

---

# 인터넷 통신에서 고속 TCP 프로토타입을 이용한 지연 제어

나하선\* · 김광준\*\* · 나상동\*\*\*

Delay Control using Fast TCP Prototype in Internet Communication

Ha-Sun Na\* · Gwang-Jun Kim\*\* · Sang-Dong Ra\*\*\*

---

이 논문은 2003년도 조선대학교 교내학술 연구비를 지원받아 연구되었음.

---

## 요 약

본 논문에서는 고속 프로토타입 TCP 트래픽 제어 프레임워크를 TCP 기반의 신뢰할 수 있는 전송 및 윈도우 기반 혼잡제어로 확대한다. TCP의 소용 대역폭 응답의 적극성을 LTS(large time scale) 네트워크 상태의 함수 형태, 즉 RTT(round-trip time)가 결정된 피드백 루프의 한계를 넘어서는 정보의 형태로 조정하는 LTS 모듈과 TCP를 연계시키는 방법으로 수행한다. 특히, TCP 트래픽의 ACK를 지연시키는 메커니즘을 제시하고, 제어된 ACK 흐름은 TCP 활동의 트래픽 특성을 비교 평가한다.

링크 사용 또는 버퍼 점유에 대한 정보를 사용함으로써 혼잡 제어에 대한 ACK 지연을 통보할 수 있고, ACK 지연 메커니즘을 다양한 네트워크 노드에서 구현된다. 본 논문에서는 이러한 혼잡 제어를 통한 연결의 공정성 뿐 아니라 고속 TCP 프로토타입이 패킷 손실을 보다 효과적으로 줄이며, 시뮬레이션 결과를 통해 요구된 버퍼 크기에 대한 처리량이 뛰어남을 입증하였다.

## ABSTRACT

Measurements of network traffic have shown that self-similarity is a ubiquitous phenomenon spanning across diverse network environments. We have advance the framework of multiple time scale congestion control and show its effectiveness at enhancing performance for fast TCP prototype control. In this paper, we extend the fast TCP prototype control framework to window-based congestion control, in particular, TCP. This is performed by interfacing TCP with a large time scale control module which adjusts the aggressiveness of bandwidth consumption behavior exhibited by TCP as a function of "large time scale" network state. i.e., conformation that exceeds the horizon of the feedback loop as determined by RTT. Performance evaluation of fast TCP prototype is facilitated by a simulation bench-mark environment which is based on physical modeling of self-similar traffic. We explicate out methodology for discerning and evaluating the impact of changes in transport protocols in the protocol stack under self-similar traffic conditions. We discuss issues arising in comparative performance evaluation under heavy-tailed workload.

## 키워드

Fast TCP Prototype, Self-Similarity, Transport Protocol, Feedback Loop

---

\* Dept. of electrical and computer Eng. Univ. of Colorado at Boulder \*\* 여수대학교 컴퓨터공학과  
접수일자 : 2003. 10. 8

\*\*\*조선대학교 컴퓨터공학과

### 1. 서론

TCP(Transmission Control Protocol)는 최근 몇 년 동안 TCP/IP 인터넷의 폭발적인 성장에 힘입어 가장 널리 사용되는 전송 계층 프로토콜이 되었다. 혼잡 제어 및 복구 수행에 사용되는 알고리즘들은 TCP의 주요 구성요소에 해당한다[1,2].

전송된 TCP 세그먼트는 반대 방향 연결의 ACK(acknowledgment)와 동일한 물리적 경로를 사용하며, 또한 패킷과 ACK는 네트워크 스위치/라우터 뿐 아니라 최종 시스템에서도 버퍼를 함께 공유한다. 이러한 공유는 한 연결의 여러 ACK가 한꺼번에 스스로 몰려들기 때문에 ACK 압축(ACK compression)이라는 효과를 발생시킨다[6,7]. ACK 압축은 최종 시스템 뿐 아니라 스위치/라우터에서도 발생하기 때문에 어떠한 경우라도 클라이언트로 향하는 ACK의 흐름에 혼잡이 발생하며, 해당 TCP 연결 처리량(throughput)이 감소하게 된다.

무선 네트워크의 트래픽 제어에서 ACK 압축 효과로 인한 전송 프로토콜의 동적 특성은 네트워크 시뮬레이터를 응용한 시뮬레이션을 이용하여 트래픽 제어에서 라우터의 대기열 양이 주기적인 응답을 나타낸다.

