

리눅스 기반 **SCTP & TCP** 성능 비교 분석

2005년 5월

하종식, 고석주

{mugal1,sjkoh}@cs.knu.ac.kr

Abstract

Stream Control Transmission Protocol(SCTP)는 TCP와 UDP에 이어 세 번째로 등장한 전송계층 protocol이다. SCTP는 TCP와는 다른 특징들을 제공하는데, 본 논문에서는 TCP와 SCTP의 특징에 따른 성능 비교 분석을 수행하였다. 우리는 세 가지의 다른 시나리오에 대하여 성능을 비교 하였는데, 먼저 SCTP와 TCP 데이터 전송과정에서 socket system call에 대한 user data 바이트 수 변경 시 성능을 비교분석 하였고, 다음으로 동일 호스트에서 TCP와 SCTP 세션 전송을 동시에 수행하는 경우에 대한 traffic fairness 분석, 그리고 마지막으로 single-homing과 multi-homing에서의 throughput 성능을 비교 분석 하였다. 실험 결과, socket send system call에 대한 user data 크기를 크게 할수록 SCTP가 TCP보다 더 좋은 throughput을 제공하며, 동일 조건에서 TCP와 SCTP 트래픽은 공정하게 경쟁한다는 점, 그리고 SCTP multi-homing 전송에서 데이터 송수신 채널의 분리로 인해 single-homing의 전송 효율이 개선됨을 확인 하였다.

Table of Contents

1. INTRODUCTION.....	3
2. SCTP MULTI-STREAMING AND MULTI-HOMING FEATURES.....	3
2.1 SCTP MULTI-STREAMING.....	3
2.2 SCTP MULTI-HOMING	4
3. TEST SCENARIOS.....	5
3.1 SCENARIO 1: DIFFERENT SIZE OF USER INPUT DATA.....	5
3.2 SCENARIO 2: COMPETITION OF SCTP AND TCP TRAFFIC.....	6
3.3 SCENARIO 3: PERFORMANCE OF SCTP MULTI-HOMING	7
4. EXPERIMENTAL RESULTS	7
4.1 RESULTS FOR SCENARIO 1	7
4.2 RESULTS FOR SCENARIO 2	10
4.3 RESULTS FOR SCENARIO 3	11
5. CONCLUSION.....	12
REFERENCE.....	13

1. Introduction

SCTP(Stream Control Transmission Protocol)은 IETF SIGTRAN Working Group에서 2000년 10월 표준으로 승인된 프로토콜이다. SCTP는 연결 지향적인 프로토콜로서 통신을 위해서 종단(end point)간에 커백션을 설립해야 하며, SCTP에서는 이 커백션을 'association' 이라 부른다. SCTP는 TCP와 마찬가지로 신뢰성 있는 프로토콜이며, 몇 가지 사항을 제외하고 congestion control 과 flow control 등 대부분 TCP 프로토콜 방식을 채택하고 있다.

TCP와 다른 SCTP의 대표적인 특징에는 multi-streaming과 multi-homing이 있다. Multi-streaming 기능을 통해 SCTP는 여러 개의 다른 stream으로 나누는 것이 가능하며 TCP의 Head of Line (HoL) Blocking으로 인한 성능저하 문제를 해결하였다. 특히 SCTP 단말이 여러 개의 IP주소를 바인딩(binding) 할 수 있는 multi-homing 기능이 있어 데이터 전송의 신뢰성을 높일 수 있다. 즉, TCP는 하나의 호스트가 단지 하나의 IP주소만을 가지는 것을 가정하는 반면 SCTP는 한 호스트가 여러 개의 IP 주소를 가지는 것을 허용한다. 이에 따라 커백션 구분을 위해 TCP가 송신자 주소, 송신자 포트 번호, 수신자 주소, 수신자 포트 번호의 조합을 이용하는 것과는 달리 SCTP는 송신자의 IP 주소 집합, 송신자의 포트 번호, 수신자의 IP 주소 집합, 수신자의 포트 번호로 구분하게 된다. 이러한 multi-homing의 지원은 네트워크 장애(failure)에 좀 더 유연하게 대처할 수 있도록 하며, 더 높은 수준의 신뢰성을 지원할 수 있도록 한다.

본 논문은 SCTP의 이런 특징들이 TCP에 비해 어느 정도 전송효율(throughput) 성능을 개선시킬 수 있는지에 대한 비교 분석에 중점을 두려 한다. 본 논문에서는 먼저, SCTP와 TCP 데이터 전송에서 응용 계층에서 호출하는 socket API 함수의 "전송 바이트 수" 변경에 의한 성능을 비교 분석하였고, 또한 TCP와 SCTP 세션을 동시에 수행하는 경우 두 프로토콜 사이의 트래픽 경쟁 상태 분석, 그리고 마지막으로 single-homing과 multi-homing에서의 전송성능을 비교 분석 하였다.

