

초기 슬로우 스타트 구간에서 웹 객체의 평균 전송 시간 추정을 위한 수학적 모델

이 용 진*

<국문초록>

현재 인터넷은 응용계층에서 HTTP를 사용하고 트랜스포트 계층에서는 TCP를 사용하여 서비스를 제공하고 있다. 새롭게 제안된 전송 계층 프로토콜인 SCTP(Stream Control Transmission Protocol)는 슬로우 스타트 기간 동안에 초기 윈도우의 값 등을 제외하고는 TCP와 유사한 혼잡 제어 메커니즘을 사용한다. 본 논문에서는 이 점에 주목하여 슬로우 스타트 기간 동안에 HTTP over SCTP의 평균 전송 시간을 구하는 수학적 모델을 제시하고 이를 기존의 HTTP over TCP와 비교한다. 비교 결과는 HTTP over SCTP의 평균 전송 시간이 HTTP over TCP의 그것보다 평균 11 % 우수함을 보여준다.

주제어 : HTTP, SCTP, TCP, 평균 전송 시간, 슬로우 스타트, 수리적 모델

* 교신저자: 이용진(lyj@knue.ac.kr), 한국교원대학교 기술교육과 교수

I. 서 론

현재 사용되고 있는 대부분의 HTTP 응용은 인터넷에 존재하는 웹 객체를 인출하는 데 있어서 TCP를 하위 계층 프로토콜로 사용하고 있으나, HTTP와 TCP는 최적으로 대응되는 프로토콜이 아니다. HTTP는 연결 지향 특성을 갖고 있기 때문에 TCP를 사용하는 웹 서버는 슬로우 스타트 구간과 혼잡 회피 구간으로 구성되는 혼잡 제어 메커니즘을 사용하여 전송률을 조절한다. 일반적으로 TCP의 혼잡 제어는 1개의 세그먼트로부터 출발하여 데이터 전송률을 지속적으로 증가시킨다. 즉, 하나의 전송 윈도우에 대해 RTT(Round Trip Time) 동안에 수신측의 긍정 응답을 받으면 2 개의 윈도우로 전송량을 늘리고, 다음 RTT 동안의 긍정 응답에 대해 다시 4 개의 윈도우 만큼의 전송량을 보낸다. 이러한 과정은 이미 정의된 임계값(threshold)에 이르기까지 반복된 후에 혼잡 회피 단계로 진입한다. 혼잡 회피 단계는 웹 서버가 각 RTT 마다 하나의 윈도우 세그먼트 단위로 긍정 응답을 확인하게 되므로 선형 증가 단계라고도 한다.

주지하다시피, 대부분의 HTTP 전송은 그 크기가 작고 작은 량의 데이터를 포함하고 있기 때문에, 거의 대부분의 전송이 슬로우 스타트 구간에서 종료된다. 따라서 본 논문의 초점은 TCP와 SCTP의 슬로우 스타트 구간에 맞춰진다. SCTP(Stream Control Transfer Protocol)는 IETF에 의해 표준화되었고, 윈도우 기반의 혼잡 제어, 에러 탐지, 그리고 재전송 등의 TCP 혼잡 제어 메커니즘을 상당 부분 흡수하였다 (Stewart et al., 2000; Fu et al., 2005). 그러나 SCTP는 TCP에 없는 몇 가지의 새로운 특징을 추가하였는데, DOS 공격에 대한 견고성, TCP의 헤드-오브-라인 블로킹(head-of-line blocking)을 감소시키기 위한 멀티-스트리밍(multi-streaming), 그리고 양 종단 호스트 사이에 여러 개의 IP 주소를 바인딩 함으로써 얻어지는 멀티-호밍(multi-homing) 등의 기능들이 그것들이다. 또한, SCTP의 초기 슬로우 스타트 구간에서 초기 윈도우가 $2 \times \text{MTU}$ (maximum transfer unit) 인데 비해 (Stewart et al., 2000) TCP의 그것은 $1 \times \text{MSS}$ (Maximum Segment Size)로 정의되어 있다(Stevens, 1997). 물론, 개정된 TCP 명세는 초기 윈도우의 크기를 임의의 세그먼트까지 증가할 수 있도록 되어 있고, 실제 NetBSD 버전의 운영체제는 초기 윈도우의 값으로 4를 사용하고 있기도 하다 (Allman et al., 1998). 한편 SCTP의 경우에서도 RFC-4460(Stewart et al., 2006)에서는 초기 윈도우의 개수로 $4 \times \text{MTU}$ 가 제시되어 있다. 그러나 현재 모든 호스트들이 RFC-4460을 채택하고 있지 못하므로 본 논문에서는 공정한 비교를 위해서 광범위하게 유포된 TCP와 SCTP 표준안들인 RFC-2001(Stevens, 1997)과 RFC-2960(Stewart et al., 2000)을 이용하고자 한다.