본 논문은 TCP 트래픽 지연 제어에 중점을 두어 안정적인 전송율, 예측 가능한 지연, 순서에 따른 지연을 특징으로 하는 ATM 네트워크에서 TCP를 운용하는 경우에도 동일하게 적용하였다.

또한 ATM 네트워크의 전송율 제어 네트워크를 기반으로 하는 고속 프로토타입 TCP를 적용함으로써 트래픽 제어의 성능을 분석함으로써 최종 시스템의 세그먼트 및 ACK가 공통 대기열 양을 공유할 경우, TCP ACK 압축 효과 및 그로 인한 일정 시간 내에 처리되는 대기열 양의 손실이 감소함을 시뮬레이션 결과를 통해 입증하였다. 이러한 공통 대기열 양은 IP 계층 또는 TCP 연결의 데이터 세그먼트가 혼잡 노드의 ACK와 가상 채널을 공유하는 ATM 계층에 존재하며, 혼잡 노드에서 각 방향마다 하나의 TCP 연결이 설정되는 단순한 구성에서 연결의 주기적인 응답이 나타나며, 연결 중 최소한 하나 이상은 일정 시간 내에

처리되는 대기열 양은 감소된다. 또한, 일정 시간 내에 처리하는 대기열 양의 감소는 TCP 상호 연결의 상대적 전송 지연과는 상관이 없으며, 다만 윈도우 크기, 전송율 및 네트워크 지연으로부터 영향을 받는다.

### II. 트래픽 환경에서 TCP의 동적 특성

이 장에서는 성능 분석 및 시뮬레이션의 전제가 되는 최종 시스템 및 네트워크 모델링을 소개하고, 트래픽 제어 TCP 환경에서 ACK이 발생하는 과정에 대해 설명하며, 이러한 응답이 TCP의 일정 시간 내에 처리되는 대기열 양에 미치는 영향을 정량화함으로써 성능을 분석한다.

#### 2.1 네트워크 모델

그림 1에서는 포인트 투 포인트 방식의 전용 링크를 통해 노드가 상호 연결되어 있고, 각 링크는 실제 네트워크에서의 VC를 의미한다.

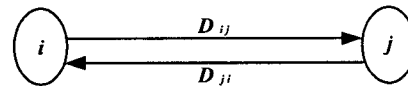


그림 1. 양방향 트래픽의 단순화된 모델  
Fig.1 Two-way traffic Simple Model

노드  $i$  에서 노드  $j$  로 데이터를 전송하는 TCP 연결을 연결  $i$  로, 반대 방향의 연결은 연결  $j$  로 설명하며, 노드  $i$  에서 노드  $j$  로 전송되는 TCP 세그먼트가 인식하는 일정한 지연을  $D_{ij}$ 로서 또한,  $D_{ji}$ 는 노드  $j$  에서 노드  $i$  로 전송되는 세그먼트가 인식하는 지연을 나타내며, 전송율은 TCP 세그먼트 수/초 단위를 사용한다.

그림 2에서 순방향 연결의 세그먼트 및 역방향 연결의 ACK는 발신 링크의 전송율 수준에서 서비스되는 IP 계층의 공통 FIFO 대기열을 공유한다. 이 공통 대기열이 ACK 압축이 발생하는 데 핵심적인 역할을 한다. 즉, 역방향 연결의 ACK들이 순방향 연결의 데이터 세그먼트 뒤에서 대기하면서 한꺼번에 물리적으로 TCP 동적 특성을 유발하게 된다. 그러나 IP 서비스 속도가 링크의 전송율과 동일하다는 실제

하에 역방향 연결의 ACK와 순방향 연결의 데이터 세그먼트가 공통의 ATM VC를 공유한다면 한꺼번에 혼잡이 발생하는 것을 방지할 수 있다. IP 대기열의 서비스 속도가 높아지면 IP 계층의 대기열이 감소하지만, 데이터가 대기하는 ATM 즉 적응 계층에서는 동일한 데이터 세그먼트의 혼잡이 발생한다.

