

# 네트워크의 대역폭 응답특성 모델링 및 이를 이용한 트래픽 제어 시스템의 시뮬레이션

박종진\*, 박상진\*, 허의남\*\*, 노민기\*\*\*, 문영성\*

\*숭실대학교 컴퓨터학부

\*\*삼육대학교 컴퓨터학과

\*\*\*한국과학기술정보연구원

e-mail : spel@sunny.soongsil.ac.kr

## The Modeling for Response Characteristics of Bandwidth and the Simulation of Traffic Control System

Jongjin Park\*, Sangjin Park\*, Euinam Huh\*\*, Minki Noh\*\*\*, Youngsong Mun\*

\*School of Computing, Soongsil University

\*\*Dept. of Computer Science, Sahmyook University

\*\*\*Korea Institute of Science and Technology Information

### 요 약

본 연구에서는 네트워크에서 QoS를 지원하기 위해 사전에 검토할 수 있는 성능 예측을 위한 시스템을 개발하기 위해서 네트워크의 대역폭 응답 특성을 연구하여 동적인 모델로 구성한다. 그리고 이를 이용하여 네트워크의 환경변화에 따라 적응적인 트래픽 피드백 제어 시스템을 네트워크에 적용하여 시뮬레이션을 수행한 후 모델의 타당성을 검토하고 실제 피드백 제어 시스템을 구성하기 위한 기초 자료를 제공한다

### 1. 서론

최근 보편화된 초고속 인터넷을 기반으로 한 각종 인터넷 서비스가 활성화되어 있다. 특히 전자상거래, 주식거래, 동화회활동 및 각종 홈페이지를 활용한 홍보 및 정보교환이 인터넷의 보급으로 널리 이용되고 있다. 그러나 최근 추가적으로 요구되고 있는 이슈로서 보안, 이동환경의 지원 및 차별화된 품질(QoS; Quality of Service) 등이 있으며 이를 지원하는 많은 연구개발이 진행중인 상황이다. 인터넷의 표준화 기구인 IETF[1]에서는 기존의 인터넷이나 차세대 인터넷을 위한 QoS를 제공하는 방법에 대하여 표준화 작업을 진행했으며 또한 계속 진행중이다. 특히 IntServ (Integrated Services) 모델[2], DiffServ (Differentiated Services) 모델[3] 등은 IETF에서 제안된 대표적인 인터넷에서의 QoS를 지원하는 방안

이다.

일반적으로 QoS라고 하는 것은 컴퓨터시스템의 자원을 확보하는 것을 의미하며, 즉시성(timeliness), 데이터의 신뢰성 및 보안과 관련된 요구사항으로 구분할 수 있으며, 특히 네트워크에 있어서는 대역폭(bandwidth)의 확보와 일정한 지연시간(delay)의 유지가 주 요구사항이다.

보통 컴퓨터에서 처리되는 어플리케이션들은 시스템의 자원을 공유하면서 동작하고 있다. 그러므로 특정 어플리케이션이나 특정 대상과 관련하여 QoS를 지원하기 위해서는 우선순위에 의해 자원을 할당하는 방식이 일반적이다. 그러나 대부분의 방식이 네트워크의 트래픽과 같은 환경변화에 관계없이 일정한 방식으로 동작한다.

본 연구에서는 QoS 지원과 관련하여 네트워크의 대역폭 특성을 살펴보고 네트워크의 환경변화에 따라

적응적인 QoS 지원을 위해 필요한 모델을 개발하고자 한다.

## 2. 대역폭 응답 특성

### 2.1 대역폭 지정에 의한 응답 분석

리눅스에서 제공되는 트래픽 제어 모듈인 tc (linux kernel 2.2부터 지원)[4]를 사용하여 특정 컴퓨터로 전송되는 데이터의 대역폭을 조절하여 그 응답(전송률)을 살펴본다. 그림1은 대역폭을 실험 시작시 1Mbps로 지정하고 60초 후 5Mbps로 지정하여 총 100초 동안의 단위시간(1초)당 데이터 전송량을 그래프로 표시한 것이다.

