

# NGN에서의 QoS 제공 기술

상명대학교 정진우, 한국전자통신연구원 송종태, 이순석

차 례

I. 개요

II. 기존 단대단 QoS 보장방안

III. ITU-T SG13에서 제안된 QoS 보장방안

IV. 결론 및 보완책

## I. 개 요

BcN(Broadband Convergence Network, 광대역통합망) 등의 차세대 네트워크에서의 QoS 보장은 중요한 이슈이다. ITU-T에서는 Y.1540/1541 표준에서 QoS와 성능관련 지표, 성능측정 방법과, 품질관리 표준모델을 제시하고 있으며, FGNGN (현재 NGN-GSI) WG3를 통하여 QoS 관련 여러 가지 문제들을 해결해 나가고 있다. IETF에서는 IP 성능 Metrics (IPPM) WG에서 성능관련 지표, 성능측정 방법을 제시하고, RFC3270 (MPLS support of DiffServ), RFC 3564 (Requirements for Support of DiffServ-aware MPLS TE) 등을 통해 차세대 인터넷의 QoS 기반구조를 제시하고 있다. 양 진영 모두 DiffServ와 MPLS를 QoS 제공방안의 시작점으로 보고 있으며 이들의 적절한 융합과 발전을 통해 진정한 의미의 QoS 제공이 가능할 것으로 보고 있다.

한국형 광대역통합망을 구축하는 문제에 있어서는 두 가지 사항이 동시에 고려되어야 한다. 첫 번째는 한국 고유기술을 바탕으로 한국 고유 실정에 맞게 전체 구조와 세부 사항을 정하는 것이다. IT 강국이라 자부하는 한국에서 실제 기간망은 외산 제품들로만 혹은 외국에서 전적으로 결정된 세부 사항에 맞추어서 설계 도입되고 있는 실정이다. 이러한 현실을 차세대 통신망에서는 필히 극복해야 한다. ETRI에서 주도하여 진행하는 BcN 구축사업은 이런 점에서 의미가 크다. 광대역통합망 구축을 위한 또 다른 고려사항은 전세계적인 표준에 부합하여야 한다는 것이다. 이를 위해 국내 연구기관들이 BcN을 바탕으로 ITU-T 등의 표준 작업에 적극 동참하고 있으며 이를 통해 국가 기간망 사업이 본 궤도에 오를 수 있을 것이다. 이러한 BcN 사업에서 가장 중요하게 고려되고 있는 부분이 QoS와 보안, 그리고 이동성 지원의 문제이다. 본 연구에서는 QoS 보장 기술의 기본 시작점을 제공하는 DiffServ와 이의 변형된 아이디어

들을 분석하여 여러 가지 국내 사정과 세계적인 기술 발전 동향과 부합하는 솔루션을 제시함으로써 BcN 정착에 일익을 담당하고자 한다. 이러한 연구에서 해결하고자 하는 문제는 현재 인터넷이 가지고 있는 문제가 아직 차세대 통신망에서도 해결되지 않고 있는 부분이며 이에 따라 이러한 연구를 통하여 기술개발이 성공한다면 이것이 내포하는 시장에서의 충격은 무척 크다고 할 것이다.

본 논문에서는 QoS architecture 중 data-plane, 혹은 user-plane이라 일컫는 부분에 집중하여 관련 기존의 많은 연구 내용을 종합하여 NGN에서 필요하다고 여겨지는 최적의 솔루션들을 종합해 보고 ITU-T SG13에서 최근 제안되고 있는 단대단 QoS 보장방안, 특히 Y.flowreq를 분석하는 것을 연구의 범위로 정하였다. 이를 위해 기존 연구들의 분석이 이루어졌다. 이들은 크게 1) IntServ의 stateful 특성을 DiffServ에 최대한 반영하여 이들의 hybrid 솔루션을 찾음으로써 scalability를 유지하는 연구 접근방법, 2) IntServ의 틀을 유지하되 제어하는 대상이 되는 플로우를 통합하여 복잡도를 낮춤으로써 scalability를 유지하는 방법 3) Strict-guarantee가 아닌 class간 relative QoS를 정확하게 보장해주는 방법, 4) Admission control등의 시그널링을 최소화하여 네트워크의 scalability를 유지하는 방법 등을 연구 분석하였고, 기존 Y.flowreq로 대변되는 ITU-T SG13의 QoS 보장방안을 분석하고 문제점을 찾았으며 이에 따른 보완책을 간단히 언급하였다.

## II. 기존 단대단 QoS 보장방안

### 2.1. Differentiated Service

지난 10 여 년간 Internet에서 단대단 QoS

(Quality of Service)를 보장해 주는 방안에 대한 연구와 관련 제품개발에 쏟아 부어진 노력은 막대하다. Integrated Services(IntServ)는 최초의 통합 QoS architecture라는 면에서 의의가 크지만 scalability의 결여로 실제로 구현된 예가 없으므로 그 설명은 생략하기로 한다. Differentiated Services (DiffServ) 구조[1]는 기존의 IntServ와 비교하여 보았을 때, Internet의 규모에 걸맞은 확장성을 염두에 두고 고안된 네트워크 구조이다. 이 구조의 핵심 아이디어는 각 플로우에 대한 상태 (State) 정보가 없는 네트워크의 코어 노드와 상태 정보를 유지, 처리하는 주변 (edge) 노드를 가지고서 여러 개의 같은 성질의 플로우들이 속한 "class" 간의 QoS 차이를 제공하는 것이다. 이러한 DiffServ 구조에서는 Premium 서비스 (Virtual Leased Line, VLL) [2]과 Assured 서비스 [3] 등의 서비스 레벨을 정의하고 있다. Premium 서비스를 제공 받기 위해서 종단의 사용자는 미리 약속된 플로우 특성 (Traffic Conditioning Specification, TCS)에 부합하는 만큼의 트래픽만을 망으로 내보내야 하며 망은 주변 노드에서 이를 monitoring/policing 하게 되며 코어에서는 할당된 만큼의 대역폭을 제공함으로써 지연시간 최대치 등의 성능을 보장해 준다.