2. SCTP Multi-Streaming and Multi-Homing Features

2.1 SCTP Multi-Streaming

SCTP는 하나의 SCTP Association에 여러 개의 stream을 독립적으로 전송할 수 있는 multi-stream 기능이 있다. SCTP는 "각 stream마다" stream 순서번호(SSN: Stream Sequence Number)가 있어 stream 순서를 유지하며, 해당 stream으로 전송되는 데이터를 관리하여 신뢰성 있는 데이터 전송이 가능하게 한다. 이를 통해, TCP의 "Head of Line" blocking 문제를 방지 할 수 있다. 즉, SCTP에서는 한 stream의 데이터 전송이 지연되더라도, 다른 stream의 데이터들은 이와 무관하게 수신 응용에 전달될 수 있다. 이러한 특징은 "TCP의 byte-oriented" 특성과 대별되는 SCTP의 "message-oriented and stream-oriented" 특성에 기인한다.

그림 1은 SCTP 멀티스트리밍(multi-streaming)의 개념을 보여준다.

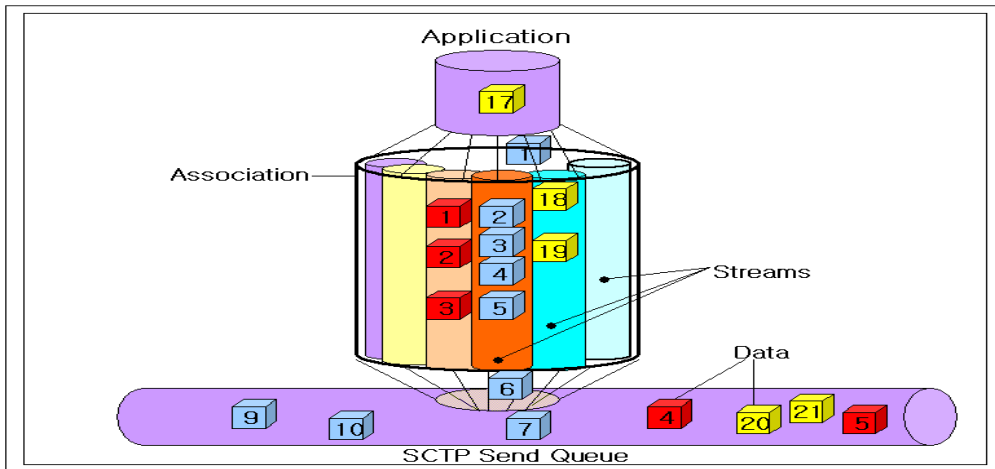


Figure 1. SCTP multi-streaming

SCTP 응용은 association 내에서 사용가능한 스트림의 수를 연결 초기화 단계에서 설정할 수 있으며, 이 스트림의 수는 다른 endpoint 와 협상을 통해서 결정된다. User 메시지는 스트림 ID별로 전송된다. 즉, 내부적으로 SCTP는 응용에서 전달되는 각 메시지들에 고유의 stream ID와 SSN (stream sequence number)를 부여하여 전송하게 된다. 특정 스트림 ID에 대하여 데이터는 순서적으로(ordered) 수신 응용에게 전달된다.

2.2 SCTP Multi-Homing

그림 2는 SCTP 멀티호밍 특성을 보여준다.

