

SCORE (Scalable CORE) 네트워크 모델을 기초로한 망의 확장성 및 실시간 서비스 지원에 대한 연구

홍 흥 표, 석 정 봉

연세대학교 전산학과

전화 : 033-760-2293 / 핸드폰 : 016-362-8974

Supporting Real-Time Service and Scalability of Network based on SCORE (Scalable CORE) network model

Heung-Pyo Hong, Jeong-Bong Suk

Dept. of Computer Science, Yonsei University

E-mail : yoguru@empal.com

Abstract

본 논문은 인터넷에서 실시간 서비스를 효과적으로 지원하면서도 망의 확장성을 제공하는 방법을 제안하였다. 이를 위해 코어노드에서 각 플로우의 상태 정보를 관리하지 않는 SCORE (Scalable CORE) 네트워크 모델 [1]과 종단간 딜레이 보장이 우수한 QoS (Quality of Service) 스케줄링 방법중 하나인 WF²Q+ (Worst-case Fair weight Fair Queuing+) [2]에 대해서 기술하고 구현하였다. 이러한 개념을 바탕으로 새로운 망의 구조를 제안하였으며 시뮬레이션을 통한 성능분석을 수행하였다.

I. 서론

현재의 인터넷 환경은 실시간 형태의 멀티미디어 응용들을 위하여 새로운 형태의 프로토콜이 개발되고 있고, 라우터 및 호스트 환경도 급격하게 진화하고 있다. 이러한 네트워크 환경을 위해서는 기존의 best-effort 서비스뿐만 아니라, 음성 및 비디오와 같은 멀티미디어형 서비스 등의 다양한 서비스의 요구 사항을 지원할 수 있는 새로운 개념의 네트워크 구조와 방법이 요구된다.

IETF (Internet Engineering Task Force)에서

는 이러한 요구를 충족시키기 위하여 인터넷 통합 서비스 모델 (Internet Integrated Services Model)을 제안하였다 [3]. 통합 서비스 모델에서는 자원예약 프로토콜을 이용하여 각 패킷의 종단간 지연에 대한 제어를 제공하며 서로 다른 트래픽 클래스들이 특정 링크의 대역폭을 공유할 때 이것을 제어할 수 있도록 하고 있다. 이러한 통합 서비스의 문제점은 플로우의 수가 증가하면 관리해야 하는 정보의 양도 증가하기 때문에 그 정보를 저장할 버퍼 공간에 대한 제약이 발생하고, 망 내의 플로우 정보를 처리하기 위한 오버헤드로 인한 확장성의 제약이 따른다는 것이다. IETF에서는 확장성 문제의 해결을 위하여 차별화 서비스 모델 (Differentiated Service Model) [4]을 제안하고 있다. 차별화 서비스 모델은 플로우 단위가 아닌 플로우를 몇 개의 클래스로 군집화 하여 서비스함으로써 통합 서비스 모델에서의 복잡한 패킷 처리과정을 보다 간략화 시켜 네트워크에서의 오버헤드를 줄이는 방식이다.

본 논문에서는 네트워크에 확장성을 제공하고 실시간 서비스를 지원하기 위하여, 차별화 서비스 모델과 비슷하게 코어 노드에서 각 플로우의 상태정보를 관리 하지 않는 SCORE (Scalable CORE) 네트워크를 기반으로 연구를 수행하였다. 2장에서 SCORE 네트워크의 기본적인 개념에 대

해서 살펴보고, 3장에서는 실시간 서비스 지원을 위한 스케줄링 방법으로 종단간 딜레이 보장이 우수한 WF^2Q+ 에 대해서 기술한다. 4장에서 시뮬레이션 모델의 기본 개념을 기술하고 결과를 분석한 뒤 5장에서 결론을 내린다.

II. SCORE 네트워크

인터넷 통합 서비스 모델의 문제점인 확장성의 문제를 해결하기 위해 제안된 방식중의 하나가 SCORE 네트워크 모델이다.

SCORE 네트워크에서 에지 라우터들은 각 플로우의 상태 정보를 관리하고 코어 라우터들은 각 플로우의 상태정보를 관리하지 않는다. 이렇게 네트워크 코어에서 각 플로우의 상태 정보를 관리하지 않으면서도 통합 서비스 모델과 같은 stateful 네트워크에서 제공되는 서비스와 유사한 서비스를 제공하는 것이 이 모델의 목적이다.