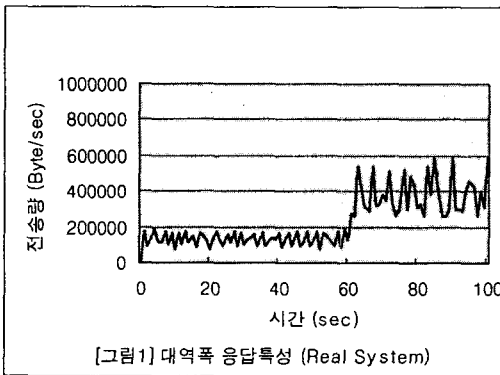


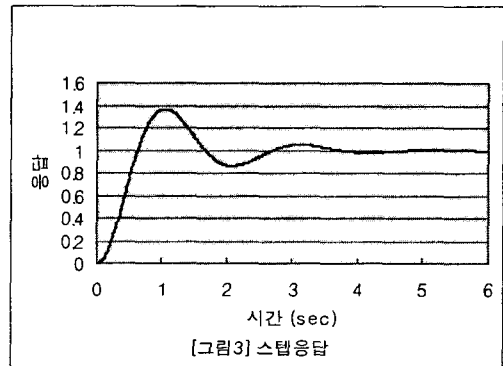
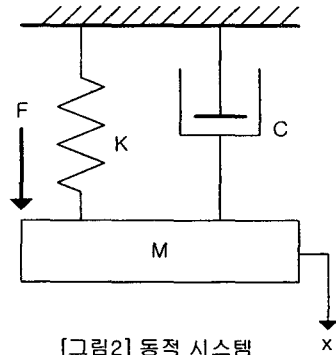
그림1을 살펴보면 1Mbps를 지정할 경우 실제 시스템의 응답이 평균 약 0.99Mbps의 응답을 보여준다. 그러나 5Mbps를 지정하였을 경우에는 평균 약 3.00Mbps의 응답을 나타내 실제 지정량과는 많은 차이를 보여준다. 이는 트래픽 제어시스템인 tc 모듈의 경우 클래스(우선순위) 기반의 버퍼제어를 하기 때문에 이더넷(ethernet)망의 트래픽부하에 따라 할당량의 상대적인 크기에 따라 우선순위를 조절하는 결과로 해석된다. 즉, 할당량이 1Mbps로 이더넷망의 최대부하인 10Mbps에 비해 상대적으로 적은 경우 거의 그대로 반영되지 않지만 5Mbps의 경우에는 상대적으로 커 할당량 만큼 반영되지 못하고 있는 것을 결과는 보여준다.

또한 데이터 전송률의 변화폭이 거의 200Kbps에 달하고 있음을 알 수 있다. 이는 네트워크의 전송률이 매우 변동이 심한 동적인 시스템임을 보여준다.

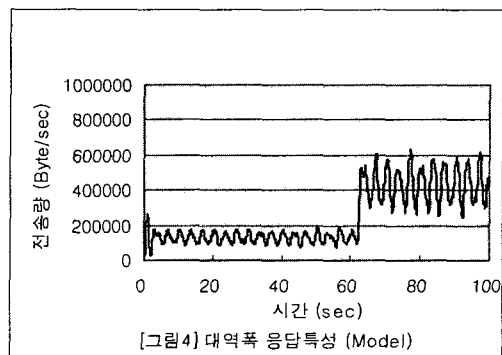
### 2.2 네트워크의 대역폭 지정에 따른 응답특성의 모델링

그림1에서 살펴본 바와 같이 대역폭 지정에 따른 네트워크의 응답특성이 동적인 시스템임을 알 수 있다. 고전적인 제어시스템에서 동적인 시스템은 일반적으로 식1과 같이 2차 미분방정식으로 표현되고 시스템은 그림2와 같이 표현한다. 또한 그 응답 특성은 그림3과 같이 나타난다.

$$M \frac{d^2x}{dt^2} + C \frac{dx}{dt} + Kx = F(t) \quad (\text{식 1})$$



그러나 그림3의 결과에서는 실제 상황에서 전송률을 측정할 결과인 그림1과 다른 형태임을 알 수 있다. 대체적인 추세는 일치하지만 그림1과 같은 변동을 보여주기 위해서는 식1에서 왜란(noise) 효과를 추가해야 한다. 그림1에서 나타난 왜란이 어느 정도 주기적임을 알 수 있으므로 왜란을 주기함수로 적용하여 그림1을 구한 것과 같은 조건으로 네트워크의 대역폭을 지정 후 모델의 응답특성으로 구한 데이터 전송률은 그림4와 같이 나타난다.



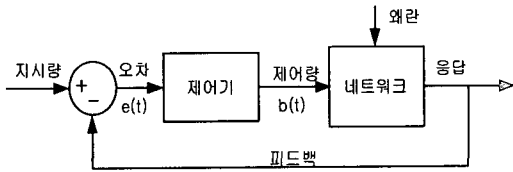
여기서 그림1과 그림4를 비교해보면 그 결과가 상당히 유사함을 알 수 있다. 즉, 네트워크의 대역폭 지정에 따른 응답 특성인 데이터 전송률은 고전적인 동적 모델링으로 간단히 상사시킬 수 있음을 알 수 있다. 그러므로 이 모델을 사용하면 네트워크의 트래픽과 관련된 여러 기법을 다양한 조건으로 검토할 수 있다.