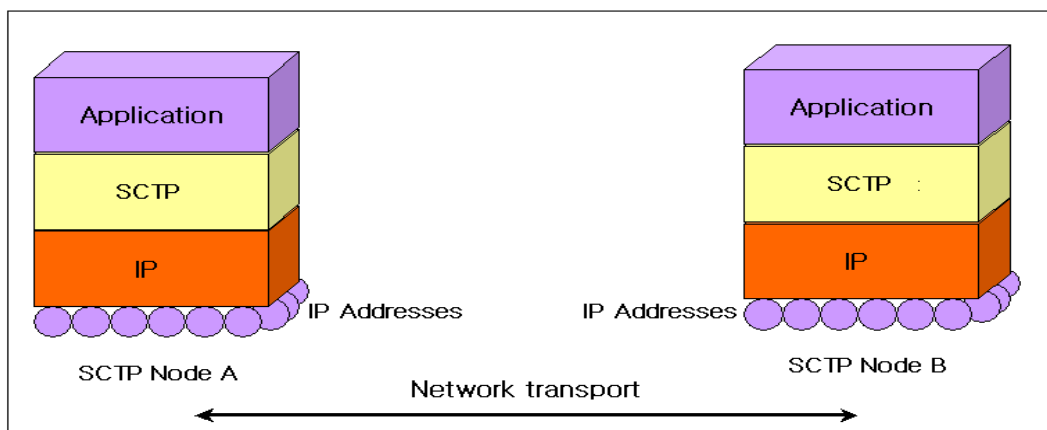


Figure 2. SCTP multi-homing

SCTP는 association을 맺고자 하는 endpoint들이 multiple NIC 각각에 IP가 할당된 multi homed 단말에 적용될 수 있다. 여러 개의 IP 주소가 사용되는 경우에도, 실제 데이터 전송은 primary IP 주소 하나를 통해 수행되며, 나머지 IP 주소는 backup IP 주소로 사용된다. 여러 개의 주소를 동시에 데이터 전송에 사용하는 "load-balancing" 이슈는 현재 논의 중에 있으나 표준화 작업 범위는 아니다. 한편, 세션 도중에 primary IP 주소를 다른 IP 주소로 변경하는 기능이 제공된다.

한편, 멀티호밍 경우에 단말은 "수신 IP 주소"와 "송신 IP 주소"를 각각 다르게 설정함으로써, NIC의 송수신 처리 부하를 분산시킬 수도 있다. 관련 성능 실험이 다음 절에 기술된다.

3. Test Scenarios

3.1 Scenario 1: Different Size of User Input Data

먼저 실험 1에서는 SCTP와 TCP의 성능평가 실험을 위해 데이터 전송 클라이언트와 데이터 수신 서버의 두 대의 단말로(각각 client, server) 테스트 베드를 구축하였다.

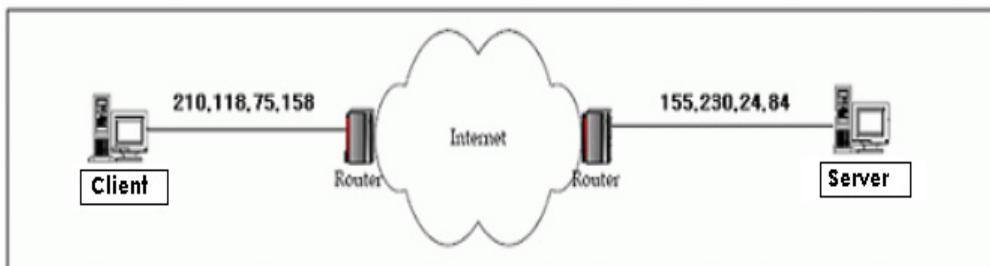


Figure 3. Network configuration for Scenario 1

각 단말에는 Linux kernel 2.6.10 및 LK-SCTP 도구를 설치하였다. 실험환경은 100Mb 의 데이터를 각각의 시나리오에 맞게 설정하여 전송하였다. User Level에 send buffer size를 점진적으로 변경하여 SCTP 와 TCP의 총 패킷 사이즈, 전송 시간, 패킷 당 데이터 효율성 비교를 테스트 한다. 다음은 SCTP와 TCP에서의 성능 비교분석을 위한 테스트 베드 구성도이다. 본 논문에서는 SCTP와 TCP에서의 파일전송 성능 비교 분석을 위해 아래와 같은 네트워크 상황 (HOP COUNT=13)의 테스트 베드를 구성하였다.

실험 1에 적용된 SCTP와 TCP의 성능평가 시나리오는 다음과 같다.

- 두 Router을 연결하는 중간 링크는 SCTP와 TCP에서 동일하게 적용하였고, 이에 따라 SCTP와 TCP는 동일한 대역폭 및 지연을 가진다.

- 먼저, **Server**는 고정서버로서 동작하며 '155.230.24.84' 주소를 이용하여 **bind()** 함수를 수행한 후 접속을 기다린다. 클라이언트는 서버주소 '55.230.24.84' 접속을 하여 세션이 맺어진 후 데이터를 점차적으로 변화 시켜서 전송한다.
- 위의 기본적인 가정 하에 두 **endpoint** 간에 전송 데이터 사이즈를 다양하게 변화시켜 보면서 **100Mb** 파일을 전송을 하기 시작하는 시간으로부터 파일이 수신 측에 모두 도착하기까지의 시간과 데이터 양, 총 수신 **packet**수 등을 측정하였다.

3.2 Scenario 2: Competition of SCTP and TCP Traffic

그림 4는 두번째 실험 시나리오를 보여준다.

