스트림제어 전송 프로토콜의 개발

이 인 경, *조 은 경
경기공업대학, *대덕대학
전화 : 031-496-4755 / 핸드폰 : 016-790-1585

An Implementation of Stream Control Transmission Protocol

Lee, In-Kyung, Cho, Eun-kyung
Dept. of Computer Information System, Kyonggi Institute of Technology
E-mail : iklee@kinst.ac.kr

Abstract

Generally an increasing number of recent applications have found TCP too limiting. There are some characteristics in the transmission of document and binary data which some transmission delay are tolerant but the content must completely be transferred. However voice signals are more sensitive with not some packet loss but some transmission delay. Therefore, Stream Control Transmission Protocol(SCTP) is proposed to minimize the delay and packet loss in the field of delivery of voice signal. SCTP is designed to transport PSTN signalling messages over IP networks, but is capable of broader applications.
In this paper, the architecture of SCTP implementation is designed and some interface of SCTP software library which are implemented are specified.

I. 서론

스트림 제어 전송 프로토콜(SCTP : Stream Control Transport Protocl)[1,2]은 IP 네트워크 위에 PSTN 신호 메시지를 전송하기 위해 고안된 프로토콜로써 IETF의 SIGTRAN(Signalling Transport)[3,4] 워킹그룹에 의해 표준화되었으며 RFC2960으로 권고하고 있다. 이러한 SCTP는 다음과 같은 특성을 가진다.

- ㅇ 신뢰성 있는 서비스와 TCP와 같은 연결지향 메커니즘을 제공한다.
- ㅇ 시그널링 전송을 위해 채택된 여러 가지 기능을 제공한다.
- ㅇ TCP처럼 엄격한 순서 보장을 원하지 않는 경우 옵션에 따라서 불필요한 지연을 방지할 수 있다.
- ㅇ TCP의 SYN에 의한 해킹에 보다 강한 매커니즘을 가지고 있다.
- ㅇ Multi-streaming과 Multi-homing을 지원한다.

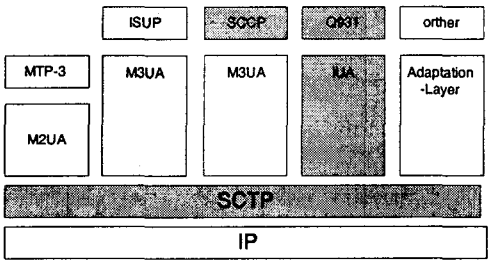
스트림 제어 전송 프로토콜은 기존의 TCP 프로토콜에서 발생할 수 있는 전송지연을 대폭 줄이고 네트워크 오류에 대응하기 위한 해결방법 등을 포함한다. 스트림 제어 전송 프로토콜은 현재 폭발적으로 증가하고 있는 인터넷 전화 서비스 및 멀티미디어 전송 서비스 등에 대한 품질을 보장하기 위해서는 기존의 TCP에서 제공하는 정도의 신뢰성을 보장하면서 TCP 보다 전송 속도를 향상시키고, 네트워크 오류에 대해 보다 유연한 대처를 할 수 있는 새로운 전송 기술의 절실한 필요에 의해 개발되었다.

II. 스트림제어 전송 프로토콜

2.1 시그날링 전송

IETF의 SIGTRAN에서 정의된 프레임워크 안에서

SCTP의 서비스는 Adaptation 계층을 통해 존재하는 시그날링 프로토콜에 유용하다. 그림 1은 여러가지 종류의 Adaptation 계층과 시그날링 응용프로토콜에 있어서의 프로토콜 구조[7]를 보여주고 있다.



M2UA -- MTP-2 User Adaptation Layer
M3UA -- MTP-3 User Adaptation Layer
IUA -- ISDN Q931 User Adaptation Layer
SCCP -- Signaling Connection Control Part

그림 1. SIGTRAN 프로토콜 구조

2.2 SCTP 기능 및 서비스

(1) Association 시작과 종료

Association은 SCTP 유저의 ASSOCIATE (또는 SEND) 프리미티브로부터 시작된다. 쿠키 매커니즘을 이용하여 해킹 등에 대한 공격을 방지하고 시작과정은 TCP의 Three-way handshake 와 달리 Four-way handshake를 사용한다. 또한 SCTP는 SHUTDOWN 프리미티브를 제공하여 SCTP 유저로 부터 활성화된 Association에 대한 정상적인 종료를 가능하게 하고 비정상적인 종료 또한 제공한다.(ABORT 프리미티브)

(2) 스트림내의 순서 제어

SCTP는 Association시작 때 어플리케이션에서 사용할 스트림의 수를 대국과 협상을 통해 확정하고 SCTP유저에 의해 전달된 메시지 각각에 고유한 스트림 아이디를 부여하고 수신한 쪽에서는 주어진 스트림 내에서 시퀀스가 일치 되도록 상위의 유저에게 전송된다.