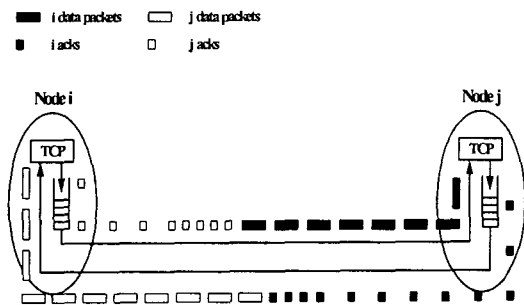


그림 2 양방향 트래픽 구성의 트래픽 패턴  
Fig.2 Traffic pattern of two-way traffic configuration

그림 3은 최종 노드의 모델을 나타내는 것으로서 TCP 및 IP 프로토콜은 전송을 제어 ATM 계층에서 운영하며, 적절한 분할 및 재조합, 적응 및 전송을 제어 절차가 IP 계층에서 구현된다.

두 개의 애플리케이션이 각 노드 내에 상주하는데, 하나는 해당 노드에서 시작하는 TCP 연결에 대해 데이터 소스 역할을 하는 송신 응용이고, 다른 하나는 반대쪽 노드로부터 도착하는 데이터를 수용하는 수신 응용이다. 송신 응용은 과중하고, 수신 응용은 모든 수신 데이터를 즉시 수용한다고 가정한다.

하나의 공통 처리가 노드 내부의 모든 TCP 프로세싱을 처리하며, TCP 처리는 네트워크로부터 데이터 세그먼트를 수신하고, 이를 수신 응용으로 전달한다. 또한 하나의 ACK가 생성되고 각 수신 세그먼트의 송신 대기 열에 추가된다. 동일한 TCP 프로세스가 상대편 노드로의 데이터 전송도 처리하는데, 즉 상대편 노드로부터 ACK가 수신될 때마다 송신 응용의 데이터 세그먼트 1개 이상을 송신 대기 열에 추가시킨다. 마지막으로 이 TCP 처리는 느린 구동 및 혼잡 회피 단계에서 송신 TCP의 윈도우 증가를 제어할 책임이 있는데, 즉 윈도우가 최대 크기에 도달할 때까지, ACK가 수신될 때마다 윈도우를 증가시킨다.

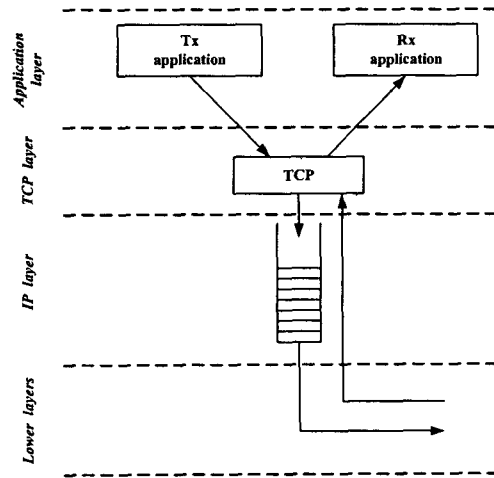


그림 3. 분석을 위한 최종 노드 모델  
Fig. 3 End node model for analysis

### III. 고속 TCP 트래픽 제어

TCP는 TCP 발신자만이 ACK 정보를 통해 네트워크 혼잡 여부를 알 수 있다는 것을 기본 전제로 하며, TCP는 오직 ACK 정보만을 근거로 종단간 (end-to-end) 흐름 제어를 사용하는 것이다. 그러나 라우터나 ATM 스위치와 같이 새로 등장한 네트워크 요소들은 혼잡 초기에 이를 감지할 수 있는 보다 발전된 메커니즘을 갖게 된다.

고속 TCP 제안은 TCP 트래픽 제어 시 ACK를 지연시키는 방법이다. TCP 송신자나 수신자의 구현을 전혀 변경시키지 않고, 단지 정체된 중간 노드에서 ACK를 지연시킴으로써 중간 노드(예: 라우터)로 하여금 TCP 루프 상에서 몇 가지 명시적 제어를 실행하게 한다. 고속 TCP는 ICMP(internet control message protocol) 소스 억제 메시지[1] 및 ECN (explicit congestion notification)[2]와 같은 다른 기법으로 TCP 송신자 구현을 수정하지 않고 ACK 지연 방식을 사용함으로써, TCP 트래픽의 형태에 간접적으로 영향을 미친다.