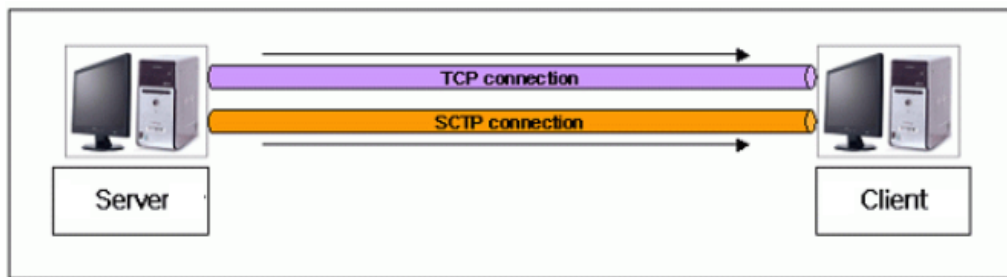


Figure 4. Competition of SCTP and TCP traffic

실험 2에 적용된 평가 시나리오는 다음과 같다.

- 두 **Router**를 연결하는 중간 링크는 **SCTP**와 **TCP**에서 동일하게 적용되고, 이에 따라 **SCTP**와 **TCP**은 동일한 대역폭 및 지연을 가진다.
- **TCP**와 **SCTP**의 경쟁을 분석하기 위해서 서버 쪽 응용프로그램과 클라이언트 응용프로그램에 **SCTP**와 **TCP** 모듈을 각각 설치하고 동시에 **100Mb**의 데이터를 전송하였다.
- 그 후 전송을 하기 시작하는 시간으로부터 파일이 수신 측에 모두 도착하기까지의 시간과 데이터량 , 총 수신 **packet**수 등을 측정하였다.

3.3 Scenario 3: Performance of SCTP Multi-Homing

그림 5는 세번째 실험 시나리오를 보여준다.

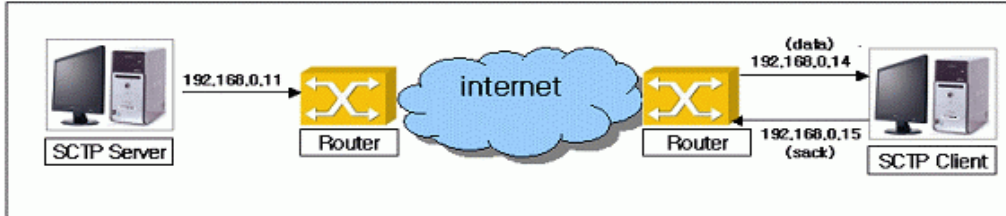


Figure 5. Network configuration for SCTP multi-homing

위의 그림과 같이 클라이언트가 Multi-homing 상황의 서버로 접속을 하고 클라이언트에서 서버로 100Mb의 데이터를 전송 시 데이터 흐름을 관찰 하였다. 본 실험에서는 SCTP에서의 Single-homing과 Multi-homing에서의 성능 평가를 알아보기 위해서 테스트 베드를 구축하고 실험을 수행 하였다. SCTP는 호스트가 Multi-homing을 지원하는 경우 즉 목적지는 여러 개의 주소를 가지고 그 목적지에 도달하기 위해 여러 개의 경로를 가지는 경우에 이 경로를 이용할 때와 호스트가 Multi-homing을 지원하지 않을 경우에 하나의 경로만을 이용하여 데이터의 송수신이 될 때에 대해서 실험을 수행하였다.