### 2.2 기타 hybrid 방식의 QoS 보장방안

위에서 언급한 바와 같이 IntServ는 확장성의 문제로 전혀 구현되고 있지 않으며 실제로는 DiffServ만이 구현 대상으로 고려되고 있는 실정이다. 하지만 최근 연구결과에 따르면 DiffServ에 필연적으로 수반되는 플로우 통합으로 인해 premium 서비스가, 상당히 under-loaded된 네트워크에서만 예측 가능한 정도의 지연시간 최대치를 제공해주며 [4], assured 서비스도 실제 구현에 있어서는 상당히 복잡하다

[5]. 특히 assured 서비스를 TCP 플로우에 적용시키는 경우 (i) 실제 할당된 대역폭이 assured rate에 비례하지 않으며, (ii) assured rate을 항상 제공해 줄 수 없다는 것이 밝혀 졌다[6].

DiffServ에 기반한 QoS 보장방법은 한 편으로는 널리 사용되고 있지만, 다른 한 편으로는 계속 진화해 나가고 있는 중이다. 중요한 발전을 가능케 하는 개념들이 학계를 중심으로 속속 발표되고 있으며 망 기계조업체와 망 사업자 등의 산업계는 이를 적극적으로 받아들이려고 노력하는 중이다. 특히 NGN에서는 이러한 DiffServ 기반의 QoS 제공방안을 전체 망 구현을 위한 가장 중요한 요소 중 하나로 보고 있다.

기존 DiffServ는 다음과 같은 문제점을 가지고 있다:

- 실시간 트래픽을 지원하기 위해서 premium 서비스를 사용해야 하며, 이를 위한 필요조건은 자원예약, 트래픽 조절, 대역폭을 보장해 주는 클래스-레벨 스케줄링 기법 (Weighted fair queueing 류) 등의 사용이다.
- 하지만 이들은 충분 조건은 되지 못한다. 즉, 네트워크 utilization이 상당히 낮은 수준이어야 하는데, 이는 조절하기가 무척 어렵거나, 혹은 망 규모가 어느 정도 이상이면 불가능하다.

따라서 이를 개선하고자 하는 노력들이 고안되었는데 이들을 크게 종류별로 나누어 보면 다음과 같다.

1. 플로우 별 상태 정보 없이 플로우 별 대역폭을 보장하는 망 구조: SCORE[7], FADE[17], LBFA [13], 그리고 그 밖의 수 많은 논문들이 기존 IntServ에서처럼 DiffServ에서도 단대단 QoS 보장이 가능하도록 여러 가지 방법들을 제시하고 있다.
2. 단대단 QoS를 포기하고 DiffServ를 더 간단하

게 만드는 구조 (RDS[14]): 기존 DiffServ의 assured 서비스와 유사한 면이 있지만 트래픽 조절, 자원 예약 등이 필요 없어진다.

1의 문제점은 결국 플로우 상태 유지에 필요한 만큼은 아니지만 망의 복잡도가 높아진다는 것이고, 2의 문제점은 (당연하게도) 단대단 QoS 보장이 불가능하다는 것이다.

이에 따라, 상태 정보를 유지하지 않는다는 DiffServ의 원칙을 유지하면서 단대단 QoS를 보장해 주는 방안들이 모색되었다 [7] [8] [9] [10] [11]. SCED+ 구조 [10]는 플로우 통합을 거치면서 guaranteed rate과 statistical rate을 보장해 주는 알고리즘을 제시하였다. 잘 알려진 다른 망 구조로는 SCORE (Scalable Core) 구조[7]를 들 수 있다. SCORE는 기존 IntServ 기반의 “기존 네트워크”를 상정해서 이를 최대한 emulate하는 것을 목표로 하고 있는데, 기존 네트워크에서는 Virtual clock 스케줄러를 사용하는 것을 상정하였고, 이를 emulate하기 위해 여러 가지 VC-스케줄러의 변형 본들을 고려하였다. SCORE에서 제시한 중요한 개선 아이디어는 상태를 패킷의 헤더에 저장해서 전달하고 이를 각 노드에서 분석, 활용한 후 수정한다는 것이며, 이 방법으로 네트워크-레벨 상태가 유지된다는 것이다. 구체적으로 설명하면, 패킷에 기록된 지연시간 기한 정보를 가지고 각 노드는 각 패킷에 대하여 QoS 보장을 해주게 되며 결국 모든 노드가 이렇게 함으로써 단대단 지연시간이 보장되는 방식이다. 하지만 이러한 망 구조도 마찬가지로 실제 구현은 상당히 복잡하며, 더군다나 여기서 제안하는 Core-stateless Jitter Virtual Clock (CJVC)은 non-work conserving 특성을 가지고 있어 statistical multiplexing gain을 충분히 활용하지 못하여 필요 이상으로 네트워크 부

하를 증가시킨다는 단점이 있다.