(3) 유저 데이터 분할과 조립

필요에 따라서 SCTP는 pathMTU에 맞도록 하위 프로토콜로 보내는 메시지를 분할하여 보낸다. 수신한 쪽에서는 상위 SCTP 유저 쪽으로 메시를 전달하기 전에 완전한 메시지로 다시 조립된다.

(4) 전송 확인과 혼잡 제어

SCTP는 각각의 유저 데이터에 TSN (Transmission Sequence Number)를 부여하고 이것은 Stream level에서 부여되는 Stream sequence number와는 독립적이다. 따라서 수신한 쪽에서 데이터이 손실이 발생해도 그 스트림에만 영향을 미치고 다른 스트림에는 영향을 미치지 않는다. 이렇게 함으로써 스트림 레벨에서 신뢰성 있는 전송이 가능해 진다. 전송 확인과 혼잡 제어는 TCP 에서 제공하는 재전송과 같은 방법으로 제공된다.

(5) 청크 번들링

SCTP 패킷은 하나의 공통 헤더에 하나 또는 여러 개의 청크로 이루어 질 수 있는데 각각의 청크는 유저 데이터 또는 제어 정보일 수 있다. SCTP 유저는 하나의 SCTP 패킷에 하나 이상의 청크를 포함하도록 하는 선택권을 가지고 있다. 청크 번들링 기능은 완벽한 SCTP 패킷을 구성하는 기능과 수신단 쪽에서 그것을 분할하는 기능으로 구성 된다.

(6) 패킷 검증

패킷 검증은 크게 2가지로 하나는 Adler-32 책섬이고 다른 하나는 검증 태그를 이용한 것이다. 두가지 모두 공통 헤더의 한 필드로 존재 하는데 Adler-32 책섬은 수신된 패킷에 대한 책섬을 조사해서 유효한 책섬이 아니면 그 패킷을 버리게 된다. 검증 태그는 양단간에 Association 시동을 할 때 주고 받도록 되어 있는데 수신된 패킷의 검증 태그가 유효하지 않으면 그 패킷 전체를 버려야 한다.

(7) 패스 관리

송신하는 SCTP 유저는 전송 주소들을 관리하는데 패스 관리 기능은 SCTP 유저의 지침과 현재 알고 있는 도달 상태를 기준으로 목적지 전송 주소를 선택하게 되는데 패스 관리 기능은 이러한 도달 상태를 SCTP 유저에게 보고하고 관리 되도록 하는 기능이다

III. SCTP 시스템 설계 및 API

3.1 시스템 설계

그림 2는 SCTP시스템의 모듈구조를 보여주고 있다. 각 모듈을 간략히 설명하면 다음과 같다.

- Message Validation and Distribution 모듈 : SCTP 데이터그램을 검증하고 그에 따른 Association을 식
- Path Management 모듈 : 상대편 A-sso ciation의 다른 트랜스포트주소에 대한 도달성을 모니터링하는 모듈

- (De-)Bundling 모듈 : 여러가지 데이터와 제어청크를 하나의 IP패킷 안에 전송되어지는 하나의 SCTP 데이터그램으로 Multiplexing (de-multiplexing)하는 모듈
- Window- and Flow-Control 모듈 : TCP같은 흐름제어와 혼잡회피 방법을 구현하는 모듈
- SCTP Control 모듈 : Association의 상태를 제어하는 모듈
- Reliable Transfer 모듈 : Outgoing 메시지가 Association의 상대편에 의해 통지될 때까지 버퍼링하며 필요한 경우 재전송을 시작하는 모듈
- Reception Control 모듈 : 모든 Incoming 메시지를 추적하고 요구되는 통지관련 제어청크를 생성하는 모듈
- Stream Engine 모듈 : 스트림당 사용자 데이터그램에 대해 전달의 순서성을 실현하고 필요한 경우 큰 사용자 데이터그램의 분할과 조립을 수행하는 모듈