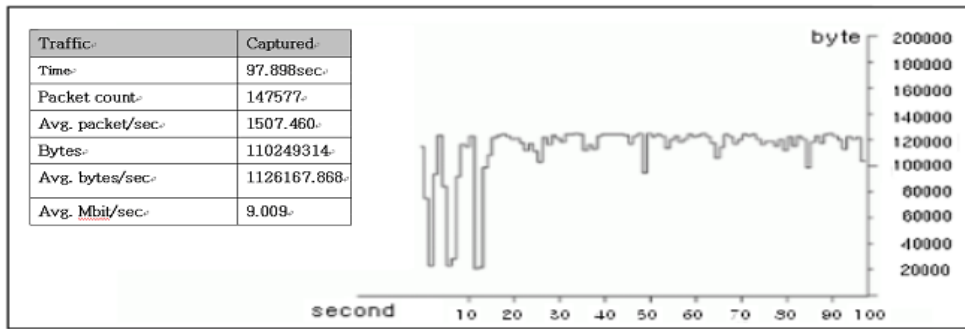
4. Experimental Results

4.1 Results for Scenario 1

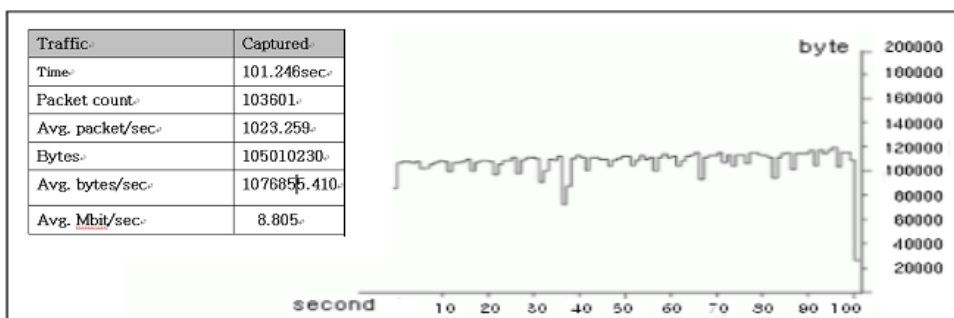
이 경우는 user level에서 application을 작성 시 TCP와 SCTP의 성능을 비교하고 각각의 프로토콜에서 어떤 상황에서 최상의 Performance를 보일 수 있는지에 대한 Testing이다.

Figure 6은 send 관련 socket API 호출시에 "8,192" 바이트씩 전송한 실험에 대한 결과이다.

그림에서 동일한 100Mb 전송 시 패킷의 수량은 SCTP가 147,000여개의 패킷이 생성되고 TCP가 103,000여개의 패킷을 생성하였다. 생성된 패킷들의 총 데이터량은 SCTP에 제어 CHUNK가 많이 포함되므로 패킷카운트와 data size가 증가 하였다. 하지만 packet 수량 자체는 전체적 데이터 전송속도에 많은 영향을 끼치지 못한다.



(a) Throughput of SCTP



(b) Throughput of TCP

Figure 6. Results for the user input data size of 8192 bytes

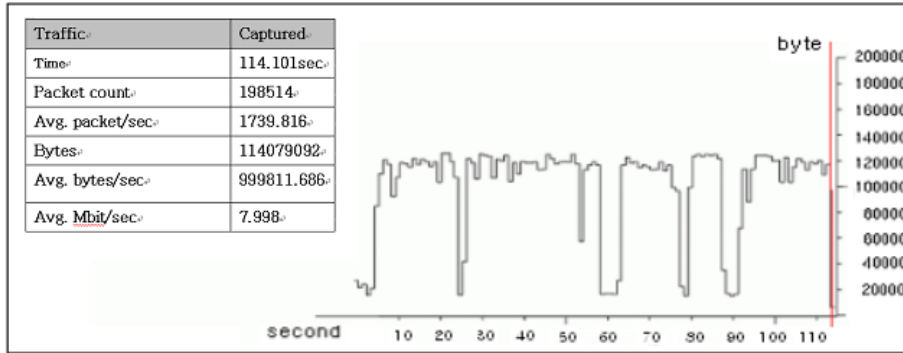
Figure 6(a), (b)를 비교 분석한 결과, 수행 시간 비교에서는 평균 초당 전송되는 SCTP의 데이터가 많다. 결론적으로 SCTP가 TCP보다 높은 Throughput을 보여준다. 이는 다음과 같은 TCP와 SCTP Congestion control의 특성으로 인해 기인한다.

TCP와 SCTP의 slow-start, 혼잡 회피, 빠른 재전송으로 이루어 지는 윈도우 기반 혼잡 제어 메커니즘을 사용하여 신뢰성 있는 전송을 보장하고 손실된 패킷, 순서에 어긋나게 도착한 패킷, 중복된 패킷을 감지한다.

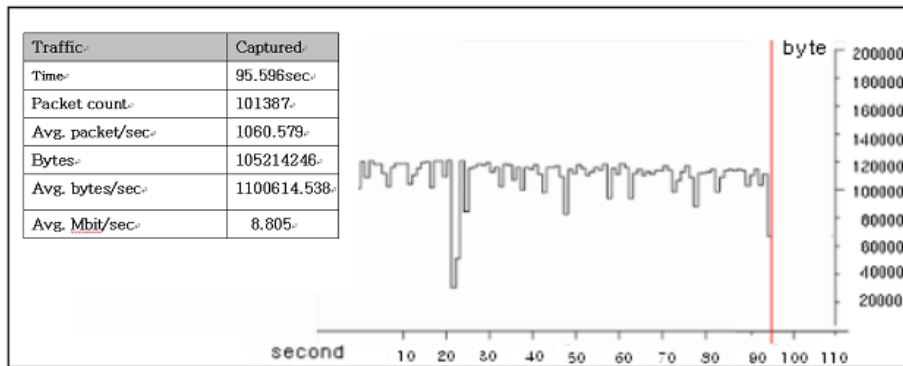
기본적인 메커니즘은 TCP와 유사하지만, 다음과 같은 차이점을 가진다. TCP의 경우 cwnd 초기값이 1MTU이나, SCTP에서는 2*MTU이다. 이는 동일한 네트워크 상황에서 SCTP가 TCP에 비해 좀더 빠른 시간 내에 많은 데이터를 전송할 수 있도록 한다.