한편, [4]에서도 명시된 바와 같이 네트워크 utilization이 충분히 작다는 조건만 만족된다면 상대적으로 간단한 스케줄링 알고리즘 만으로 최대 단대단 지연시간을 보장할 수 있으며 [12]에서는 이러한 static-priority 기반의 스케줄러를 사용하는 망 구조가 제안되었다. 링크 레벨의 통합을 통해서 단대단 QoS 보장을 도모한 사례도 있다. IntServ가 플로우 레벨, DiffServ는 클래스 레벨, SCORE가 패킷 레벨의 제어를 각각 필요로 한다면, LBFA (Link-based fair aggregation) [13]은 각 노드의 출력 링크에서 클래스 별로 re-shaping을 함으로써 링크 레벨의 제어만으로 (플로우 별 정보를 유지하지 않은 채) 플로우의 대역폭 제공을 보장한다.

전혀 다른 접근방법으로는 Relative Differentiated Services (RDS) 구조를 들 수 있다[14] [15]. 이 구조에서는 DiffServ와는 달리 각각의 트래픽이 N개의 general 서비스 클래스에 속하며 지연시간 혹은 손실률 등, 클래스 간의 상대적인 성능 지표만이 유지되며 망의 각 노드는 지역적인 정보만으로 클래스를 처리한다.

이렇게 함으로써 사용자는 기존 DiffServ에서 약속 받았던 절대적인 QoS 지표 대신 상대적인 서비스 차이만을 제공 받고, 따라서 어떤 클래스를 선택할 것인가 하는 문제는 사용자가 네트워크 상황에 맞게 선택할 수 있도록 유연하게 대처하는 네트워크를 구성할 수 있게 된다[15]. 구체적으로 설명하면, RDS는 클래스간의 평균 지연시간 비율 혹은 손실률을 일정하게 유지해 주는 기존 네트워크 모델 (지연시간의 경우 PDD 모델, Proportional Delay Differentiation 모델)을 생각하여 이를 최대한 emulate하는 스케줄러들을 고안한다는 것이다. RDS에서는 따라서 기존 DS premium 서비스를 위해 필요했던 자원 예약, 트래픽 monitoring/policing 등이 필요 없게 된

다. 하지만 당연히, 단대단 QoS는 보장하지 못한다. 이러한 RDS에 대비하여 기존의 DiffServ를 Absolute DS (ADS)로 칭하기도 한다.

DiffServ에 대한 또 다른 개선은 코어 노드와 주변 노드 간의 control plane에서의 동적인 정보교환으로 가능하다는 주장도 제기되고 있다 [16] [17] [18]. [16]은 기존 DiffServ 구조에 간단한 피드백 메커니즘을 추가하는 것을 제안하고 있으며 FADE (Fair Allocation Derivative Estimation) [16]는 이를 기반으로 각 노드에서 플로우 별 상태 정보 없이 대역폭을 보장하는 방법을 제시하였다.

이 방법은 SCORE[7]와 비교하면, 플로우들의 상태정보 없이 각 플로우의 대역폭을 보장한다는 면에서 유사하다. SCORE가 data-plane을 이용해서 (패킷 내부에 정보를 기입하는 식으로) 노드간 정보를 교환한다면, FADE는 control-plane을 통해 명백히 정보 교환을 한다고 볼 수 있다.

FADE에서는 하지만 이를 위해 전체 네트워크 자원이 일정 수준 이상이 되는지 복잡하게 미리 계산해 놓아야 하는 admission control이 필요하며, 노드의 링크에서 클래스 별 arrival rate을 계속 측정해야 하는 단점이 있다. Active Resource Management (ARM) [18]은 한 걸음 더 나아가 할당되는 대역폭을 동적으로 조절해야 한다고 주장한다.

결론적으로 DiffServ의 premium 서비스 제공 체계를 개선하기 위해서는 1의 방법들을, assured 서비스제공 체계를 개선하기 위해서는 2의 방법들이 점차 채택되어 나갈 전망이다. 각 제안들의 우열은 적용될 네트워크의 상황에 크게 좌우될 것이므로 어떤 제안이 향후 우세할 것인가는 미리 짐치기 무척 어렵다. ITU-T에서 추구하는 망 구현 철학과 실제 산업계와 시장의 동향을 면밀히 관찰하여 미래를 예측하는 자세가 필요하다.

