

# Initial TCP window size의 변경에 따른 성능분석

김정훈<sup>1</sup>, 이계상<sup>2</sup>, 김창규<sup>3</sup>

<sup>1</sup>동의대학교 산업기술대학원 정보통신공학과

<sup>2</sup>동의대학교 전자통신공학과

## Performance analysis of changing initial TCP window size

Jung-hun Kim<sup>1</sup> · Kye-sang Lee<sup>2</sup> · Chang-kyu Kim<sup>3</sup>

<sup>1</sup>Dept. of Information Communication Engineering, The Graduate School of Industrial Technology,  
Donggeui University

<sup>2</sup>Dept. of Electronic Communications Engineering, Donggeui University

<sup>3</sup>E-mail : junghun@hyomin.donggeui.ac.kr

### 요 약

현재 인터넷상의 QoS(Quality of Service) 향상을 위해서 여러 가지 방법과 기술들이 소개되고 있다. 이러한 QoS 향상을 위해 트래픽 제어기법 과 IS(Integrated Services), DS(Differentiated Services), Tag Switching, MPLS(Multi Protocol Label Switching) 등과 같은 방법으로 인터넷에서 보다 나은 서비스를 지원하려는 노력이 시도되고 있다. 본 논문에서는 특히 TCP connection에서의 initial window size의 변경이 네트워크의 QoS, 즉 link의 utilization과 packet의 drop, performance에 어떠한 영향을 주는지를 살펴본다. 그리고, 이에 대한 특성을 분석하고 개선점을 현재의 인터넷에서 가장 많이 사용되어지고 있는 FTP와 web에 대한 모델을 사용하여 가상적으로 시뮬레이션을 해본다.

### I. 서 론

단순하게 트래픽을 제어하는 것은 네트워크 상에서의 트래픽 흐름을 보완하려는 단순한 트래픽 제어 기법에서 출발한 것이다. 여기에, 기존의 best effort service에 QoS보장을 요구하는 real time service도 지원할 수 있도록 기존의 IP 모델을 확장하여 보완하려는 연구로서 나온 것이 IS과 차세대 IPv6이다. 그리고, 인터넷의 실시간(Real-time) 서비스 요구의 증대에 따라, QoS와 차별화된 서비스 제공이 필요하여졌다. 그래서 간단하면서 자연스러운 진화를 허용한 연구결과가 DS이다. IP packet에 식별자인 Label 혹은 Tag를 붙여서 ATM switch처럼 router에서 고속으로 IP 패킷을 switching하는 기술로는 Tag Switching, MPLS 등이 있다.

이러한 모든 기법들은 QoS의 향상을 위해 노력한 결과들이다.

### II. 본 론

#### 2.1 TCP의 Window

인터넷상의 모든 컴퓨터들이 동일한 속도로 동

작하고 있지 않다. 수신측의 컴퓨터가 자료를 받아들이는 것 보다 더 빠르게 송신측의 컴퓨터에서 네트워크를 통해 자료를 송신할 경우 자료 범람이 발생한다. 이는 결과적으로 자료의 손실을 유발한다.

그렇기 때문에 몇 가지 기술이 자료 범람을 처리하기 위해 이용되고 있다. 이 기술들은 흐름 제어(flow control) 방식으로 알려져 있다. 흐름 제어의 가장 간단한 방식으로는 각 패킷을 전송한 후에 송신자가 기다리는 stop-and-go 시스템이 있다. 수신자가 이미 다른 패킷을 받을 준비가 되었다면, 수신자는 보통 용낙의 형식인 제어 메시지를 송신자에게 보낸다.

그리고, 높은 처리율을 얻기 위해 프로토콜은 이동창(sliding window)이라고 알려진 흐름제어 기술을 사용한다. 이는 송신자와 수신자가 용낙이 도착되기 전에 보내질 수 있는 자료의 최대양인 창 크기를 사용하도록 프로그램된다.

그림 1은 이동창의 개념을 보이고 있다.

이동창은 극적으로 처리율을 증진시킬 수 있다. 그 이유를 이해하기 위해 stop-and-go 방법과 이동창 방법을 이용한 일련의 전송과정을 비교해 본다. 그림 2는 4개의 패킷 전송을 위한 비교를 보이고 있다.

그림 2의 (a)는 stop-and-go 에 대한 일련의 전

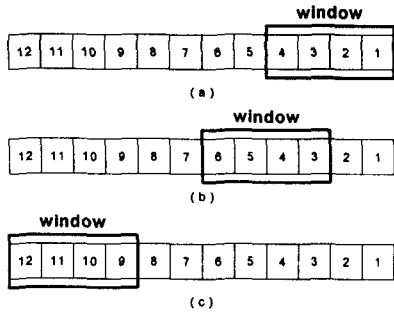


그림 1. 이동창 개념

송을 보여준다. 패킷이 보내진 후, 다른 패킷을 보내기 전에 프로토콜은 응답을 기다린다. 만약 네트워크를 통해 하나의 패킷이 보내지는데 요구되는 지연이  $N$ 이라면, 4개의 패킷을 보내는데 요구되는 총시간은  $8N$ 이다.