Figure 7은 send 관련 socket API 호출시에 "2,048" 바이트씩 전송한 실험에 대한 결과이다.

또 실험결과 Figure 7. (a), (b)에서 나타난 결과 와 같이 2048씩 데이터를 전송 시에 SCTP에서 아주 낮은 효율을 보여준다. Packet count가 190000개 정도가 생성되면 이때의 TCP packet count는 TCP-8192byte 전송 시와 비슷한 101387이다.



(a) Throughput of SCTP



(b) Throughput of TCP

Figure 7. Results for the user input data size of 2048 bytes

상기 실험 결과, 다음과 같은 흥미로운 점을 관측할 수 있다. TCP의 전송 시 User Level 데이터를 어떤 사이즈로 전송하여도 TCP는 byte Stream 기반 한 경계가 없는 데이터의 전송이기에 데이터 전송 시 TCP계층에서의 Segmentation 될 때 packet 부분에 데이터를 가득 채워서 보낸다. 하지만 SCTP 전송은 메시지 기반 전송이다. 하나의 send 함수의 콜이 하나의 메시지를 발생시키고 하나의 메시지는 하나의 SSN값을 나타낸다. 예를 들어 8192byte 전송 시 하위 레벨에서 여러 개의 Segmentation packet으로 나뉘지고 그 여러 개의 Segmentation packet속에 포함된 data CHUNK SSN필드 값은 동일한 값들을 가지게 된다. Segmentation 된 데이터들은 Server 측 SCTP Level에서 Reassemble 된다. 각각의 packet에 1000byte의 데이터를 실을 수 있다고 가정하고 2048 data를 TCP와 SCTP에 각각 5회씩 전송한다면 TCP에서는 아래와 같이 총 11개의 Data packet이 생성된다.

하지만 SCTP에서는 각 함수 호출이 하나의 stream으로 나타내어 진다. 따라서 $2048/1000=3$ 이므로 $3*5=15$ 즉 15개의 Packet이 생성되어 효율성이 떨어진다. 이를 해결하기 위해서는 CHUNK bundling을 수행하도록 하거나 전송시 버퍼 사이즈를 늘려 주어야 한다.

4.2 Results for Scenario 2

그림 8은 실험 2에 대한 트래픽 측정결과이다. 중단간에 TCP와 SCTP의 데이터를 동시에 100MB를 전송하였다. 그림에서 가로축은 시간을 나타내고 작은 칸 한 칸이 1초를 나타낸다. Y축은 전송된 데이터의 byte를 표현한다. 빨간색은 TCP를 나타내고 녹색 그래프는 SCTP 프로토콜의 데이터를 나타낸다.

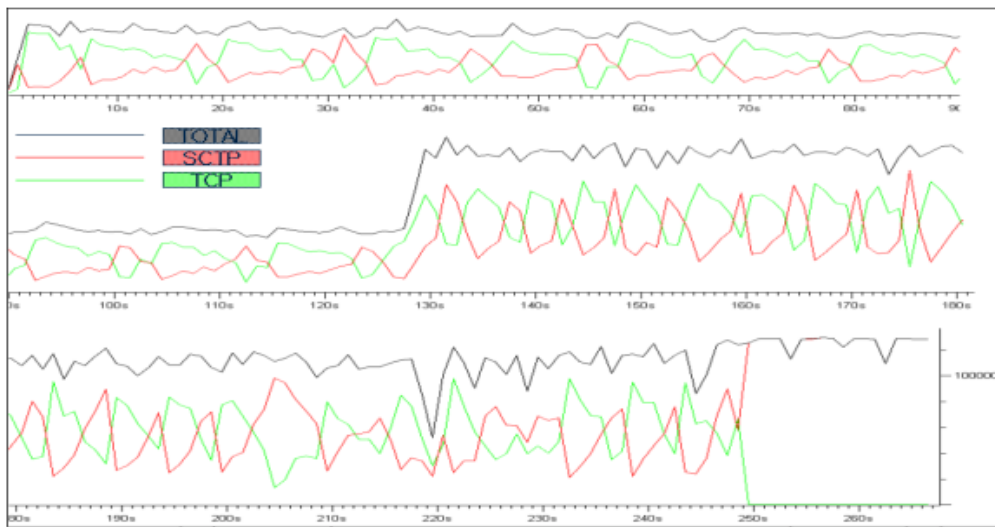


Figure 8. Results for competition of TCP and SCTP traffic

그래프에서 보는 것과 같이 TCP와 SCTP 프로토콜은 같은 망 내에서 서로서로 공정한 경쟁을 하다가 TCP데이터 전송이 끝이 나고 SCTP는 TCP가 끝이 나는 시점에서 SCTP의 전송률은 거의 2배로 증가한다. 이는 TCP와 SCTP의 slow-start, 혼잡 회피, 빠른 재전송으로 이루어지는 윈도우 기반 혼잡 제어 매커니즘을 사용하여 신뢰성 있는 전송을 보장하고 손실된 패킷, 순서에 어긋나게 도착한 패킷, 중복된 패킷을 감지하는데 기인한다.