### 2.3. France Telecom의 QoS 보장방안

France Telecom (이하 FT)에서는 위의 1, 2절에서 살펴본 기존 IntServ, DiffServ, 혹은 이들의 hybrid 형태의 QoS 보장방안에서 제시하는 철학과 크게 다른 아이디어를 바탕으로 하는 방안을 제안하고 있다[19] [20]. FT는 기존의 IntServ와 DiffServ는 트래픽이 혼합되면서 발생하는 statistical multiplexing gain을 염두에 두고 있지 않기 때문에 네트워크의 효율성을 크게 해친다고 주장하면서 이들 두 가지 방안이 모두 현실적으로 cost-effective하지 않다고 결론 내린다. 이들은 현존하는 네트워크를 관찰하여 streaming (real time) 트래픽과 elastic (non-real time) 트래픽이 대략 1:9의 비율로 존재한다는 것을 발견하였다. 이 수치는 향후 application 특성이 변화하면서 달라질 수 있는 수치이긴 하지만 당장 크게 변하는지는 않을 것으로 예측했다. 이러한 비율을 가지는 조건하에서 simulation을 실시한 결과 90%의 네트워크 utilization 상황에서도 네트워크 각 링크에서 문제없이 서비스가 이루어지는 것을 발견했다. 하지만 이 utilization이 100%를 넘어가면서 급격히 링크의 서비스 특성이 나빠지며 전체 트래픽이 정체되는 현상을 발견했다. 따라서 전체 트래픽 중 상당량에 이르는 elastic한 특성을 가진 트래픽에 대한 적절한 control, 특히 admission control이 필요하다고 주장한다. 이를 위해 FT는 'Implicit admission control'을 제안하였다. 이것의 특징 혹은 동작원리는 다음과 같이 설명될 수 있다.

- 자원예약을 하지 않으며 따라서 이를 위해 필요한 시그널링도 전혀 없다.
- 패킷 헤더의 IP address pair등을 이용하여 flow를 identify한다.
- 현재 네트워크 상태를 모니터 하여 새로운 플로우의 인입을 막는다.

위의 단계 중 현재 상태를 모니터 하여 새로운 플로우의 인입을 어느 시점부터 막는가를 결정하는 것이 어려운 문제이며 실제로 FT에서도 이 문제에 대한 적절한 해결책을 제시하고 있지 못하다. 다만 특별한 네트워크 parameter하에서 M/D/1 큐를 가정하여 approximation하는 정도로 block 시점을 결정하는 정도이다.

이러한 implicit admission control은 다음 두 가지 advancement로 크게 성능이 향상될 수 있으며 실제로 구현 및 deploy가 가능한 정도의 완성도 있는 QoS architecture로 발전하였다.

1. 전체 network의 load가 일정 수준 이하이면 congestion 측정 및 admission control을 전혀 수행하지 않는다.
2. Misbehaving flow들을 제어하는 방법으로 per-flow fair queueing을 채택한다.

첫 번째 feature는 직관적으로 이해가 가능한 부분이다. 즉, admission control 전체를 중지시키고, 전체 network의 load가 일정 수준 이상이 되는 순간부터 각 flow별 모니터링을 실시하는 것이다. 이로 인해 네트워크의 부하가 크게 줄어들 것으로 예상된다. 두 번째의 fair-queueing feature는 implicit admission control로 훌륭히 조화를 이루면서 FT가 Flow-aware Network (FAN)이라고 이름 지은 QoS architecture를 완성하는 부분이다. 즉, flow별로 정확히 protection을 해준다면 다른 일체의 control 없이 네트워크에 존재하는 트래픽의 총량만을 제어하여 필요한 성능을 얻을 수 있다는 것이다.

이러한 FAN은 기존 IntServ와 DiffServ와는 전혀 다른 QoS architecture이다. 기존 방식들은, 필요한 만큼의 자원을 end-station이 파악하여, end-station이 주문, 예약하고, 예약된 만큼의 트래픽을 생성하여 보내주면 이에 대해 네트워크는 수동적으

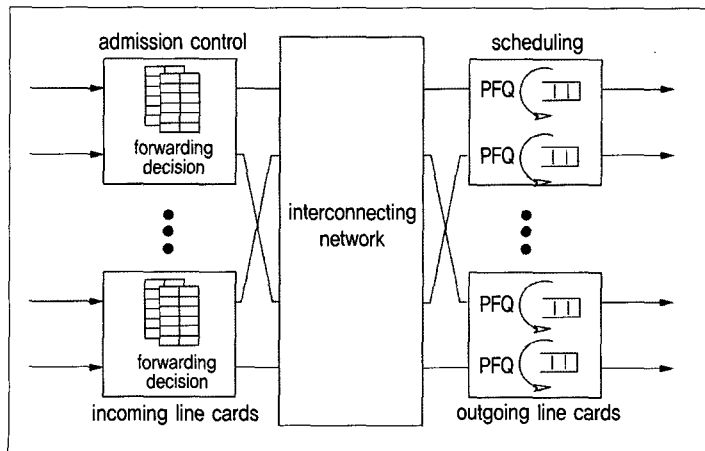
로 서비스를 제공하는데 반하여, FAN에서는 end-station은 전혀 intelligence 없이, 네트워크가 플로우를 identify하여 플로우 별로 protection을 해주고, 네트워크상의 트래픽을 총량을 규제하여 전체 플로우를 만족시킨다. 결과적으로 per-flow fair queueing을 양 쪽 모두 수행하지만, 한쪽에서는 예약된 자원의 제공의 측면에서 이것이 사용되었다면 FAN에서는 플로우간 protection방법으로써 사용되었다고 볼 수 있다.

이러한 FAN의 아이디어를 실제로 구현한 예가 그림 1의 Cross-protect router이다[19]. 여기서는 크게 두 가지 기능 entity가 정의되는데 ingress line card상에 위치하는 forwarding decision (admission control을 포함하는)과 egress line card상의 per-flow queueing module이 그것이다. Ingress line card는 Protected Flows List (PFL)를 가지고 있어서, 이 list에 포함되어 있는 플로우들은 congestion 상황에서도 admission control을 받지 않는다. 즉, 이미 네트워크에서 일정 기간 이상 서비스 받고 있었으므로 새로 발생하는 congestion상황에서는 새로운 플로우들만 제어하게 되는 것이다.