TCP와 SCTP에 관한 대부분의 성능 비교 연구가 NS-2와 같은 시뮬레이션 환경에서 이루어진데 비해 실제로 구현된 SCTP 모듈을 장착한 리눅스 환경에서 시도된 연구로

김주현,이용진(2007)이 있다. 그 연구에서는 패킷 손실이 없는 싱글흐드를 가정하여, 초기 슬로우 스타트 동안에 혼잡 제어 알고리즘의 동작 과정을 추적하면서 SCTP와 TCP의 평균 전송시간을 비교하였다. 그 결과, SCTP의 평균 전송시간이 TCP의 그것보다 15 % 정도 우수한 것을 발견하였다. 그러나 일반적으로 실험 연구는 내재적인 한계점인 실험 환경과 파라미터에 따른 결과의 유동성을 극복하기가 어렵다. 따라서 본 논문에서는 일반화된 수학적 모델을 수립하여 초기 슬로우 스타트 단계에서 TCP와 SCTP의 평균 지연 시간을 비교하고자 한다. 본 논문의 초기 버전은 (Lee & Atiquzzaman, 2007)에 발표된 바 있다.

전체 4 장으로 구성되는 본 연구는 2장에서 웹 객체의 평균 전송 지연을 구하기 위한 수학적 모델을, 3장에서는 성능 평가를, 마지막으로 4장에서는 본 연구의 요약과 추후 연구 과제에 대해 논의한다.

II. SCTP와 TCP의 웹 객체 전송 지연 모델

1. SCTP와 TCP의 혼잡제어

현재 우리 눈에 보이지 않는 인터넷이 큰 무리 없이 작동되고 있는 이면에는 중간 의 많은 라우터들의 역할과 함께 종단 컴퓨터의 운영체제 내에 존재하는 메커니즘의 역할이 크다. 특히, TCP가 가지고 있는 혼잡 제어 알고리즘은 네트워크에 유입되는 데이터의 양을 조절함으로써 인터넷이 범람하지 않고 작동되도록 도와주는 매우 유용한 알고리즘이다.

SCTP의 혼잡 제어는 TCP와 마찬가지로 윈도우에 기반을 둔 전송률 조절 알고리즘에 기반을 두고 있으며, TCP와 공통으로 갖고 있는 기능들은 다음과 같다.

- 수신 윈도우(receiver advertised window, rwnd)는 수신측 호스트에서 사용가능한 버퍼의 크기이다.
- 혼잡 윈도우(congestion control window, cwnd)는 송신측 호스트가 조절하는 것으로 네트워크에 전송할 수 있는 데이터 양이다.
- 슬로우 스타트 임계값(slow start threshold, ssthresh)은 송신측 호스트가 슬로우 스타트 구간과 혼잡 회피 구간을 나누기 위해 사용하는 변수이다.
- 패킷 손실이 발생하는 경우, 타임아웃 이전에 3개의 중복 응답(duplicate acknowledgement)을 받으면 슬로우 스타트 대신에 빠른 재전송 단계로 들어가고, 그렇지 않으면 슬로우 스타트 단계로 들어간다.

그러나, 다음의 몇 가지 기능들은 TCP와 비교했을 때, SCTP가 개별적으로 가지고 있는 기능들로, TCP의 성능을 개선하는 요인이 될 것으로 기대된다.