SCTP는 기본적으로 SACK를 사용하므로 SACK의 전송 경로를 통한 데이터 송수신의 다른 경로 이용에 의한 성능 차이를 볼 수 있다. 먼저 Client에서 Server 쪽으로 접속한 후 100Mb의 데이터를 전송 받았다. Client에서 서버로 접속 시 SCTP는 4way hand shake 메커니즘에 의해 INIT메시지를 보낸다. 이 INIT메시지에는 Client의 address list가 포함되어져 있다. 본 실험 결과 연결이 종료 될 때 까지 client 측의 Primary address는 데이터의 수신 단에만 관련 되어져 수행을 하였다. 192.168.0.14번 IP에서 데이터를 192.168.0.11으로부터 지속적으로 전송 받고 192.168.0.15에서는 receive data에 대한 SACK 데이터만을 전송을 하였다. 이런 Multi-homing 과정을 수행함으로써 packet의 경로 관리가 되어 좀더 효율적인 Performance를 나타낼 수 있다.

4.3 Results for Scenario 3

그림 9는 실험 3에 대한 성능 측정 결과이다.

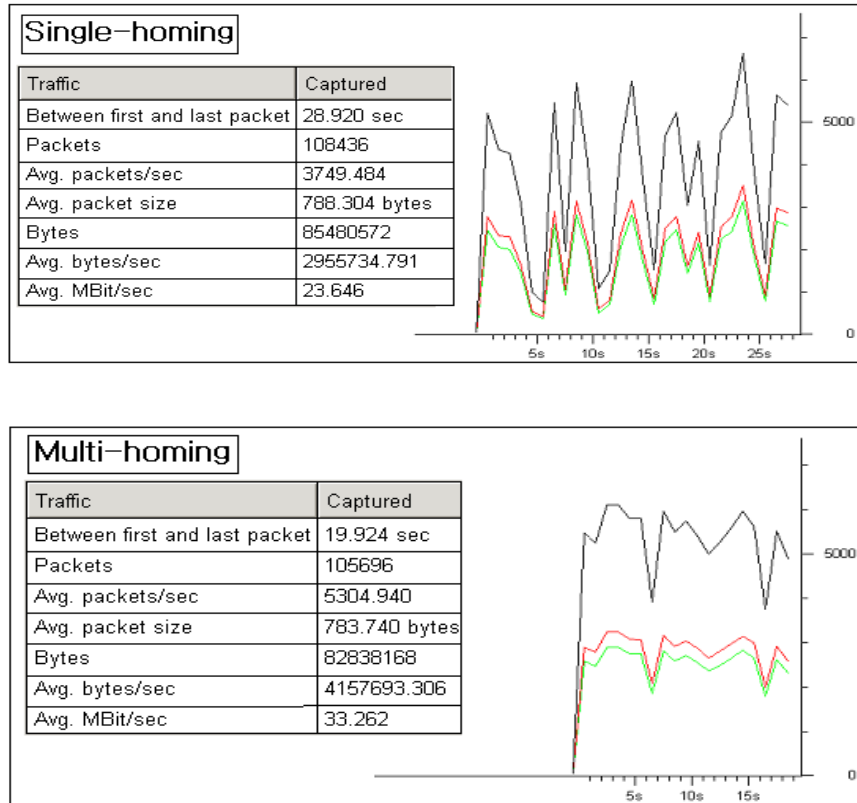


Figure 9. Effects of SCTP single-homing and multi-homing

위의 실험 결과를 보면, x축은 시간 축을 나타내고 y축은 시간에 따른 packet을 나타낸다. 검은색 그래프는 전체 packet 수량을 나타내고 빨간색은 데이터 부분의 패킷 수량을 나타내며 녹색 부분은 SACK와 같은 control Chunk를 나타낸다.