SCORE 네트워크를 구현하는데 핵심이 되는 기술은 DPS (Dynamic Packet State)이다. DPS의 메인 아이디어는 라우터가 각 플로우의 상태 정보를 인스톨하고 관리하는 대신 패킷에 각 플로우의 상태 정보를 담아서 보내자는 것이다. 상태 정보는 에지 노드에서 초기화 되어 패킷의 헤더에 삽입된 뒤 코어 노드로 보내진다. 코어 노드에서는 다음 홉으로 패킷을 전송하기 전에 라우터의 내부 상태와 패킷 헤더의 상태 정보를 갱신하게 된다. 이렇게 DPS를 사용함으로써 에지 노드에서만 각 플로우의 상태 정보를 관리하고 코어 노드에서는 단지 패킷 헤더에서 정보를 얻어 패킷을 처리하게 되므로 각 플로우의 상태 정보를 유지하지 않으면서도 상태정보를 유지하는 네트워크와 같은 효과를 달성할 수 있는 것이다.

III. WF^2Q+ 스케줄러

3.1 GPS (Generalize Process Sharing)의 개념

인터넷에서 각 플로우가 요구하는 다양한 서비스 품질 중 실시간 서비스를 요구하는 플로우에 대해서는 일정한 대역폭과 종단간 딜레이를 보장

하고, 각 플로우들이 할당된 대역폭에 따라서 자원을 공평하게 공유하도록 하는 방법이 필요하다.

GPS (Generalized Process Sharing) 스케줄링 방식은 실시간 서비스 요구 특성을 만족시키기 위해 제안된 것이다. 이 방식은 각 플로우를 물 흐르는 것처럼 가정하고 각 플로우에 서비스를 동시에 제공하여 각각의 서비스 클래스에 대해서 완전한 독립성과 공평성을 보장하는 방법이다.

3.2 WF^2Q+ (Worst case Fair weight Fair Queuing+)

본 연구에서는 보다 정확하게 GPS 방식을 구현하기 위해서 시작시간과 종료시간 두 가지를 이용하여 서비스 순서를 결정하는 WF^2Q+ 를 구현하였다.

WF^2Q+ 는 WF^2Q 와 동일한 알고리즘을 사용하고 있으며, WF^2Q 의 구현 복잡성을 줄이기 위해 수정된 알고리즘이다. WF^2Q+ 는 GPS를 근사화한 많은 PFQ (Packet Fair Queuing) 알고리즘들 중에서 가장 잘 근사화 된 알고리즘이다. WF^2Q+ 는 GPS에 가장 가깝게 근사화 하기 위하여 eligible이란 개념을 도입하고 있다. eligible이란 다음에 서비스될 패킷으로 선택될 수 있는 자격이 있는지의 여부를 나타낸다. 그리하여 WF^2Q+ 알고리즘에서 가장 중요한 개념은 패킷이 eligible인지 아닌지의 여부를 가리는 것이다. 이러한 판단의 기준은 GPS 시스템 상에서 그 패킷의 서비스 시작 여부를 판단하게 된다. 즉 GPS 시스템 상에서 서비스를 받고 있을 것으로 예상되는 패킷들은 WF^2Q+ 시스템에서 eligible한 패킷으로 판정되고, 그렇지 못한 패킷들은 eligible하지 못한 패킷으로 판정된다.

WF^2Q+ 알고리즘은 SEFF (Smallest Eligible Finish time First) 서비스 원칙을 따른다. 이것은 eligible한 패킷들 중에서 가상 종료 시간이 가장 작은 패킷을 선택해서 서비스한다는 것을 말한다. 이에 반해 WFQ (Weighted Fair Queuing)와 SCFQ (Self Clocked Fair Queuing) 알고리즘은 가장 작은 가상 종료 시간을 가진 패킷을 서비스하는 SFF (Smallest Finish time First) 서비스 원칙을 따른다. 일반적으로 SFF 서비스 원칙을 따르는 트래픽 스케줄링 알고리즘의 출력 링크에서의 트래픽 특성을 살펴보면 예약된 서비스 비율이 높

은 세션의 트래픽이 폭발적으로 증가하여 출력되는 것을 관찰할 수 있다. 이러한 트래픽은 네트워크 내에서 또 다른 혼잡을 유발할 수 있다. 하지만 WF²Q+ 알고리즘은 이러한 현상의 발생을 최대한으로 억제하면서 공정성도 좋게 하였다.