- SCTP는 명시적인 빠른 회복(fast recovery) 단계를 가지고 있지 않기 때문에, SACK을 사용하여 묵시적으로 빠른 회복을 기도한다.
- SCTP는 TCP와 달리 초기 슬로우 스타트 단계에서 초기 혼잡 윈도우를 2로 시작한다.
- SCTP는 부분적으로 응답된 데이터(partial_bytes_acked, pba)를 이용하여 혼잡 회피 상태에서 cwnd를 조절한다. pba는 데이터 전송 중에 손실된 패킷을 송신자에게 알려주고 이를 통해 송신자는 네트워크에 혼잡이 발생했음을 알고 손실된 패킷을 재전송한다.
- SACK의 사용이 TCP에서는 선택적(optional)인데 비해, SCTP에서는 의무적(mandatory)이다. 이를 통해 단일 윈도우내에서 발생하는 여러 개의 패킷 손실에 대해 적절한 조치를 취할 수가 있다. 또한, 여러 개의 세그먼트의 손실 이후에 일어나는 지루하고 시간 소모적인 슬로우 스타트 단계를 피할 수 있기 때문에 대역폭을 절감하고 처리율을 증가시킬 수 있다.
- TCP는 3 개의 중복 응답을 수신한 이후에 빠른 재전송을 시작하는 데 비해, SCTP는 4 개의 중복 응답 이후에 빠른 재전송을 시작한다. 그러나 SCTP는 혼잡 윈도우의 크기와 관계없이 손실된 세그먼트를 재전송할 수 있다.

2. 기호 정의

본 논문에서 개발하려는 수학적 모델에 사용되는 기호는 <표 1>과 같다.

<표 1> 수학적 모델에 사용되는 기호와 의미

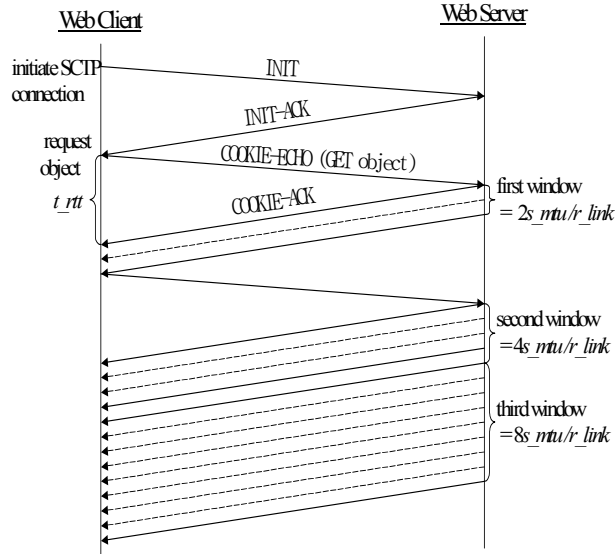
기호	의미
s_obj	전송 객체의 크기(bits)
s_mtu	SCTP의 최대 전송 단위(MTU: bits)
s_mss	TCP의 최대 세그먼트 크기 (MSS: bits)
r_link	웹 서버와 클라이언트 사이의 전송률(bps)
t_rtt	웹 서버와 클라이언트 사이의 라운드 트립 타임(RTT: sec)
t_stall	전송률 조절을 위해 웹 서버가 전송하지 못하는 정지 시간(sec)
α	객체를 전송하기 위해 필요한 윈도우의 개수
β	이론적인 웹 서버의 정지 횟수
γ	실제 웹 서버의 정지 횟수
t_tcp	TCP를 이용한 웹 객체의 평균 전송 시간(sec)
t_sctp	SCTP를 이용한 웹 객체의 평균 전송 시간 (sec)

3. SCTP를 이용한 웹 객체의 평균 전송 시간 모델

앞에서 언급한 바와 같이, 초기 슬로우 스타트 단계에서 윈도우의 크기는 SCTP에서 $2 \times \text{MTU}$ 인 반면에 TCP에서는 $1 \times \text{MSS}$ 로 세팅되어 있다. 효율성을 위해 최근의 구현에서는 이 값을 SCTP에 대해서는 $4 \times \text{MTU}$ 로 TCP에 대해서도 $4 \times \text{MSS}$ 로 하는 운영체제도 있으나, 대부분의 호스트들은 기존의 표준안에 맞추어 세팅되어 있으므로 본 논문은 RFC-2960과 RFC-2001을 따라서 모델링하고자 한다. 물론, 초기 윈도우의 크기가 달라진다고 해도 수학적 모델링의 절차는 동일하므로, 유도된 수식에서 파라미터의 값만 변경하면 될 것이다. 즉, 본 연구의 의의는 초기 윈도우의 크기의 차이에 있는 것이 아니라, 이 문제에 대한 수학적 모델링을 시도하는 데 있다.