특정 플로우를 PFL에 포함시키는 문제는 per-packet probabilistic acceptance라고 하는 criteria를 사용한다. 이 criteria에 따르면 오래 진행된 플로우일수록 list에 포함되게 될 확률이 높아진다. Admission control은 설정한 바대로 implicit admission control을 사용하며, 일정한 시간간격을 설정하여 네트워크의 congestion 상황을 모니터 하여 필요하다고 판단되는 경우 새로운 플로우 즉, PFL에 포함되어 있지 않는 플로우들에 대하여 인입을 금지한다. Egress line card에서는 per-flow scheduling이 필요한데, 이의 복잡도는 크지 않다는 것이 France Telecom의 주장이다. PFL에 등록된 플로우의 수보다 실제 output queue에서 active한 플로우의 수는 훨씬 작다고 한다.

이러한 획기적인 FAN architecture도 다음과 같은 치명적인 문제점을 내포하고 있다.

1. 측정에 의한 admission 결정 criteria에 명확하지 않으며 네트워크의 parameter에 따라 크게 달라질 수 있다. 즉, 네트워크 별로 미리 세밀하게 조정된 criteria를 가지고 있다가 상황에 맞게 대처하지 않으면 적절한 관리가 이루어지기



(그림 1) FT의 Cross-Protect router

힘들다는 것이다. Cross-protect에서는 “fair-rate”이라고 하는 measure를 이용하여 이것이 100Kbps이하이면, 즉, 플로우 별로 받는 평균화된 서비스 량이 100Kbps이하이면 congestion 상황이라고 보고 새로운 플로우를 block하게 된다. 하지만 이 수치는 전체 네트워크를 간단한 M/D/1 서버들의 집합으로 보고 계산한 수치이며 다른 네트워크에서는 일반적으로 적용되지 못한다.

2. 플로우의 특성과는 상관없이 동일한 rate로 모든 플로우들이 서비스 받는다. 이 문제는 FAN의 기본 철학과도 관계가 있는데, 만약 네트워크가 “투명하고” 각 플로우가 per-flow fair queueing으로 protection 된다면, 즉 모든 상황이 통제 가능하다면 문제가 없겠지만, 잠깐이나마 네트워크에 부하가 많이 걸려 “불투명한” 상황이 발생한다면 streaming 플로우를 포함한 모든 플로우가 적절치 못한 서비스를 받게 되어 치명적인 성능상의 문제가 될 수 있는 부분이다.
3. 마지막으로, 짧은 플로우들이 상대적으로 admission control에 의한 blocking을 덜 경험하게 된다. 이를 보상하기 위해서 PFL에서 플로우의 등록 criteria를 probabilistic하게 유지하고 있다. 하지만 일반적인 네트워크에서 서로 다른 parameter하에서도 충분한 보상이 되는지는 좀 더 검증해 보아야 할 것이다.

살펴본 바와 같이 France Telecom의 FAN QoS architecture의 특성을 정리하면 다음과 같다.

- QoS architecture 전체에 걸쳐 전혀 시그널링이 필요 없다.
- 혼잡이 발생하기 시작할 때까지 admission control이 수행 되지 않는다.

- 혼잡이 감지되면 implicit admission control을 수행한다.
- 모든 플로우에 대해 똑 같은 priority와 대역폭이 제공된다.
- 모든 플로우에 대해 fair-queueing을 적용함으로써 protection을 제공한다.
- 최소한의 제어로 일반적인 상황에서의 QoS를 보장한다.
- 복잡한 일들은 core에서 수행하고 edge와 end-station은 전혀 involve되지 않는다.

FAN이 제시하는 insight는 작지 않다. 만약 이 architecture가 성공적으로 deploy된다면 그 impact는 상당할 것이다. 그렇지 않더라도, QoS 관련 연구자들에게 새로운 연구 방향을 제시하고 있다는 점에서 주목할 만하다.

### III. ITU-T SG13에서 제안된 QoS 보장방안

#### 3.1. Y.flowreq

2006년 5월의 일본 고베 미팅에서 시작하여 최근의 7월 제네바 미팅에서 까지 British Telecom (이하 BT)에서 주도하여 draft recommendation으로 인정 받은 IP-Flow State Aware requirements (이하 Y.flowreq)는 NGN Transport stratum (data-plane)에서의 구체적인 QoS 보장방안으로, NGN QoS구조 전체에서 큰 의미를 가지고 있으며, 간략히 핵심내용을 설명하자면 다음과 같다. ITU-T Recommendation Y.1221, “IP Transfer Capability”에 의하면, 플로우 별 “state”정보를 각 네트워크 노드가 관리하고 있어서 필요에 따라서 (혼

잡상황 발생시 등) 특정 플로우의 패킷만을 집중적으로 drop하게 할 수 있어야 한다. 따라서 각 노드에서 “Flow-state aware” capability를 가지고 있어야 하며 이에 대한 구체적인 구현방법, 특히 Transport stratum에서의 동작원리를 Y.flowreq에서 정의하려고 하고 있는 것이다. 여기서 정의하는 QoS 보장방안은 다음과 같은 기본 구조를 가지고 있다.