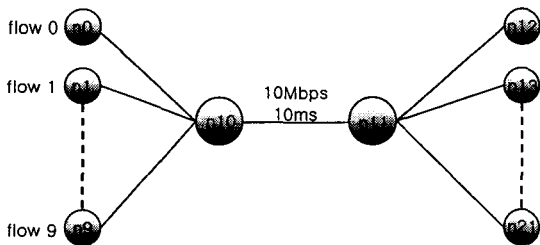
IV. SCORE 네트워크 기반 WF2Q+의 모델링 및 성능 평가

본 장에서는 제안된 모델이 인터넷에서 실시간 서비스가 요구하는 QoS를 보장할 수 있도록 대역폭을 보장할 수 있는가와 패킷에 대한 지연시간 및 도착시간에 대해 분석하였다.

시뮬레이션은 리눅스 환경에서 NS-2 (Network Simulator) 2.1b7a 버전을 이용하였고 실제 네트워크를 단순화시켜 수행하였다.

4.1 시뮬레이션 모델

첫 번째 시뮬레이션 네트워크 구조는 [그림 1]과 같다.

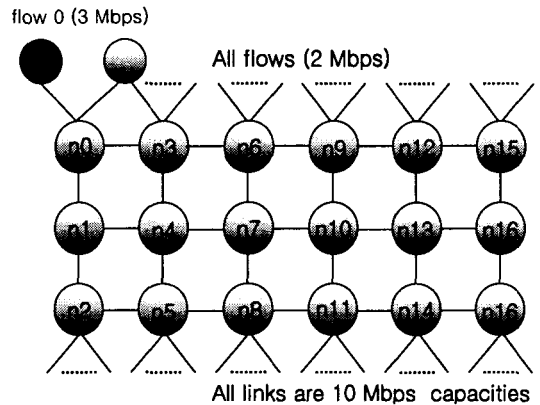


[그림 1] 기본 구조 네트워크 모델

시뮬레이션 네트워크에서 각각의 노드는 n0로부터 n9까지 10개의 송신 노드, n12부터 n21까지 10개의 수신 노드 그리고 n10과 n11의 라우터로 구성되어 있다. 각각의 노드를 연결하고 있는 모든 링크는 10Mbps의 대역폭과 10ms의 지연 시간을 갖고 10개의 송신 노드는 1000bytes의 길이를 갖는 패킷을 발생 시킨다. 노드 n0에서 통합 서비스 모델의 GS (Guaranteed Service)에 준하는 트래픽을 발생시킨다. 나머지 노드는 Best-effort 서비스를 요구하는 트래픽을 발생시킨다. 10개의 송신 노드들이 발생시키는 전체 트래픽을 노드 n10과 노드 n11 사이의 링크 대역폭을 초과하도

록 발생시키며 이러한 혼잡 링크 (congestion link) 상에서 실시간 서비스가 요구하는 대역폭을 보장받을 수 있는가와 패킷의 지연시간, 도착 시간 간격에 대한 성능평가를 수행한다.

두 번째 시뮬레이션 네트워크의 구조는 [그림 2]와 같다.



[그림 2] 혼잡 구조 네트워크 모델

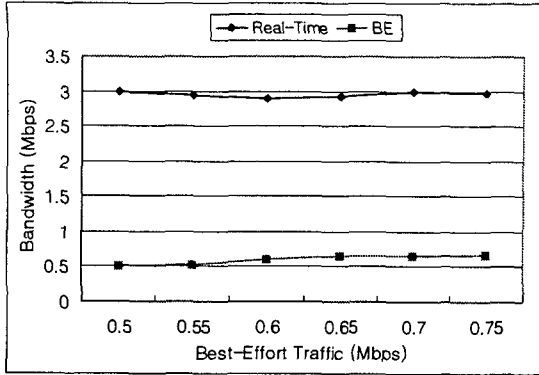
시뮬레이션 네트워크에서 노드 n0는 실시간 서비스를 요구하는 트래픽을 발생시키며 나머지 노드들은 Best-effort 서비스를 요구하는 트래픽을 발생시킨다. 각각의 노드를 연결하고 있는 모든 링크는 10Mbps의 대역폭과 10ms의 지연 시간을 갖고 10개의 송신 노드는 1000 bytes의 길이를 갖는 패킷을 발생시킨다. 노드 n0에서 발생시킨 실시간 트래픽에 대해서 혼잡 링크수를 1부터 5까지 증가시켜가면서 대역폭을 보장 받을 수 있는가를 성능평가 한다.

4.2 시뮬레이션 결과 분석

[그림 1]의 시뮬레이션 네트워크 구조에서 환경 변수는 [표 1]과 같다.