슬로우 스타트 구간에서 SCTP에 의한 웹 객체의 전송에 대한 시간 다이어그램이 [그림 1]에 나타나있다. SCTP는 4-way-handshake를 사용하는 데, 그 이유는 쿠키 메커니즘을 사용하여 어소시에션을 설정하고 blind SYN attack을 막기 위해서이다. 웹 클라이언트는 초기에 INIT chunk를 웹 서버로 전송하고, 이에 대해 웹 서버는 INIT-ACK를 돌려준다. 웹 클라이언트는 COOKIE-ECHO를 서버로 전송하면서 웹 객체를 요구하는 GET object를 같이 보낸다. 이에 대해 웹 서버는 쿠키의 유효성을 검증한 후에, COOKIE-ACK와 함께 요청된 웹 객체를 클라이언트에게 돌려준다. 따라서 [그림 1]에서 4-way-handshake를 사용하지만, 실제로는 TCP에서와 같이 3-way-handshake만에 웹 객체가 전송될 수 있다. 따라서 연결 설정 과정에 따른 SCTP의 추가 지연은 없다.

(s_{obj}/s_{mtu}) 는 객체 내에 포함되어있는 세그먼트의 개수를 의미한다. [그림 1]의 예에서 (s_{obj}/s_{mtu}) 는 14이다. 이제, 각 윈도우에 포함되어 있는 세그먼트의 개수를 고려해 보자. 첫 번째 윈도우는 2 개의 세그먼트가 포함되고, 두 번째 윈도우에는 4 개의 세그먼트가, 세 번째 윈도우에는 8 개의 세그먼트가 포함되어 있다. 따라서 k 번째 윈도우에는 2^k 개의 세그먼트가 포함된다. 기호의 정의에서 a 는 웹 객체를 전송하기 위해 필요한 윈도우의 개수이고, [그림 1]에서 보듯이 $a=3$ 이다.



[그림 1] 슬로우 스타트 구간에서 SCTP에 의한 웹 객체의 전송

일반적으로 a 는 객체의 크기를 최대 전송 단위(MTU)의 크기로 나누면 되므로 다음 식으로 유도된다.

$$\begin{aligned}
 &= \min \left\{ k : 2^{1/i} + 2^{2/i} + \dots + 2^{k-1/i} \leq \frac{s_obj_r}{s_mtu_r} \right\} \\
 &= \min \left\{ k : 2^{k+1/i} \leq \left(\frac{s_obj_r}{s_mtu_r} + 2 \right) \right\} \\
 &= \min \left\{ k : k \leq \log_2 \left(\frac{s_obj_r}{s_mtu_r} + 2 \right) - 1 \right\} \\
 &= \log_2 \left(\frac{s_obj_r}{s_mtu_r} \right)
 \end{aligned} \tag{1}$$

웹 서버는 데이터 윈도우를 전송한 후 클라이언트로부터의 응답을 기다리면서 정지할 수 있다. [그림 1]에서, 웹 서버는 첫 번째 윈도우를 전송한 후 정지된다. 이제, 이를 일반화하여 k 번째 윈도우를 전송한 후에 정지하는 경우를 생각해 보자. 먼저 k 번째 윈도우의 전송 시작 이후에 해당 윈도우의 첫 번째 세그먼트에 대한 응답을 받을 때까지의 시간은 $2 \times (s_mtu/r_link) + t_rtt$ 가 된다. 다음으로 k 번째 윈도우의 전송 시간은 $(s_mtu/r_link) \times 2^k$ 가 된다. 웹 서버의 정지 시간은 이 두 시간의 차이로 정의할 수 있으므로 식 (2)가 된다.