플로우는 IPv4의 경우는 IP address pair, Port number pair, Protocol의 5-tuple, 그리고 IPv6위 경우는 {DA, SA, Flow label}이 동일한 packet들의 집합으로 정의된다.

플로우는 다음과 같은 사항에 대해서 네트워크에 explicit하게 요구한다.

- Maximum Rate request
- Preference Level request
- Authorization & Authentication (optional)
- Service Context request

이러한 request는 “start packet”의 in-band 시그널링 형태로 네트워크에 전달된다.

위에서 명시한 기본적인 request 중 Service Context는 플로우의 특성을 가장 잘 대표하는 request이며 다음의 네 가지 종류의 Service Context가 정의되어 있다.

- Available Rate: 사용자 터미널 (최종 터미널 혹은 Customer Premises Equipment, CPE)은 start packet으로 최초 request를 시도하며 이 후 start packet을 반복함으로써 request의 내용을 변경한다. Start packet에는 원하는 “maximum rate”을 명시하며, CPE는 이 후 해당 maximum rate보다 작은 rate으로 패킷을

전송하여야 한다. 네트워크는 요구하는 maximum rate으로 해당 플로우의 rate을 조절해 주어야 한다. 내부 혼잡상황에 따라 필요한 경우 CPE는 ATM ABR의 RM cell과 같은 형태의 in-band 시그널링을 사용하여 주기적으로 네트워크의 상황을 파악하여야 하며 이 경우 CPE는 maximum rate을 낮추어야 한다.

- Guaranteed Rate: 본 service context는 기존 IntServ의 Guaranteed Rate service과 유사하다. CPE는 start packet으로 최초 maximum rate을 요구하면 네트워크는 이에 대해 admission control을 실시한다. 즉, 플로우의 요구를 받아들이지 않고 해당 플로우를 reject하거나 혹은 accept하고 플로우의 path상의 모든 노드들이 필요한 만큼의 자원을 예약하고 이를 제공한다. 이런 경우 해당 플로우는 “discard last” state을 지정 받아, 혼잡상황에서도 가능한 패킷 drop이 최소화될 수 있다. 이러한 GR service context는 streaming 실시간 플로우 등의 응용에 적합하다.

- Maximum Rate: 본 service context는 CPE가 start packet에서 명시하여 요구하는 maximum rate과는 전혀 다른 것이다. MR service context의 플로우는 GR과 마찬가지로 start packet을 이용하여 maximum rate을 요구하며, 현재 네트워크에서 사용 가능한 자원의 존재 여부와는 상관없이 받아들여 진다. 이 부분이 GR service context와는 다르다. 각 플로우 별로 명시하는 maximum rate에 대해서 서비스를 보장하지는 않지만 타 service context와 비교했을 때 loss나 delay, delay variance 등의 성능 면에서 높은 priority를 제공하는 것을 원칙으로 하고 있다. 이것이 GR과 MR service context간의 priority 순위를 결정하는



것을 어렵게 만드는 점이다. 본 service context는, 명시된 바는 없지만, emergency call (911.으로 대표되는) 등의 응용에 적합하다.

- Variable Rate: 본 service context는 기본적으로 GR에 AR의 feature를 혼합한 것으로 볼 수 있다. CPE는 start packet으로 플로우의 총 생존 기간 동안 보장 받아야 하는 rate을 명시하고, 이후 AR과 마찬가지로 필요와 상황에 따라 maximum rate을 변동시킬 수 있다. 따라서 AR과 비슷하지만 어떠한 경우라도 보장되는 minimum rate이 있다는 점에서 차별된다.

이러한 CPE의 요구사항에 따라 네트워크는 노드마다 flow-state을 유지하여 각 플로우를 관리한다. 이러한 flow-state은 아주 간단하게는 “discard first”와 “discard last”로만 지정 될 수도 있지만 현실적으로는 플로우 별 lookup이 가능하도록 hash key가 필요할 것이며, time stamp, IP header, 네트워크에서의 루트, 여러 가지 counter들과 특히 플로우가 속한 service context에 대한 정보들이 포함되어야 할 것이다.

마지막으로 시그널링 방법에 대해서 살펴보면, 언급한 바와 같이 start packet등의in-band. 시그널링을 사용하며, 이에 따라 이를 표시할 수 있는 방법이 필요한데 현재로는 IP header의 DiffServ field가 유일한 표시방법으로 여겨진다. 따라서 해당 필드로 시그널링 패킷임을 명시하고, 내부 payload부분에 다음과 같은 내용을 표시한다.

- Type of message
- Request, Response, Confirm, Missed, Renegotiate, Close
- Parameters
- Rate: (1kbps ~ 3Tbps, or 1kbps ~  $3 \cdot 2^{32}$

kbps)

- Preference priority
- Delay priority
- Code for type indication
- Code for service context indication
- Burst tolerance

### 3.2. 문제점

Y.flowreq 방식은 아직 충분히 검토되고 검증되지 않은 단계이며 다음과 같은 문제점을 가진다.

