 Edge-to-Edge 사이의 Round-Trip Delay 를 추적할 수 있다.

7) 내부 Node 의 DSCP Encoding Procedure

Edge 노드는 Event-driven 방식으로 제어 패킷을 생성하여 전송하고, Interior 노드는 각 서비스 클래스의 해당 Control 패킷이 오면 다음과 같은 Procedure 를 적용하여 가용 자원 용량을 엔코딩한다.

```

If ecn is detected, then
    update corresponding resource status
if dscp is control code('xxxx11'), then
    if dscp is Reservation, then
        increment or decrement reservation
    else if dscp is Refreshment, then
        refresh soft state for aggregated amount in payload
        encode AF_resrc_min in the CU field
    else if dscp is Probing, then
        if my_af_resrc is smaller than pkt_resrc, then
            encode my_resrc in dscp field
        else forward pkt
    else forward pkt
else do other Per Hop Behavior
    
```

4. 결론 및 향후 계획

DiffServ 도메인에서의 동적인 Resource Provisioning 을 위해 제안한 메커니즘의 개념과 동작에 대해 기술하였다. 가능한 한 헤더의 DS Field 를 이용하여 제어 정보를 엔코딩하면, 모든 노드는 DS field 를 검사하므로 간단한 기능은 Header 만의 처리로 끝나며, 패킷 Payload 를 이용하는 Signalling 방식은 부득이한 경우에만 적용하도록 사용을 최소화하였다.

제안한 메커니즘을 이용하여 분산된 Edge 노드에서 활용하면, 적어도 Local Domain 내에서는 적은 오버헤드로 정밀하고 (Fine-grained) 동적인 Intra-domain 의 Resource Provisioning 이 가능하다. 또한 이러한 Intra-domain 의 정확한 Resource 상태 정보를 바탕으로 하여, Inter-domain 간의 Policy 서버와 같은 Central 서버에게 정밀한 도메인 상태 정보를 실시간적으로 제공할 수 있으므로 시너지 효과를 기대할 수 있다.

Deployment 관점에서 보면, 초기에는 Configuration 에 의한 PHB deployment 와 유사한 작업이 필요하나, Programmable 네트워크의 도입을 위한 활동이 활발해지고 있는 상황에서, 새로운 PHB 를 네트워크에 도입하는 것보다 적은 수준의 오버헤드가 필요하게 될 것으로 예상된다.

본 논문은 제안 메커니즘의 Framework 에 중점을 두고 설계되었으며 구체적인 구현이나 실험을 거친 결과가 아직 없으나, 제안 방식의 성능 파라미터를 검증하기 위한 시뮬레이션 과정을 진행하여 성능 결과를 구체적으로 확인할 예정이다.

참고문헌

- [1] Y. Bernet, et al. "A Framework for Differentiated Services", Internet Draft, draft-ietf-diffserv-framework-02.txt, February, 1999.
- [2] Grenville Armitage, "Quality of Service in IP Networks : foundations for a multi-service Internet", MTP, April 2000.
- [3] Stefano Salsano, et al. "COPS Usage for Outsourcing DiffServ Resource Allocation", Internet Draft, draft-salsano-issll-cops-odra-00.txt, February 22, 2000.
- [4] L. Westberg, "Load Control of Real-Time Traffic - A 2-bit Resource Allocation Scheme", Internet Draft, Oct. 23, 1999.
- [5] Kalevi Kilikki, "Differentiated Services for the Internet", MTP, 1999.