$$t_{stall} = \left[\frac{2 \times s_mtu}{r_link} + t_{rtt} - 2^k \left(\frac{s_mtu}{r_link} \right) \right] \quad (2)$$

최악의 경우, 웹 서버는 $(a-1)$ 개의 윈도우 각각의 전송 이후에 정지될 수 있다. 따라서 SCTP의 웹 객체를 전송하기 위한 지연은 SCTP의 연결 설정, 객체 요구 및 전송 그리고 웹 서버의 정지 시간의 합이다. 이를 수식으로 표현하면 식 (3)이 된다.

$$t_{sctp} = 2 \times t_{rtt} + \frac{s_obj}{r_link} + \sum_{k=1}^a \left[\frac{2 \times s_mtu}{r_link} + t_{rtt} - 2^k \left(\frac{s_mtu}{r_link} \right) \right] \quad (3)$$

이제, 보다 일반화된 지연 시간 모델을 얻기 위해, 이론적인 정지 횟수인 β 를 도입한다. β 는 무한대의 세그먼트를 포함하는 객체를 전송할 때 웹 서버가 정지하는 횟수로 정의한다. 식 (1)에서 a 를 유도한 방법을 사용하여 β 를 구하면 식 (4)를 얻는다. 이때, 실제 웹 서버의 정지 횟수인 y 는 $\min(a-1, \beta)$ 로 주어진다.

$$\begin{aligned} \beta &= \max \left\{ k : \frac{2 \times s_mtu_r}{r_link_r} + t_{rtt_r} - \frac{2^k \times s_mtu_r}{r_link_r} \geq 0 \right\} \\ &= \max \left\{ k : 2^k d \left(2 + \frac{t_{rtt} \times r_link_r}{s_mtu_r} \right) \right\}_r \\ &= \max \left\{ k : k \leq \log_2 \left(2 + \frac{t_{rtt} \times r_link_r}{s_mtu_r} \right) \right\}_r \\ &= \log_2 \left(2 + \frac{t_{rtt} \times r_link_r}{s_mtu_r} \right)_r \end{aligned} \quad (4)$$

식(3)과 식(4)로부터 SCTP를 이용한 웹 객체의 평균 전송 시간을 나타내는 식 (5)를 얻는다.

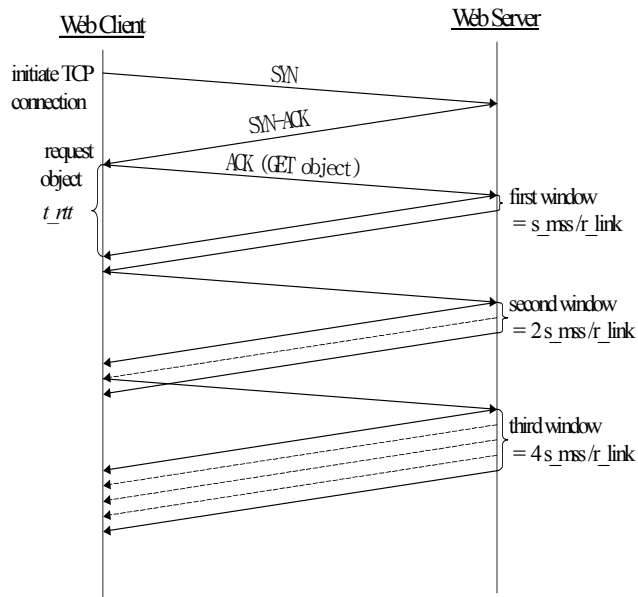
$$\begin{aligned} t_{sctp} &= 2 \times t_{rtt} + \frac{s_obj_r}{r_link_r} + \sum_{k=1}^{\beta} \left[\frac{2 \times s_mtu_r}{r_link_r} + t_{rtt_r} - 2^k \left(\frac{s_mtu_r}{r_link_r} \right) \right] \\ &= 2 \times t_{rtt} + \frac{s_obj_r}{r_link_r} + \ddot{Y} \left[t_{rtt} + \frac{2 \times s_mtu_r}{r_link_r} \right] (2^{\beta+1} - 2) \frac{s_mtu_r}{r_link_r} \end{aligned} \quad (5)$$

where $_r$

$$\ddot{Y} = \min \left\{ \log_2 \left(2 + \frac{t_{rtt} \times r_link_r}{s_mtu_r} \right) \right\}_r, \log_2 \left(\frac{s_obj_r}{s_mtu_r} \right) - 2_r$$