실험결과에서 100Mb를 전송 시 Multi-homing에서의 결과는 20초 정도의 시간이 걸렸고 Single-homing에서는 29초의 정도의 시간이 걸렸다. 그래프를 보면 Single-homing보다 Multi-homing이 확실히 좋은 throughput을 나타낸다. 이는 100Mb의 데이터를 client에서 서버 쪽으로 전송 했을 시에 데이터 패킷 count와 SACK packet count 수가 거의 1:1 비율을 이루고 이 데이터를 처리하기 위해서 같은 경로를 사용하여 송수신하는 Single-homing에서의 throughput보다는 Multi-homing에서의 데이터 송수신의 분리에 의한 송수신 throughput이 복잡한 망에서는 눈에 띄게 좋은 결과를 볼 수 있다.

5. Conclusion

SCTP는 TCP와 마찬가지로 신뢰성 있는 데이터의 전송을 위한 수송계층 프로토콜이다. SCTP는 대부분의 TCP 기능을 가지며 거기에 더해서 **Multi-homing**, **Multi-streaming** 특성을 가진다. 이에 대해 본 연구에서는 지금까지 TCP와 SCTP의 데이터 전송 상의 성능 평가를 하였다.

실험 결과, 일반적으로 SCTP가 TCP보다 **packet** 수량과 데이터량이 많다. 하지만 전송속도 면에서는 데이터수량이 비례하지 않고 내부 흐름제어 메커니즘에 의해 TCP보다 더 좋은 **Throughput** 을 나타내었다. 다만 SCTP 전송 시 **Application Level**에서 **data** 전송 시 **send size**를 신경 쓰지 않는다면 TCP보다 못한 **Throughput**을 나타내다. 이는 **Application** 설계 시 특성에 맞는 **send size**를 이용해야 된다는 것과 TCP와 SCTP의 경쟁 시 공정하게 **Traffic**을 유지함을 알 수 있다. 그리고 **Multi-homing**에서는 여러 개의 **address**를 사용함으로써 데이터 수신경로와 전송경로의 분리가 이루어져서 좀 더 효율적인 전송효율을 얻을 수 있다. 이를 통해서 효율적이 **Multi-homing**에서의 데이터 송수신관리가 전체적인 전송효율에 큰 영향을 미친다는 것을 볼 수 있었다. 하지만 현재와 같은 상황에서는 **association**이 맺어지는 상황에서 데이터 송수신 경로가 확정이 되는데, 네트워크 상태에 따라 능동적인 **path**관리를 하는 방안이 제시 되어야 할 것이다.

향후 연구로써 좀 더 다양한 상황에서의 **application**과 좀더 큰 **HOP**을 가지는 네트워크 환경에서 실험, 분석 작업도 고려 해 볼만하다.

Reference

- [1] R. Stewart, *et al.*, "Stream Control Transmission Protocol," RFC 2960, October 2000
- [2] A. Jungmayer, M. Schopp and M. Tuxen "Performance Evaluation for the Stream Control Transmission Protocol," IEEE ATM Workshop 2000, June 2000
- [3] T. Ravier, *et al.*, "Experimental studies of SCTP multi-homing," First Joint IEI/IEE Symposium on Telecommunications Systems Research, 2001
- [4] Linux Kernel SCTP Project, available from <http://lksctp.sourceforge.net/>
- [5] R. Stewart, *et al.*, "Sockets API Extensions for Stream Control Transmission Protocol", IETF Internet Draft, draft-ietf-tsvwg-sctpsocket-10.txt, Feb. 2005
- [6] Ethereal, available from <http://www.ethereal.com>
- [7] M. Allman, *et al.*, "TCP Congestion Control," RFC 2581, April 1999
- [8] J. Hoe, "Improving the Startup Behavior of a Congestion Control Scheme for TCP," ACM SIGCOMM, August 1996
- [9] Solaris SCTP, Available from <http://playground.sun.com/sctp/>
- [10] SCTPLIB, Available from <http://www.sctp.de/sctp-download.html>
- [11] Network Simulator 2, Available from <http://www.isi.edu/nsnam/ns/>
- [12] Stevens *et al.*, Unix Network Programming: The sockets Networking API, Vol. 1.3rd, 2004
- [14] Linux kernel Archives, Available from <http://kernel.org>
- [15] "Transport Layer Mobility Support Utilizing Link Signal Strength Information", IEICE Transactions on Communications, ISSN: 0916-8516, Vol. E87-B, No. 9, pp. 2548-2556, September 2004.
- [16] "Limiting the Length of BET for Tunnel-Based IP Fast Handover", IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences, ISSN: 0916-8508, Vol. E87-A, No. 6, pp. 1527 - 1530, June 2004.
- [17] "A Transport Layer Mobility Support Mechanism", LNCS 3090, ISSN: 0302-9743, Vol. 3090, pp. 287 - 296, May 2004.
- [18] "mSCTP for Soft Handover in Transport Layer", IEEE Communications Letters, Vol. 8, No. 3, pp. 189 - 191, March 2004.