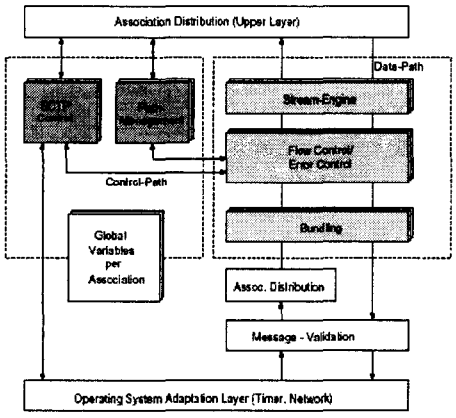


그림 2. SCTP 시스템 구조

3.2 ULP-to-SCTP 인터페이스

SCTP 상위 계층으로부터 SCTP를 사용할 때의 인터페이스로 간단히 설명하면 다음과 같다.

- sctp_initLibrary() : 네트웍으로 부터 SCTP 패킷을 캡처하는 raw 소켓을 열고 타이머목록을 초기화한다. libsctp 라이브러리의 다른 어떤 함수 이전에 호출될 필요가 있다.
- sctp_registerInstance() : 하나의 SCTP 인스턴스를 초기화하기 위해 이 함 수가 호출된다. 한 응응이 여러 가지의 콜백함수를 갖는 여러 가지의 인스턴스를 등록할 수 있다.
- sctp_unregisterInstance() : 이전에 등록된 SCTP 인스턴스에 의해 사용되는 자원을 해제하기 위해 호출된다.
- sctp_associate() : Association을 설정하기 위해 사용되는 함수로 INIT청크를 서버로 송싱하게 하며 Association이 설정될 때 CommunicationUP통지가 호출된다. ULP는 이 Association이 속하는 SCTP인스턴스를 기술해야만 한다.
- sctp_shutdown() : Association을 종료하는 함수로 기본적으로 SHUTDOWN 청크를 송신한다. 종료절차가가 완료된 후 Shutdown Complete 통지가 호출된다.
- sctp_abort() : Association을 강제종료 시키는는 함수이다.
- sctp_send() : ULP에 의해 데이터 청크로써 데이터를 보내는 함수이다.
- sctp_setPrimary() : Association의 프라이머리 패스를 변경한다.
- sctp_receive() : 수신된 데이터를 얻기 위해 Data-Arrive 통지에 대한 응답으로 호출된다.
- sctp_getAssociationDefaults() : SCTP 인스턴스의 모든 기본값을 반환한다. 즉, SCTP_InstancParameters 구조의 내용을 채운다.
- sctp_setAssocDefaults() : 새로운 SCTP인스턴스를 위한 모든 기본값을 설정한다.
- sctp_getAssocStatus() : 어떤 Association에 속하는 많은 값 또는 매개변수를 검색하기 위해 사용될 수 있다. 호출될 때 SCTP_ AssociationStatus구조의 매개변수를 채운다.
- sctp_setAssocStatus() : 어떤 Association에 속하는 많은 값 또는 매개변수를 설정하기 위해 사용될 수 있다.
- sctp_getPathStatus() : 존재하는 Association안에서의 많은 패스에 특정한 값 또는 매개변수를 검색하기 위해 사용될 수 있다.
- sctp_getPrimary() : 현재 프라이머리 주소의 패스 인덱스를 얻기 위해 쉽게 사용될 수 있다.
- sctp_setPrimary() : 새로운 프라이머리 주소를 설정한다.
- sctp_getSrttReport() : 하나의 Association안에서 어떤 목적지 패스에 대한 스무디드 라운드트립 시간을 밀리세컨드로 반환한다.

3.3 SCTP-to-ULP 인터페이스

SCTP로 부터 SCTP 상위 계층으로의 인터페이스로 간단히 설명하면 다음과 같다.