4. TCP를 이용한 웹 객체의 평균 전송 시간 모델

TCP를 이용하여 웹 객체의 평균 전송 시간을 나타내는 시간 다이어그램이 [그림 2]에 나타나있다. TCP는 3-way-handshake를 사용하여 연결을 설정한다. 웹 클라이언트는 SYN 패킷을 보내 웹 서버와 연결을 요청하고, 웹 서버는 SYN-ACK로 응답한다. 웹 클라이언트는 이에 대한 응답으로 ACK와 함께 웹 객체를 요청하는 HTTP 메시지 (GET object)를 전송한다. 이에 대해 웹 서버는 해당되는 웹 객체를 클라이언트에게 전송한다.



[그림 2] 슬로우 스타트 구간에서 TCP에 의한 웹 객체의 전송

TCP를 이용한 웹 객체의 평균 전송 시간을 구하면 식 (6)을 얻는다(Heidemann et al., 1997; Cardwell et al., 2000; Kurose & Rose, 2005, pp. 275-284).

$$t_{tcp} = 2 \times t_w + \frac{s_{obj}}{r_{link}} + \ddot{Y} \left[t_{rtt} + \frac{s_{mss}}{r_{link}} \right] (2^{\ddot{Y}} - 1) \frac{s_{mss}}{r_{link}}$$

where \ddot{Y}

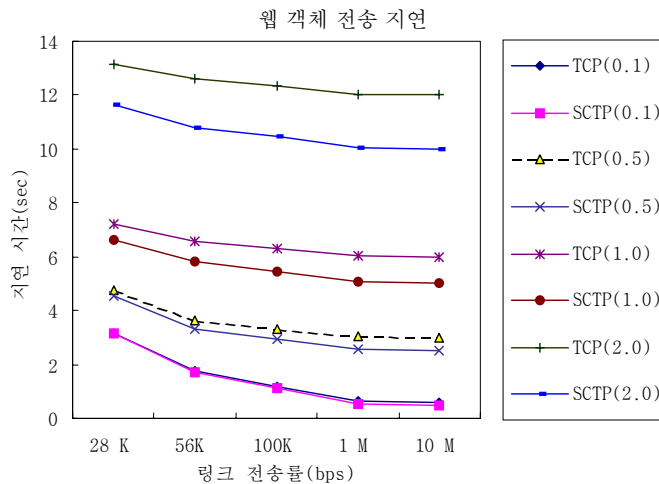
$$\ddot{Y} = \min \left\{ \log_2 \left(1 + \frac{t_{rtt} \times r_{link}}{s_{mss}} \right) + 1, \log_2 \left(\frac{s_{obj}}{s_{mss}} + 1 \right) \right\} - 1$$

(6)

Ⅲ. 성능 평가

본 연구에서 제안한 수학적 모델은 실험에 의한 방법이 아니므로 추가적인 오버헤드가 존재하지 않는다. 모델을 평가하기 위해 수리적 계산을 이용한 성능평가를 수행하였다. [그림 3]은 식 (5)와 (6)에서 링크 전송률과 RTT를 변화시켜가면서 얻은 TCP와 SCTP에 대한 웹 객체의 전송 지연을 보여준다. [그림 3]의 범례에서 TCP(0.1)과 SCTP(0.1)은 각각 RTT = 0.1 (sec)일 때의 TCP와 SCTP의 전송 지연을 표시한다. 즉, TCP와 SCTP 각각에 대해 RTT는 0.1, 0.5, 1.0, 2.0 (sec)가 사용되었으며 링크 전송률은 [그림 3]의 x축에서 보듯이 28 K, 56 K, 100 K, 1 M, 10 M (bps)가 사용되었다. 공정한 비교를 위해 SCTP의 MTU와 TCP의 MSS를 똑같이 536 B로 하였고, 웹 객체의 크기는 모두 10 KB이다.