[표 1] 시뮬레이션 환경변수 1

서비스종류	트래픽 발생시간	트래픽 발생율	예약율	패킷의 크기
Real-Time	9 sec	3 Mbps	3 Mbps	1000 bytes
BE	9 sec	0.5→0.75	없음	1000 bytes



[그림 3] BE 트래픽 증가에 따른 실시간 서비스의 대역폭 변화

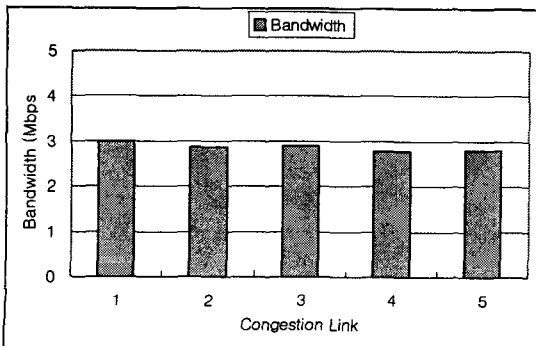
[그림 3]의 결과에서 n2에서 n9까지의 노드가 노드 n10과 n11의 링크 대역폭을 초과하지 않는 0.5 Mbps의 트래픽을 발생시켰을 경우에 3Mbps의 트래픽을 발생시킨 실시간 서비스는 자신이 발생시킨 트래픽의 대역폭을 보장 받고 있다.

최선형 서비스의 트래픽이 각각 0.55 Mbps 이상의 트래픽을 발생시킬 경우에 전체 트래픽의 양은 n10과 n11의 링크 대역폭 10Mbps를 초과한다. 하지만 최선형 서비스의 트래픽이 0.65 Mbps 이상이 된다 하더라도 실시간 서비스는 자신이 요구하는 대역폭을 보장 받고 있는 것을 볼 수 있다.

[그림 2]의 시뮬레이션의 환경 변수는 [표 2]와 같다.

[표 2] 시뮬레이션 환경변수 2

서비스 종류	Real-Time, BE
트래픽 발생시간	9 sec
트래픽 발생율	3 Mbps, 2Mbps
패킷의 크기	1000 bytes
실시간 서비스 예약율	3 Mbps
링크 대역폭	10 Mbps
혼잡 링크의 수	1, 2, 3, 4, 5



[그림 4] 혼잡링크 수에 따른 실시간 트래픽의 대역폭 할당

[그림 2]에서 실시간 서비스는 3 Mbps의 대역폭을 예약하고 3 Mbps의 트래픽을 발생시키며, 최선형 서비스는 2 Mbps의 트래픽을 발생시킨다. 따라서 전체 트래픽은 링크 용량인 10 Mbps를 초과하게 된다. 하지만 [그림 4]의 결과를 보면 제안된 방식의 네트워크에서는 링크 대역폭을 초과한 혼잡 링크의 수를 1에서 5까지 증가시키더라도 예약한 대역폭인 3 Mbps 혹은 거의 근사치의 대역폭을 할당 받는 것을 볼 수 있다. 따라서 실시간 서비스 트래픽이 요구하는 QoS를 보장해 줄 수 있음을 확인하였다.

V. 결론

본 논문에서는, 네트워크의 코어에서 각 플로우에 대한 정보를 관리하지 않으면서도 플로우의 정보를 관리할 때 얻을 수 있는 이점을 가지도록 SCORE 네트워크를 구현함으로써 네트워크에 확장성을 증가시켰다. 또한 이러한 네트워크에 GPS 스케줄링 방식중 WF²Q+를 구현함으로써 실시간 서비스 요구에 대해 적절히 지원할 수 있음을 시뮬레이션을 통하여 보였다. 향후 연구 과제는 제안된 네트워크를 테스트 베드로 구축하여 다양한 환경하에서 실제 성능 분석을 수행하는 것과 차별화 서비스 네트워크 등과의 연동에 대한 연구가 필요하다.

참고문헌

- [1] Ion Stoica, Scott Shenker, Hui Zhang, "Core-stateless Fair Queuing: A Scalable Architecture to Approximate Fair Bandwidth Allocations in High Speed Networks", SIGCOMM '98.
- [2] J.C.R. Bennett and Hui Zhang, "WF2Q: Worst-case fair weighted fair queuing", In Proceedings of IEEE INFOCOM'96.
- [3] R. Braden, D. Clark and S. Shenker, "Integrated services in the internet architecture: an overview", Internet RFC 1633, June 1994
- [4] S. Blake, et al, "An Architecture for Differentiated Services", Internet RFC 2475, December 1998.