그림 2의 (b)는 이동창을 사용 할 때의 일련의 전송을 보여준다. 프로토콜은 기다리기 전에 창내의 모든 패킷을 보낸다. 연속적인 패킷전송사에 작은 지연을 가진다. 비록 지연이 보여지는 것 보다 더 적을 지라도, 전송은 결코 순간적으로 이루어지지 않는다. 하드웨어가 패킷을 전송하고, CPU에 인터럽트를 걸고, 다음 패킷을 전송하기 시작하기 위해 보통 수 밀리초의 시간이 요구된다. 그리하여, 4개의 패킷을 보내는데 요구되는 총 시간은  $2N + \epsilon$  이고, 여기서  $\epsilon$ 는 작은 지연을 표시한다.

이동창의 중요성을 이해하기 위해서, 많은 패킷을 포함하는 확장된 통신을 상상해보자. 이 같은 경우에, 전송을 위해 요구되어지는 총 시간은  $\epsilon$ 를 무시할 만큼 크다. 이동창의 장점을 평가하기 위해, 높은 처리율과 큰 지연을 가진 네트워크를 고려해보자(예: 위성채널). 이 같은 네트워크에서, 이동창 프로토콜은 1보다 큰 비율로 성능이 증대된다. 사실상, 잠재적인 향상은 다음과 같다.

$$T_w = T_g \times W$$

여기서,  $T_w$ 는 이동창을 이용하여 성취할 수 있는 처리율이고,  $T_g$ 는 stop-and-go 프로토콜을 이용하여 성취할 수 있는 처리율이다. 이 식은 그림 2의 (a)에서 예를 든 이동창 프로토콜이 왜 그림 2의 (b)에서의 stop-and-go 프로토콜의 처리율의 약 4배가 되는지를 설명한다. 물론, 창 크기를 증가하는 것만으로 처리율을 무한히 크게 만들 수는 없다. 왜냐하면 네트워크의 대역폭이 그 상한값을 제한하기 때문이다. 즉, 비트들은 그들이 전송되는 하드웨어의 대역폭보다 빠르게

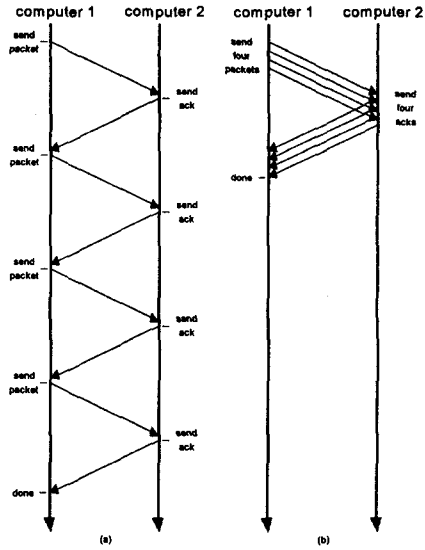


그림 2. stop-and-go와 이동창의 비교

보낼 수 없다. 따라서, 식은 다음과 같이 다시 쓰여질 수 있다.

$$T_w = \min(B, T_g \times W)$$

여기서,  $B$ 는 하드웨어 대역폭이다.[5]

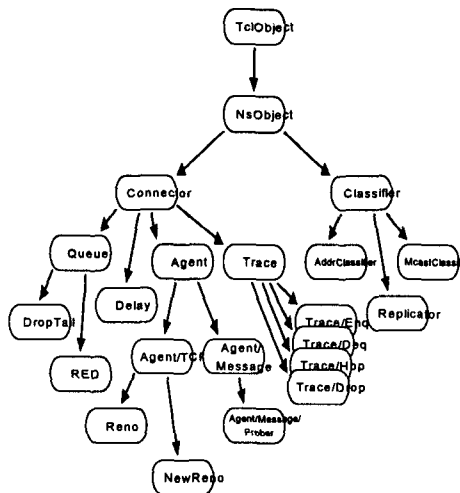


그림 3. NSv2 - Class Hierarchy

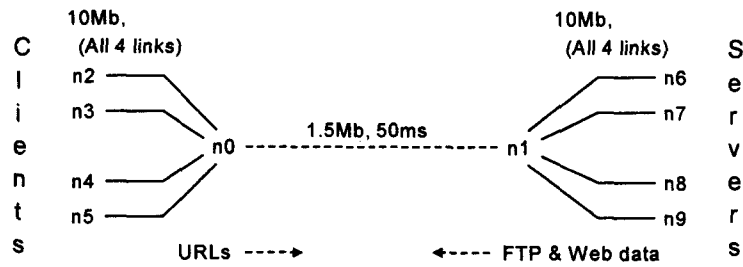


그림 4. Network Topology Model

## 2.2 NS (Network Simulator)

NS는 networking research에서 추상적인 이벤트 simulator를 목적으로 만든 것으로 TCP, routing 그리고 multicast protocols의 simulation을 지원한다.