그림 4는 라우터에 구현된 고속 TCP의 프로토타입을 나타낸 알고리즘으로서 혼잡 감지, ACK 확인, ACK 지연 등의 3가지 부분으로 구분한다.

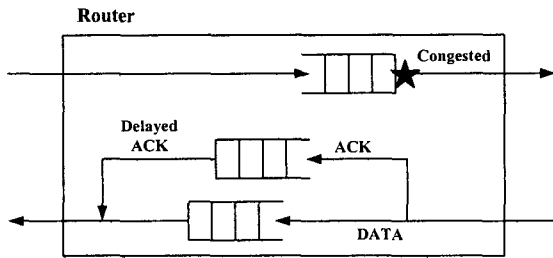


그림 4 고속 TCP의 Prototype  
Fig. 4 Prototype of fast TCP

버퍼 오버플로우가 발생하기 전에 혼잡 감지가 수행됨으로서 ACK 확인 시 ACK 흐름 뒤에 있는 링크는 정상적인 데이터 트래픽으로부터 분리시킬 필요가 있고, 또한 ACK 지연에 따라 ACK가 응답을 배포할 때 속도가 결정될 수 있기 때문에 각각의 부분에 대해 다음과 같이 설명 할 수 있다.

### 3.1 혼잡 감지

일반적으로 라우터는 메시지 저장 및 전송을 기반으로 하여 설계가 되어 큰 버퍼 용량이 필요하므로 대기 열 양 또는 증가 속도를 확인함으로써 혼잡 여부를 감지할 수도 있다.

대기 열 양 또는 증가 속도가 사전 정의된 임계값을 초과 한다면 혼잡으로 인식되므로 몇 가지 트래픽 예측 방식을 적용할 수도 있다. 퍼지나 신경망 이론을 바탕으로 한 동적 예측 기법을 이 부분에 적용하면 정확하게 혼잡을 감지하는 데 크게 도움이 될 수 있다.

### 3.2 ACK 확인

라우터는 ACK를 확인하기 위해 TCP/IP 패킷의 ACK 비트를 먼저 검사하고, ACK 비트가 설정되어 있다면 ACK 패킷이 된다. 그러나 트래픽을 추가함으로써 지연이 허용되지 않은 TCP 세그먼트에 ACK 정보가 수록되어 있는 경우도 있다. 이런 경우, 페이로드가 있는 TCP 세그먼트로부터 ACK 정보를 분리해내야 하므로 페이로드가 있는 TCP 세그먼트에서 ACK 비트를 지우고, 원래의 TCP 세그먼트로부터 ACK 정보를 복사하여 새로운 ACK 패킷으로 생성할 수 있다.

고속 TCP에서 IP 패킷은 모니터링될 수 있고,

ACK가 필터링도 되기 때문에 그림 4와 같이 필터링된 ACK를 수용하기 위해 별도의 버퍼가 사용된다.

### 3.3 ACK 지연

필터링된 ACK의 흐름은 ACK 지연 혼잡에 따라 조정되어 ACK의 정보 손실율이 명시적으로 계산됨으로서 ACK는 그 속도에 비례하여 전송되므로 느린 구동 설계와 혼잡 회피 설계가 각기 다른 방식으로 윈도우 크기를 증가시킴으로서 ACK 지연에 미치는 영향도 다르다. 예를 들어, 느린 구동 단계에서 ACK를 수신하면 윈도우 크기가 패킷 하나만큼 증가하고 이어서 최대 2개의 패킷이 배포되기 때문에 ACK는 혼잡회피 단계보다 더 큰 간격을 두어야 포워드(forward) 버퍼가 오버플로우 되지 않는다[3]. 중간 노드가 TCP 소스와 같이 윈도우 크기를 다르게 함으로써, ACK를 지연시키도록 제안한다. 그러나 실제 네트워크에서는 TCP 소스가 다양하게 구현되기 때문에 라우터가 TCP 소스에서 윈도우를 추적해 내는 것이 가능하며, 이는 혼잡 회피단계에서 충분히 고려되어야 한다.