### 3. 네트워크의 피드백 제어 시뮬레이션

#### 3.1 피드백 제어 시스템의 설계

리눅스의 트래픽 제어 모듈인 tc를 사용한 네트워크의 대역폭 지정에 의한 QoS 지원 메커니즘은 열린 루프 제어 시스템(open loop control system)이다. 즉, 지정 후 네트워크의 상황 변화에 대해서 연속적으로 제어량을 조절할 수 있는 기능은 없는 것이다. 이에 반하여 제어결과를 적극적으로 반영하는 메커니즘을 달린 루프 제어 시스템(closed loop control system)이라고 하며 그 핵심은 응답 상태의 피드백(feedback) 구조에 있다.

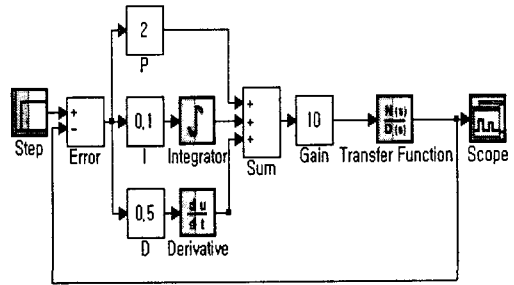
네트워크 환경에 있어서 가장 일반적인 응답 상태 변수 즉 메트릭(metric)으로는 전송률(throughput), 지연 시간(delay time), 응답시간(response time) 및 손실률(loss rate) 등이 있다. 본 연구에서는 제어를 수행하는 해당 시스템에서 가장 측정이 간편한 전송률을 유일한 피드백 상태 변수로 선택하였으며, 이때 피드백 제어 시스템은 그림5와 같이 구성하였다.



[그림 5] 피드백 제어 시스템

본 연구에서 채택한 피드백 제어 시스템의 제어기 구조로는 고전적인 제어 시스템에서 가장 일반적인 PID 제어 방식을 선택하였다[5]. 여기서 P는 비례제어(proportional control), I는 적분(integrate control), D는 미분(derivative control)을 의미한다. 비례(P) 제어 부분의 역할은 목표값과 실제값과의 차이인 오차에 비례하는 제어를 수행하며 가장 기본적인 제어량이다. 적분(I) 제어 부분의 역할은 비례제어를 수행할 때 발생할 수 있는 정상상태 오차를 제거하는 기능을 하며, 미분(D) 제어 부분은 잘 활용하면 예측 기능이 있어 응답속도를 빠르게 하는 역할을 한다. 하지만 잡음(왜란)이 있는 경우 도리어 안정성에 나쁜 영향을 미치기도 한다.

본 연구에서는 PID 제어기의 구성을 그림6과 같이 설계하고, 제어시스템 해석 도구인 CEMTOOL[6]로 각 제어기의 상수를 결정하였다.

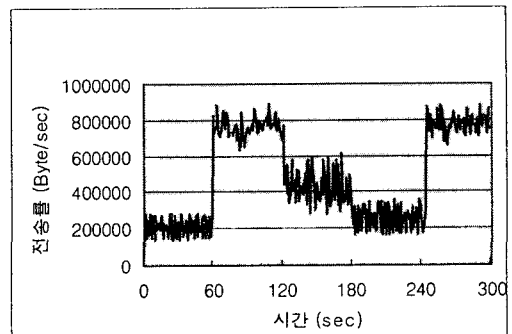


[그림 6] 제어기의 구성

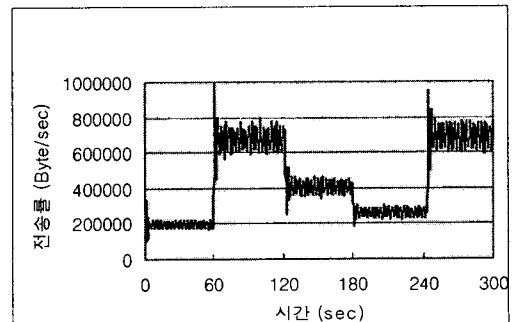
#### 3.2 시뮬레이션

2.2절에서 구해낸 네트워크 응답 특성 모델에 대역폭 지정을 실험 시작과 동시에 매 분마다 2Mbps, 8Mbps, 5Mbps, 3Mbps, 7Mbps로 지정하였다. 이에 따른 실제 시스템 응답인 전송률의 측정 결과는 그림7과 같다.

또한 응답 특성 모델을 입력을 앞에서 지정한 실제 실험과 같이 설정하여 응답을 계산한 그래프가 그림8이다. 이는 그림1과 그림4의 경우와 마찬가지로 유사한 결과를 보여준다. 그러나 정상 상태의 변화율에서는 약간의 차이를 보이고 있다.