NS는 1989년 REAL network simulator의 변형으로 시작되어, 지난 몇 년간 발전하였다. NS의 발달은 VINT(Virtual InterNetwork Testbed) project와 함께 하는 공동 제작품으로 나아가고 있다. 최근에 바뀐 버전은 version 2.1b3이다.[1]

시뮬레이션의 특징은 OTcl(Object-oriented extension to Tcl)과 C++의 이중구조로 이루어져 있다. 게다가 Data와 Control이 분리된 구조와 module화된 접근법을 취하고 있다. 그림 3은 NSv2의 Class Hierarchy를 보이고 있다. 이와 함께 NAM(Network Animator)을 이용하여 결과를 그림으로 나타낼 수 있다.

VINT project는 USC(University of Southern California)/ISI(Information Sciences Institute), Xerox PARC(Pale Alto Research Center), LBNL(Lawrence Berkeley National Laboratory) 그리고 UCB(University of California Berkeley)의 공동연구로 진행되고 있다.

## 2.3 Network Topology Model

이 논문은 TCP의 initial window size 증가 효과를 다룬 시뮬레이션 연구이다. 여기에는 long-lived TCP 연결(file transfer)과 short-lived web-browsing style의 연결을 model로 했다.

그림 4은 본 논문에서 사용한 network topology를 보여주고 있다. 그림과 같이 client를 좌측 node로 4개 연결하고, FTP와 Web server로 우측 node로 4개 연결하였다. 그리고, 그들 사이의 연결 link로서 bottleneck을 가진 10Mbps로 구성되어 있다. 모든 link는 양방향으로 구성되어 있지만, ACKs, SYNs, FINs, URLs는 왼쪽에서 오른쪽으로 흐른다.

시뮬레이션의 모든 FTP는 1MB file를 전송하며, 모든 Web page는 3개의 URLs를 포함한다고 가정한다. 여기에서 Web client는 1초에서 5초

사이의 임의의 delay후 새로운 page를 요구한다.

최고 TCP window는 11 packet로 정하고, 최고 packet/segment size는 1460 byte, buffer size는 25 packet으로 정했다.

시뮬레이션은 8, 16, 32 Web client와 0에서 3까지의 범위를 가지는 FTP client로 실행되고, initial window size는 1에서 4까지 변화한다.

## III. 결론 및 검토

이 시뮬레이션에서는, link의 congestion을 변화시킬 목적으로 file transfer client의 수를 조절해 본다. FTP client는 계속해서 1Mbyte의 파일 전송을 요구하기 때문에 initial window의 영향보다는 congestion의 영향을 더 받게 된다. 동일한 initial window를 이용하는 연결에서 1Mbyte의 file transfer에서는 initial window상에서의 변화의 효과는 찾을 수 없기 때문에, Web browsing 연결에 초점을 둔다.

실험에서의 bottleneck link utilization과 함께 packet의 drop rate가 initial window size와 함께 증가할 것으로 예상된다. 즉, 큰 initial window size에서는 throughput의 증가와 delay 감소가 예상되지만, 그와 함께 packet의 drop rate는 증가할 것이다. 하지만, drop rate가 performance의 지표가 되지 못한다고 본다.

initial window size의 증가가 무조건적인 network performance의 증가를 가져오지는 않을 것이다. 단순한 initial window size의 증가보다는 현재의 network의 throughput과 performance의 고려가 무엇보다 필요할 것이다.

특히, FTP의 연결보다는 Web에서의 initial window size의 조절이 network performance의 중요관건이 될 것이다. 전체 네트워크의 link utilization에 따른 적당한 packet size를 선택하지 않으면 congestion과 packet의 drop으로 손실을 가져 올 것이다.

packet에 대한 initial window size 증가라는 결과로부터 performance의 개선에 도움이 될 것

이다.

### 참고문헌

- [1] <http://www-mash.cs.berkeley.edu/ns/ns.html>
- [2] James martin, "Local Area Networks", J., pp.433~470
- [3] Sally Floyd, "NS Version 1 Simulator Tests for Class-Based Queueing", J., April 28, 1997
- [4] K. Poduri, "Simulation Studies of Increased Initial TCP Window Size", J., February, 1998
- [5] Douglas E. Comer, "Computer Network and Internet", Prentice-Hall, 1997
- [6] B. Mah, "An Empirical Model of HTTP Network Traffic", Proceedings of INFOCOM '97, Kobe, Japan, April 7-11, 1997.
- [7] C.R. Cunha, A. Bestavros, M.E. Crovella, "Characteristics of WWW Client-based Traces", Boston University Computer Science Technical Report BU-CS-95-010, July 18, 1995.
- [8] K.M. Nichols and M. Laubach, "Tiers of Service for Data Access in a HFC Architecture", Proceedings of SCTE Convergence Conference, January, 1997.
- [9] K.M. Nichols, "Improving Network Simulation with Feedback", available from [knichols@baynetworks.com](mailto:knichols@baynetworks.com)
- [10] <http://netweb.usc.edu/vint>