[그림 3]에서 보듯이 TCP에 대한 SCTP의 이득은 링크전송률과 RTT가 증가함에 따라 커진다. 이것은 SCTP가 TCP에 비해 라운드 트립 타임이 크거나 보다 고속인 네트워크에서 더 우수한 성능을 갖고 있음을 의미한다. 계산 결과로부터, 슬로우 스타트 구간에서 SCTP의 평균 지연이 TCP의 그것보다 11 % 정도 작음이 확인되었다.



[그림 3] SCTP와 TCP의 웹 객체 전송 지연 비교

IV. 결 론

본 논문에서는 슬로우 스타트 구간에서 웹 객체를 전송하기 위한 프로토콜로 SCTP와 TCP를 해석적으로 비교하기 위한 수학적 모델을 제안하였다. 제안된 모델은 웹 객체의 평균 지연을 잘 묘사하였고, SCTP의 성능이 TCP에 비해 평균적으로 11 % 정도 우수한 것으로 판명되었다. 이 결과는 실제 실험에 의한 결과인 평균 15 % 보다는 약간 작은 결과이다. 앞으로의 연구에서는 패킷 손실까지를 고려하여 SCTP의 혼잡 제어에 대한 보다 정밀한 수학적 모델이 기대된다.

참고문헌

- 김주현, 이용진(2007). 초기 슬로우 스타트 단계에서 SCTP의 평균 전송 시간. **대한 공업교육학회지**, 32(2). 199-216.
- Stewart, R., Arias-Rodriguez, I., Poon, K., Caro, A. & Tuexen, M. (2006). Stream Control Transmission Protocol(SCTP) Specification Errata and Issues. *IETF RFC 4460*.
- Stewart R., Xie Q., Morneault K., Sharp C., Schwarzbauer H., Taylor T., Rytina L., Kalla M., Zhang L., & Paxson V.(2000), Stream Control Transmission Protocol. *IETF RFC 2960*.
- Fu, S., Atiquzzaman, M., Ma, L., & Lee, Y. (2005). Signaling Cost and Performance of SIGMA. *Wireless Communications and Mobile Computing*, 5(7), 825-845.
- Allman, M. Floyd, S. and Partridge, C. (1998). Increasing TCP's Initial Window, *RFC-2414*.
- Stevens, W. (1997). TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms, *RFC-2001*, 1997.
- Lee, Y. & Atiquzzaman, M. (2007). HTTP Transfer Latency over SCTP and TCP in Slow Start Phase. *Proceedings of IEEE ICC-2007 Conference*, Glasgow, Scotland.
- Heidemann, J., Obraczka, K. & Touch, J. (1997). Modeling the Performance of HTTP over Several Transport Protocols. *IEEE/ACM Transactions on Networking*, 5(5), 616-630.
- Cardwell, N., Savage, S. & Anderson, T. (2000). Modeling TCP Latency. *IEEE Infocom*, 1742-1751.
- Kurose, J. & Rose, K. (2005). *Computer Networking*, Addison-Wesley, 275-284.

<Abstract>**Mathematical Model for Mean Transfer Delay of
Web Object in Initial Slow Start Phase**

Yong-Jin Lee*

Current Internet uses HTTP (Hyper Text Transfer Protocol) as an application layer protocol and TCP (Transmission Control Protocol) as a transport layer protocol to provide web service. SCTP (Stream Control Transmission Protocol) is a recently proposed transport protocol with very similar congestion control mechanisms as TCP, except the initial congestion window during the slow start phase. In this paper, we present a mathematical model of object transfer latency during the slow start phase for HTTP over SCTP and compare with the latency of HTTP over TCP. Validation of the model using experimental result shows that the mean object transfer latency for HTTP over SCTP during the slow start phase is less than that for HTTP over TCP by 11%.

Key words : HTTP, SCTP, TCP, Mean Transfer Time, Slow Start, Mathematical Model

* Correspondence: Yong-Jin Lee(lyj@knue.ac.kr), Korea National University of Education