ACK 지연 알고리즘에서 사용되는 명칭 :

$Q$  : 버퍼 점유(패킷 또는 바이트 단위)

$Th$  : 정체 통보 트리거 임계값(패킷 또는 바이트 단위)

$D$  : 동일한 네트워크 요소에서 연속되는 2개의 ACK의 떨어진 간격(초 단위)

$d$  : 가장 정체된 링크에서 2개의 데이터 패킷의 최소 시간차(초 단위)

ACK 지연 알고리즘:

$Q < Th$ 인 경우

$D = d$

그 밖의 경우

$D = 2 * d$

### 3.4 TCP 트래픽 제어 구현

고속 TCP 구현은 *ip3\_rte* 프로세스 모델을 기반으로 하여 수정된 프로세스 모델은 *fast\_tcp\_ip3\_rte* 라

고 명명하고, 2개의 상태와 5개의 전이(轉移)를 추가하여 그림 5와 같이 구현한다.

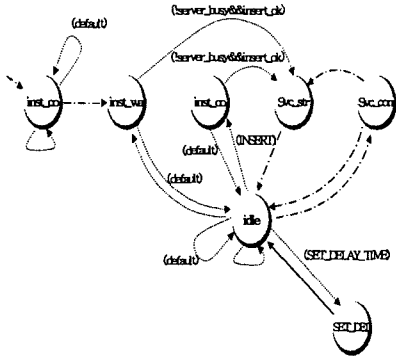


그림 5. TCP 트래픽 제어 구현  
Fig. 5 TCP traffic control implementation

상태: *inst\_comp(forced)*와 *SET\_DELAY\_TIME (forced)*  
전이:

(!server\_busy && insert\_ok): inst\_wait -> svc\_start

(INSERT): idle -> inst\_comp

(SET\_DELAY\_TIME): idle -> SET\_DELAYTIME

정의:

```
#define INSERT ((op_intrpt_type()==
OPC_INTRPT_SELF) && (op_intrpt_code()==2))
#define SET_DELAY_TIME
((op_intrpt_type()==OPC_INTRPT_STAT) &&
(ftcp_enabled == OPC_TRUE))
```

#### IV. 시뮬레이션 및 성능 분석

본 논문에서 시뮬레이션 환경 설정은 네트워크 시뮬레이터 ns-2를 기반으로 하는 RTPr 시뮬레이션 틀 환경을 적용하였다. ns는 TCP, 라우팅 프로토콜, 멀티캐스트 프로토콜, RTP(Real time Protocol), SRM(Scalable Reliable Multicast) 등 다양한 인터넷 프로토콜에 대한 시뮬레이션을 수행하기에 적절한 네트워크 시뮬레이션 도구로서 RTPr 틀 환경에서 공유된 병목 링크에서 몇 개의 동시 연결의 중복된 병목 네트워크 환경을 모델링하기 위해 ns를 수정함으로써 구현하였다.

#### 4-1 IP 네트워크

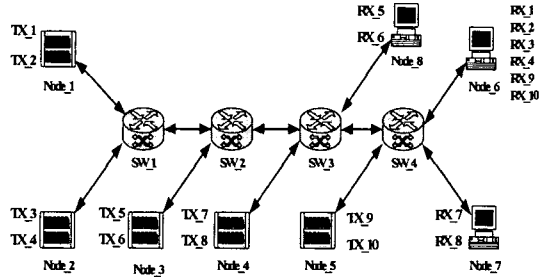


그림 6. 시뮬레이션 구성  
Fig. 6 Simulation configuration

위 그림 6에서 5개의 TCP 소스, 4개의 라우터, 3개의 TCP 수신지로 이루어진 시뮬레이션 환경을 구성하였고, 이 시뮬레이션 모델에서 사용된 몇 가지 중요한 매개변수를 아래와 같이 정의하였다.

소스와 수신지:

소스: Node\_1, Node\_2, Node\_3, Node\_4, Node\_5

수신지: Node\_6, Node\_7, Node\_8

TCP 최초 혼잡 윈도우: 1메가바이트

TCP 최대 세그먼트 크기: 512바이트

TCP 수신 버퍼: 2메가비트

TX<sub>x</sub>와 RX<sub>x</sub>는 동일한 TCP 연결이 되므로 네트워크 상에서 10개의 TCP 연결이 설정된다. RX<sub>x</sub>는 FTP 서비스를 통해 TX<sub>x</sub>로부터 파일 하나를 다운로드하고, 또 모든 RX는 동시에 10초 동안 FTP 요청을 전송하게 된다. 더 자세히 설명하면, Node<sub>1</sub>은 TX<sub>1</sub>과 TX<sub>2</sub>, Node<sub>2</sub>는 TX<sub>3</sub>과 TX<sub>4</sub>, Node<sub>3</sub>은 TX<sub>5</sub>와 TX<sub>6</sub>, Node<sub>4</sub>는 TX<sub>7</sub>과 TX<sub>8</sub>, Node<sub>5</sub>는 TX<sub>9</sub>와 TX<sub>10</sub>, Node<sub>6</sub>은 RX<sub>1</sub>, RX<sub>2</sub>, RX<sub>3</sub>, RX<sub>4</sub>, RX<sub>9</sub>, RX<sub>10</sub>, Node<sub>7</sub>은 RX<sub>7</sub>과 RX<sub>8</sub>, Node<sub>8</sub>은 RX<sub>5</sub>와 RX<sub>6</sub>을 각각 포함한다.

라우터:

라우터는 SW<sub>1</sub>, SW<sub>2</sub>, SW<sub>3</sub>, SW<sub>4</sub>를 포함한다.

IP 전송 속도: 1,200패킷/초

IP 포워드 버퍼: 5메가비트

IP 백워드 버퍼: 5메가비트

링크:

링크 속도 : 1.544메가비트/초

지연:

Node\_1에서 SW\_1까지는 1ms, Node\_2에서 SW\_1까지는 1ms, Node\_3에서 SW\_2까지는 9.5ms, Node\_4에서 SW\_3까지는 9.5ms, SW\_3에서 Node\_8까지는 9.5ms, Node\_5에서 SW\_4까지는 25ms, SW\_4에서 Node\_6까지는 1ms, SW\_4에서 Node\_7까지는 9.5ms, SW\_1에서 SW\_2까지는 1ms, SW\_2에서 SW\_3까지는 1ms, SW\_3에서 SW\_4까지는 1ms이다. SW\_4의 고속 TCP는 forward 버퍼 점유 대역폭을 감소시키는 걸 알 수 있다.

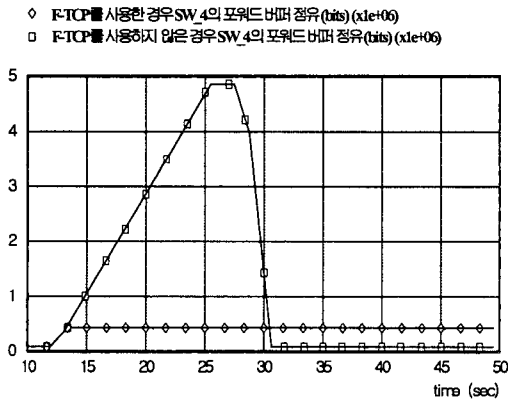


그림 7. SW\_4의 포워드 버퍼 점유  
Fig. 7 Forward buffer bandwidth of SW\_4

고속 TCP는 포워드 버퍼 점유를 감소시킨다는 사실을 명확하게 보여주기 때문에 고속 TCP가 전혀 사용되지 않는 경우 포워드 대기 열 양은 버퍼 오버플로우가 발생할 만큼 높은 수준까지 이르게 된다. 따라서 패킷 손실을 피하려면, 포워드 버퍼 용량이 버퍼 오버플로우를 방지할 만큼 커야 한다. 이에 비해 고속 TCP가 활성화된 경우에는 혼잡된 노드를 지나서 모든 TCP 연결의 ACK가 지연되므로, 포워드 버퍼 점유는 낮은 수준에서 매우 안정적인 상태를 유지되는 것을 그림 7에서 나타내고 있다. 따라서 고속 TCP는 작은 버퍼에서도 패킷 손실을 예방할 수 있으며, 특히 대기열 양과 길이가 안정적이라는 것은 소스트래피가 실질적으로 평활 상태에 있음을 의미한다.