- dataArriceNotif() : 새로운 데이터가 상대방으로부터 도착한 것을 표시한다.
- sendFailureNotif() : 데이타가 어떤 이유로 송신될 수 없음을 표시한다.
- networkStatusChangeNotif() : 네트워크 상태의 변경을 표사함다. 즉, 패스가 SCTP_PATH_OK로부터 SCTP_PATH_ UNRECHABLE, 또는 그 반대로 변경된다.
- communicationUpNotif() : Asociation이 성공적으로 설정되었음을 표시한다.
- communicationLostNotif() : 상대방과의 Association이 어떤 이유로 종료되었음을 표시한다.
- communicationErrorNotif() : 통신오류가 발생했음을 표시한다.
- restartNotif() : Restart가 발생했음을 표시한다.
- shutdownCompleteNotif() : 정상종료가 성공적으로 완료되었음을 표시한다.

IV. 결론

인터넷 전화 서비스와 같이 인터넷 상에서 신호를 전송하는 응용 분야에서 서비스 품질을 떨어뜨리는 가장 심각한 요소는 전송 지연과 패킷분실 현상이다. 전화망에서는 통화하는 두 사람이 통화가 끝날 때까지 하나의 채널을 독점하기 때문에 지연이나 패킷분실 현상이 발생하지 않지만 인터넷에서는 하나의 통로를 통해 수 많은 데이터 패킷들이 전송되기 때문에 지연이나 패킷 분실과 같은 현상이 발생하게 된다. 일반적으로 많은 인터넷 응용들은 데이터 전송에 전송을 보장하는 TCP/IP 프로토콜을 이용하고 있다. 이러한 TCP는 중간에 패킷이 분실되었을 경우 재 전송을 하게 되는데 완벽한 전송을 보장하는 대신 심각한 지연을 유발하게 된다. 문서나 바이너리 데이터를 주고 받는 응용 분야에서는 어느 정도의 지연이 발생하더라도 내용이 완벽하게 전송되어야 하는 특성이 있다. 그러나 음성 신호는 분실 보다는 지연에 더 민감한 특성이 있다. 스트림 제어 전송 프로토콜은 이와 같이 음성 신호의 전송과 같은 서비스 분야에서 전송 지연 및 패킷 분실을 최소화 하기 위한 여러 가지 기능들을 정의하고 있다.

따라서 현재 폭발적으로 증가하고 있는 인터넷 전화와 같은 인터넷 서비스 분야에서 서비스 품질 향상에 획기적인 기여를 할 수 있는 기술이라 할 수 있다. 또한 무선 통신을 위한 새롭고 견고한 신호 전송 기술로 사용될 수 있는 IETF SIGTRAN 운영분과에 의해 고안되어 SS7 신호 프로토콜의 후속 프로토콜로, 단일 연결만으로 여러 개의 데이터 흐름을 유지할 수 있는 용량 성능을 갖고 있어서 무선 휴대폰과 인터넷 이플라이언스의 연결 및 모니터링에 적합하여 이를 이용하여 연결 및 신호 경로를 모니터링할 수 있고 세션 고장 또는 손실을 즉각적으로 탐지할 수 있다.

이 논문에서는 SCTP 프로토콜의 기능, 서비스 개요, 프로토콜의 동작을 살펴보았으며 SCTP시스템의 모듈 구조를 소개하였다. 또한 SCTP가 상위계층과 사용될 때의 인터페이스를 설명하였다.

참고 문헌

[1] R. Stewart, etc, "Stream Control Transmission Protocol", 2000.10.

[2] L. Coene, "Stream Control Transmission Protocol Applicability Statement", 2002.4.

[3] http://www.ietf.org/html.charters/sigtran-charter.html

[4] http://www.ietf.org/home.html

[5] http://www.ietf.org/html.charters/ipngwg-charter.html

[6] http://www.ietf.org/html.charters/ipsec-charter.html

[7] L. Ong etc., "Framework Architecture for Signaling Transport", RFC 2719, 1999.10.

[8] eering, S., and R. Hinden, Editors, "Internet Protocol, Version 6 (IPv6) Specification", RFC 1883, 1995.12.

[9] Atkinson, R., and S. Kent, "Security Archiecture for the Internet Protocol", RFC 2401, 1998.11.

[10] Kent, S., and R. Atkinson, "IP Authentication Header", RFC 2402, 1998. 11.

[11] Kent, S., and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, 1998.11.

[12] D. Maughan, M. Schertler, M. Schneider, and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, 1998.11.

[13] Harkins, D., and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, 1998. 11.