- Transport stratum (data plane)에서의 시그널링 기능이 없는 대부분의 사용자 터미널에의 서비스 제공에 대한 고려가 없다. 따라서 이들 대부분의 트래픽은 다른 방안이 제시되지 않는다면 기존 방식대로 best-effort service를 제공할 수 밖에 없다.
- GR service context를 제외한 나머지 service context의 플로우에 대한 admission control 방안이 명시되어 있지 않다.
- 노드마다 flow-state을 유지 관리해야 하는 이 유가 명확하지 않다. GR service context를 제외하면 per-flow scheduling이 필요 없어 보이며, 그렇다면 per-flow 관리는 buffer management, 즉 혼잡 시의 선별적인 패킷 drop을 위해서만 필요하다. 하지만 선별적으로 drop하기로 결정한 플로우들이 있다면 이들에 대한 state만 유지해도 충분할 것이다.
- 플로우 별 state 정보의 유지관리 등의 per-flow maintenance, 또는 패킷 별 lookup등의 per-packet maintenance 에 코어 노드의 복잡성이 위험할 정도로 높아질 가능성이 있다. 플로우 별 state 정보를 저장하기 위한 메모리 용량은 10Gbps 링크당 25MByte 정도로 크지

않겠지만, 문제가 되는 것은 사실 5-tuple lookup을 통한 packet classification 등의 processing complexity이다.

#### IV. 결론 및 보완책

3장에서 명시한 Flow-state aware requirement의 문제점을 바탕으로, 2장에서 검토한 많은 대안들이 제시하는, 효율적이면서 적용 가능한 아이디어 구체적인 방안으로는 다음과 같은 것들을 들 수 있다.

- 사용자 패킷 표시: IP 헤더의 DSCP 필드를 이용하여 해당 패킷이 Service context등이 명시된 사용자 패킷이라는 것을 명시한다. 이렇게 함으로써 대부분을 차지할 것으로 보이는 best-effort 트래픽에 대해서는 코어노드가 packet classification을 할 필요가 없이 longest prefix matching 만으로 처리할 수 있을 것이다.
- 선택적 플로우 통합: Service context, Preference level, 네트워크에서의 path가 같은 플로우들을 통합하여 코어노드에서의 부하를 크게 줄일 수 있다.
- 능동적 플로우 파악: 스스로 “start packet”을 만들 수 없는 CPE들, 즉 Transport stratum에서의 시그널링 기능이 없는 CPE중 premium service를 요청하는 CPE들을 대상으로 특별히 ingress edge 노드에서 해당 플로우들을 찾아내어 CPE 대신 start packet을 만드는 등 QoS 보장을 위하여 적극적으로 개입한다. 이로써 단순한 기능만 구현되어 있는 light-weight CPE에 대해서도 QoS 보장이 가능하게 된다.
- Endpoint admission control (EAC): EAC는 최근 5년 사이에 아주 많은 연구가 진행된 분야이기도 하다[21~26]. 간단히 EAC를 설명하자면, 네트워크의 end point (CPE나 ingress edge node)에서 새로운 플로우에 대해서 탐사 패킷 (probing packet)을 만들어서 내보내고, 이 탐사 패킷이 혼잡상황을 암시적으로 (implicitly) 파악한 후 플로우의 인입 여부를 결정하는 것이다. 이러한 탐사 패킷들은 실제 플로우의 특성과 같아야 하며 심지어는 플로우의 실제 패킷을 탐사 패킷으로 사용할 수 있다. 이러한 EAC는 NGN에서 정의하고 있는 AR과 VR service context와 조화를 이루어 철저한 실시간 서비스를 요구하는 application을 제외하고 상당히 많은 종류의 application에 적용될 수 있을 것이다.

제안한 몇 가지 개선방안과 기존 Y.flowreq를 기반으로 한 QoS 보장방안은 국내에서 추구하는 BcN에서의 플로우기반 QoS 실현방안을 그 시작점으로 하고, 이제껏 제시되었던 많은 QoS architecture의 장점을 살리면서 동시에 실제로 구현이 가능한 방안이다. 이렇게 실제로 구현 가능한, 성능 면에서도 효율적이며 글로벌 표준과도 부합하는 QoS 보장방안을 연구 개발하여 선점함으로써 여타의 선진국이나 선진업체보다 한발 먼저, 차세대 통신망에서의 입지를 구축할 수 있을 것이다.

#### [참고 문헌]

- [ 1 ] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An architecture for Differentiated Services,”, IETF RFC