◇ F-TCP를 사용한 경우 모든 RX의 수신데이터 합계 (bytes) (x1e+07)  
○ F-TCP를 사용하지 않은 경우 모든 RX의 수신데이터 합계 (bytes) (x1e+07)

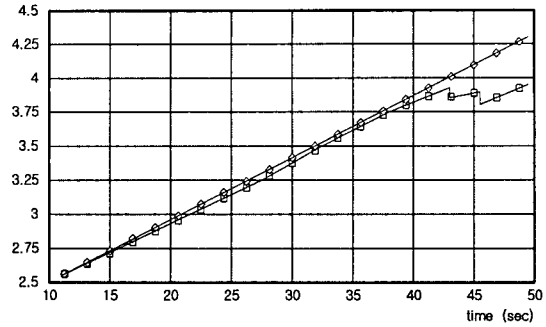


그림 8. 총 처리량  
Fig. 7 Total throughput

또한 그림 8과 같이 고속 TCP는 패킷 손실을 효과적으로 예방할 수 있으므로, 대개의 경우에 고속 TCP가 없을 때보다 전체 처리량이 대폭 증가한다. 손상된 TCP 세그먼트의 효과를 줄이는 데 고속 TCP 방식만으로는 충분하지 않은 것도 알 수 있었다. 대형 버스트 발생 시 어떤 TCP 세그먼트 하나가 손상되었다고 가정할 때 그 결과 TCP 수신기는 손상된 세그먼트를 제외하고 나머지 세그먼트를 모두 성공적으로 수신했다. TCP 소스가 손상된 세그먼트를 성공적으로 재전송하더라도, TCP 수신기는 전체 버스트를 총괄하는 단일 ACK를 생성할 것이다. TCP 소스가 이 ACK를 수신하면, 전송 윈도우를 한 단계 크게 확장시키고 고속 대형 버스트를 생성함으로써, 라우터 1의 포워드 버퍼는 오버플로우가 된다. 위 그림과 같이 18 ~ 20 초는 "오버 사이즈 된" 단일 ACK를 지연시키는 것은 별 도움이 되지 않으므로, 고속 TCP는 큰 역할을 할 수 없다는 것도 알 수 있었다.

그림 9는 TCP 소스에서 수신된 TCP 세그먼트의 ACK 개수를 나타낸 것으로 시간  $t = 25$  sec 일 때, 고속 TCP를 사용하는 시스템은 최대  $10.8 E 6$  바이트까지 TCP 세그먼트를 전송할 수 있다. 고속 TCP가 없으면 상기 지수는  $3.5 E 6$  바이트에 불과하므로 고속 TCP를 사용할 경우 크게 향상된 것을 알 수 있다.

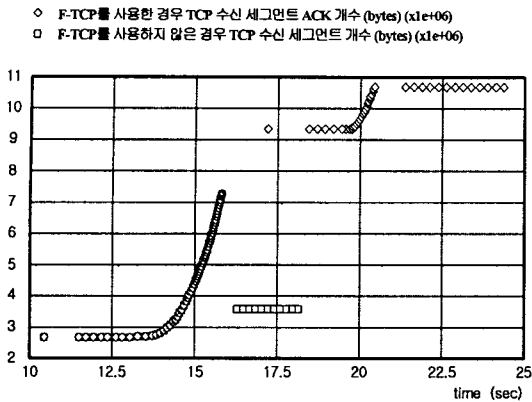


그림 9. 소스 노드에서의 TCP 수신 세그먼트 ACK 개수

Fig. 9 ACK numbers of TCP receiving segments in the source node

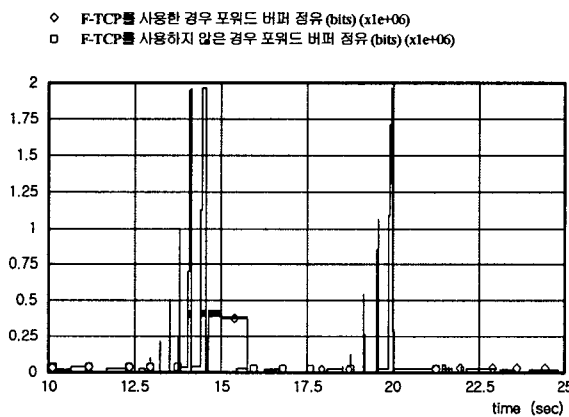


그림 10. Router 1의 포워드 버퍼 점유  
Fig. Forward buffer bandwidth of router 1

그림 10은 TCP 소스가 손상된 세그먼트를 성공적으로 재전송하더라도, TCP 수신기는 전체 버스트를 총괄하는 단일 ACK를 생성할 것이다. TCP 소스가 이 ACK를 수신하면, 전송 윈도우를 한 단계 크게 확장시키고 고속 대형 버스트를 생성함으로써, 라우터 1의 포워드 버퍼는 오버플로우가 된다. 위 그림과 같이 18 ~ 20 초는 오버 사이즈 된 단일 ACK를 지연시키는 것은 별 도움이 되지 않으므로, 고속 TCP는 큰 역할을 할 수 없다는 것도 알 수 있었다.