[그림 7] 대역폭 응답특성 (Real System)



[그림 8] 대역폭 응답특성 (Model)

피드백 제어를 네트워크의 QoS를 지원하기 위해서 사용하려면 그림6의 전달함수(transfer function)를 대신하여 응답 특성 모델을 대입하고, 입력에 해당하는 스텝함수 부분 대신에 앞에서 지정한 실제 실험과 같이 설정하여 시뮬레이션을 수행하면 그 결과가 그림9와 같이 나타난다.

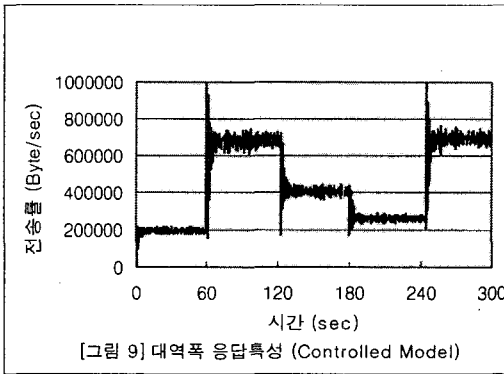


그림9를 살펴보면 피드백 제어기능이 없이 단순히 대역폭의 지정에 의한 모델 응답인 그림8과 유사하게 보인다. 그러나 자세히 검토해 보면 정상상태 오차가 많이 감소한 것을 볼 수 있으며, 응답 시간도 감소했음을 알 수 있어 제어 기능이 제대로 동작하고 있음을 알 수 있다.

그러나 제어기능의 수행으로 인한 상태변화가 훨씬 빈번해 졌음을 또한 알 수 있다. 이는 시스템의 부하가 늘어난다는 것을 의미하므로 바람직하다고는 할 수 없다.

#### 4. 결론

본 연구는 네트워크 성능 중 하나인 전송률을 동적 시스템으로 모델링하여 네트워크의 QoS를 지원하기 위한 제어기법들을 사전에 시뮬레이션을 통해 성능을 평가할 수 있는 대상으로 개발하는 것을 목표로 하였다.

리눅스에서 제공하는 트래픽 제어 모듈인 tc를 사용하여 대역폭을 실제로 할당한 후 그 응답으로 전송률을 실시간으로 측정하였다. 그리고 이 결과를 그대로 재현할 수 있도록 제어시스템 해석 도구인 CEMTOOL로 동적 시스템 모델의 파라미터를 결정하였다. 이렇게 만들어진 모델과 실제 시스템인 네트워크의 전송률과 비교함으로써 모델의 타당성을 확인하였다.

그러나 실험에 사용한 방식이 TCP를 사용한 트래픽이기 때문에 전송률의 변화가 많았지만 UDP를 사용한 데이터 전송률의 경우에는 다른 결과가 예상된다. 그러므로 추후 모델링에 TCP/UDP 모두 지원하도록 모델을 확장할 필요가 있다.

또한 지속적이고 안정적인 QoS 지원을 위해 가장 전통적인 PID 제어기법을 네트워크의 트래픽 제어에 응용하기 위해 앞에서 구한 동적 모델을 대상으로 하

여 시뮬레이션을 수행하여 트래픽 관리에 피드백 제어가 유용함을 알아보았다. 추후 이 제어기법을 실제 네트워크에 적용하여 그 결과를 비교 검토할 것이다.

TCP 트래픽의 경우 제어결과는 상당히 안정된 상태를 보이고 있어 피드백제어의 효과가 있음을 알 수 있으나 트래픽의 변동주기가 훨씬 높아지는 현상도 나타난다. 이는 실제로 제어신호의 발생횟수가 증가함을 의미하며 시스템의 부하가 증가한다는 것을 의미한다. 그러므로 PID 제어기법에 제어횟수를 감소시키는 기법의 연구가 필요하다고 판단된다. 그리고 차후 피드백 상태변수를 전송률뿐만 아니라 지연시간(응답 시간)이나 손실률 등도 추가하여 연구를 진행할 계획이다.

#### 5. Acknowledgement

본 연구는 한국과학기술정보연구원(KISTI)의 '국가 그리드 기반 구축' 관련 위탁연구과제의 일환으로 수행되었음.

#### 참고문헌

- [1] <http://www.ietf.org>, IETF Homepage.
- [2] R. Barden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: An Overview," RFC 1633, June 1994.
- [3] S. Brake, et. al., "An Architecture for Differentiated Services," RFC 2475, Dec. 1998.
- [4] <http://snafu.freedom.org/linux2.2/iproute-notes.html>
- [5] <http://www.engin.umich.edu/group/ctm/PID/PID.html>
- [6] <http://www.cemtool.co.kr>