- 2475, Dec. 1998.
- [2] V. Jacobson, K. Nichols, and K. Poduri, "An Expedited Forwarding PHB," RFC 2598, June 1999.
- [3] D. D. Clark and W. Fang, "Explicit allocation of best effort packet delivery service," *IEEE/ACM Trans. Networking*, vol. 6, pp. 362?373, Aug. 1998.
- [4] A. Charny and J. Y. Le Boudec, "Delay bounds in a network with aggregate scheduling," in *Proc. QOFIS*, Oct. 2000.
- [5] I. Stoika and H. Zhang, "LIRA: An approach for service differentiation in the internet," in *Proc. NOSSDAV*, 1998, pp. 115?128.
- [6] S. Sahu, P. Nain, D. Towsley, C. Diot, and V. Firoiu, "On achievable service differentiation with token bucket marking for TCP," in *Proc. ACM SIGMETRICS*, June 2000.
- [7] I. Stoika and H. Zhang, "Providing guaranteed services without per flow management," in *ACM SIGCOMM*, Sept. 1999, pp. 81?94.
- [8] K. Nichols, V. Jacobson, and L. Zhang, "A two-bit differentiated services architecture for the Internet," *Network Working Group*, RFC 2638, July 1999.
- [9] R. Sivakumar, T. Kim, N. Venkitaraman, and V. Bharghavan, "Achieving per-flow weighted rate fairness in a core stateless network," in *Proc. IEEE Int. Conf. Distributed Computing Systems (ICDCS)*, Taipei, Taiwan, Apr. 2000, pp. 188?196.
- [10] R. Cruz, "SCED+: Efficient management of quality of service guarantees," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar. 1998, pp.625?634.
- [11] J. Kaur and H. M. Vin, "Core-stateless guaranteed rate scheduling algorithms," In *Proc. of IEEE INFOCOM*, volume 3, pages 1484--1492, April 2001.
- [12] S. Wang, D. Xuan, R. Bettati and W. Zhao, "Providing Absolute Differentiated Services for Real-time Applications in Static-Priority Scheduling Networks," *IEEE/ACM Transactions on Networking*, Vol. 12, No. 2, April 2004.
- [13] Y. Jiang, "Link Based Fair Aggregation: A Simple Approach to Scalable Support of Per-flow Service Guarantees," in *Proceeding of IFIP Networking 2004*.
- [14] C. Dovrolis and P. Ramanathan, "A case for relative differentiated Services and the proportional differentiation model," *IEEE Network*, vol.13, pp. 26?34, Sept./Oct. 1999.
- [15] C. Dovrolis, D. Stiliadis and P. Ramanathan, "Proportional Differentiated services: Delay Differentiation and Packet Scheduling," *IEEE/ACM Transactions on networking*, Vol. 10, No. 1, Feb. 2002.
- [16] H. Chow, "A feedback control extension to differentiated services," draft-chow-diffservbctrl-00.txt. 1999.
- [17] N. Li, M. Borrego and S. Q. Li, "Achieving per-flow fair rate allocation in DiffServ," *ACM Transactions on Modeling and*

- Computer Simulation, Vol. 11, No. 2, April 2001.
- [18] M. Mahajan, A. Ramanathan and M. Parashar, "Active resource management for the differentiated services environment," International Journal of network Management, Vol. 14, 2004.
- [19] A. Kortebi, S. Oueslati, and J. Roberts, " Cross-protect: implicit service differentiation and admission control," in Proc. IEEE HPSR, April 2004.
- [20] S. Oueslati and J. Roberts, "A new direction for quality of service: Flow aware networking," in Proc. conference on Next Generation Internet Networks (NGI), April 2005.
- [21] Jingyu Qiu and Edward W. Knightly, "Measurement-Based Admission Control with Aggregate Traffic Envelopes," IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 9, NO. 2, APRIL 2001
- [22] Matthias Grossglauser and David N. C. Tse, "A Time-Scale Decomposition Approach to Measurement-Based Admission Control," IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 11, NO. 4, AUGUST 2003
- [23] Sugih Jamin, Peter B. Danzig, Scott J. Shenker, and Lixia Zhang, "A Measurement-Based Admission Control Algorithm for Integrated Service Packet Networks," IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 5, NO. 1, FEBRUARY 1997
- [24] Lee Breslau, Edward W. Knightly, Scott Shenker, Ion Stoica, and Hui Zhang , "Endpoint Admission Control: Architectural Issues and Performance," SIGCOMM '00
- [25] G. Bianchi, et.al, "Endpoint Admission Control with Delay Variation Measurements for QoS in IP Networks," SIGCOMM '02
- [26] Yingxin Jiang and Aaron Striegel "A Distributed Traffic Control Scheme Based on Edge-centric Resource Management," SIGCOMM '06



**정진우**

1992년 KAIST 전기전자공학부 학사  
1994년 Polytechnic Univ. 전기과 석사  
1997년 Polytechnic Univ. 전기과 박사  
1997년 ~ 2001년 삼성전자 중앙연구소 책임연구원  
2001년 ~ 2005년 삼성전자 종합기술원 수석연구원

2005년 ~ 현재 상명대학교 소프트웨어학부  
관심분야 : NGN QoS, Switch architecture, Wireless Network, System on Chip



**송종태**

1990년 KAIST 전기전자 졸업 학사  
1994년 남가주대 전기과 졸업 석사  
1998년 Brooklyn Polytechnic 전기과 졸업 박사  
1991년 ~ 1992년 삼성전자 정보통신 연구소 연구원  
1998년 ~ 2001년 Lucent Technologies 연구원 (MTS)

2001년 ~ 2004년 캐리어급 MPLS 라우터, NG SONET 관련 벤처  
2004년 ~ 현재 한국전자통신연구원 BcN연구단 선임연구원  
관심분야 : Switch Architecture, Next Generation Network 제어 구조, OAM, Protection



**이순석**

1988년 성균관대학교 산업공학과 공학사  
1990년 성균관대학교 산업공학과 공학석사  
1993년 성균관대학교 산업공학과 공학박사  
1993년 ~ 2002년 한국전자통신연구원 선임연구원  
2002년 ~ 현재 한국전자통신연구원 책임연구원  
2005년 ~ 현재 한국전자통신연구원 광대역통합망

연구단 BcN실계팀 팀장  
관심분야 : 차세대 네트워크 및 서비스 아키텍처, 네트워크 진화 전략, 네트워크 구조 및 최적설계, 트래픽 엔지니어링, 네트워크 및 통신시스템 성능평가