## V. 결론

본 논문에서는 라우터와 같은 네트워크 요소에서 혼잡이 발생 할 때 ACK를 지연시키는 새로운 TCP 트래픽 제어 방식을 제안 하였으며, RTO<sub>r</sub> 모델을 사용하여 그 성능을 평가하였다. 그 결과, 제안된 알고리즘은 TCP 연결의 처리량을 향상시키고 TCP 트래픽을 평활 화시킬 수 있음을 입증하였다. 시뮬레이션 틀로서 RTO<sub>r</sub>를 사용함으로써, 다양한 네트워크 환경에서 작업 내역 및 알고리즘 (고속 TCP)의 성능을 쉽게 파악할 수 있었다.

## 참고 문헌

- [1] S. Birchfield. Elliptical head tracking using intensity gra-dients and color histograms. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, pages 232?37,Santa Barbara, CA, June 1998.
- [2] G. Bradski. Computer vision face tracking for use in a per-ceptual user interface. Technical Report Q2, Intel Corpora-tion, Microcomputer Research Lab, Santa Clara, CA, 1998.
- [3] M. L. Cascia and S. Sclaroff. Fast, reliable head tracking under varying illumination. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, CO, June 1999.
- [4] A. Christian and B. Avery. Digital smart kiosk project. In Proc. ACM SIGCHI - Human Factors in Computing Systems Conference, pages 155?62, Los Angeles, CA, April 1998.
- [5] J. Crowley and F. Berard. Multi-modal tracking of faces for video communications. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, pages 640?45,Puerto Rico,PR, June 1997.
- [6] T. Darrell, G. Gordon, J. Woodfil, and M. Harville. A vir-tual mirror interface using real-time robust face tracking. In Proc. of the 3rd Int. Conf. on Automatic Face and Gesture ognition, pages 616?21, Nara, Japan, April 1998.

- [7] T. Darrell, B. Moghaddam, and A. Pentland. Active face tracking and pose estimation in an interactive room. Technical Report 356, M.I.T. Media Laboratory Perceptual Computing Section, Cambridge, MA, 1996.
- [8] Y. Ebisawa and S. Satoh. Effectiveness of pupil area detection technique using two light sources and image difference method. In A. Szeto and R. Rangayan, editors, Proceedings of the 15th Annual Int. Conf. of the IEEE Eng. in Medicine and Biology Society, pages 1268-1269, San Diego, CA, 1993.



**나상동(Sang-Dong Ra)**

e-mail: Sdna@mail.chosun.ac.kr

1968년 : 조선대학교 공학학사

1980년 : 건국대학교 공학석사

1995년 : 원광대학교 공학박사

1973년~현재 : 조선대학교 전자·정보공과대학 컴퓨터공학과 교수

1998년 : 조선대학교 전자계산소 소장 역임

1995~1996, 2001~2002년 : Dept. of Electrical & Computer Eng. Univ. of California Irvine 연구교수

※ 관심분야 : TMO, 실시간통신, 디지털통신, 데이터 통신망 및 신호처리 등임.

저자 소개



**나 하 선(Ha-Sun Na)**

1998년 한국항공대학교 항공전자공학과 졸업(공학사)

1995년 University of Colorado at Boulder, Dept. of Electrical and

Computer Eng. Graduate School.(공학석사)

※ 주관심분야 : 마이크로파 통신, RF 통신, 실시간 통신, 통신망 등



**김광준(Gwang-Jun Kim)**

1993년 조선대학교 컴퓨터공학과 졸업(공학사)

1995년 조선대학교 대학원 컴퓨터공학과 졸업(공학석사)

2000년 조선대학교 대학원 컴퓨터공학과 졸업(공학박사)

2000년~2001년 Dept. of Electrical & Computer Eng. Univ. of California Irvine Postdoc.

2003년~현재 여수대학교 컴퓨터공학과 전임강사

주관심분야 : ATM망, 인터넷 통신, 컴퓨터 네트워크, 실시간 통신 프로그래밍, 영상 통신, 프로그래밍 언어 (Visual C++, Java), 이동 통신 등