
확장 가능한 전송 프로토콜의 설계

박기현*

*스맥

The Design of Extensible Transport Protocol

Kihyun Park*

*SMEC

E-mail : khpark@esmec.com

요 약

최근의 스마트 폰과 이동통신 망의 발전으로 인하여 모바일 컴퓨팅 환경이 더 이상 보조적인 수단이 아니게 되었다. 그리고 데이터 센터의 활성화로 인하여 클라우드 컴퓨팅 빅 데이터, SDN(Software Defined Network), 그리고 NFV(Network Function Virtualization)과 같은 새로운 서비스 기술들이 나타나고 있다. 이러한 컴퓨팅 환경의 변화와 새로운 서비스 기술들의 도래로 인하여 다양한 요구를 수용할 수 있는 전송 계층 프로토콜이 필요로 하게 되었다. 또한 전송 프로토콜은 추가적인 기술 변화에도 대응할 수 있는 확장 가능한 기능이 요구된다. 본 논문에서는 전송 프로토콜이 확장 가능한 기능을 가지기 위해 기존의 전송 계층을 세 개의 계층으로 나누는 구조를 제안하고 하위 계층에 해당하는 ETP(Extensible Transport Protocol)를 설계하였다.

ABSTRACT

The mobile computing has been a main mean of computing, because of recent development of smartphone and mobile communication network. The new service technologies, such as cloud computing, big data, SDN (Software Defined Network), and NFV (Network Function Virtualization), are appearing due to activating of the data center. These changes have brought various requirements for transport layer protocol. In addition, the transport protocol should have a extensible function to adapt additional technologies. This paper proposed that transport layer divide into three layer structure for extensible function, and designed the ETP(Extensible Transport Protocol) which is located in the bottom layer.

키워드

전송계층, 전송 프로토콜, 인터넷, 프레임워크

1. 서 론

최근 스마트 폰의 대중화로 이동통신 뿐만 아니라 무선 통신 인프라가 급속히 성장하고 있다. 또한 데이터 센터의 활성화로 인하여 클라우드 컴퓨팅, 빅 데이터, SDN(Software Defined Network), 그리고 NFV(Network Function Virtualization)과 같은 새로운 서비스 기술들이 나타나고 있다. 이러한 컴퓨팅 환경의 변화와 새로운 서비스 기술들의 도래로 인하여 다양한 요

구를 수용할 수 있는 전송 계층 프로토콜이 필요로 하게 되었다. 또한 전송 프로토콜은 추가적인 기술 변화에도 대응할 수 있는 확장 가능한 기능이 요구된다.

기존의 전송 계층 프로토콜에 대한 연구는 특정 네트워크 특성에 최적화된 프로토콜을 제안하는데 주안점을 두는 문제가 있었다. 데이터 센터 간의 고속의 데이터 전송을 위한 UDP 기반의 전송 프로토콜에 대한 연구가 있었지만, 고속의 WAN 환경에 목적을 둔 한계가 있었다[1]. 전송

프로토콜의 기능을 작은 마이크로 프로토콜로 나눠서 프로토콜의 설정이나 조합에 따른 확장성을 고려한 연구가 있었으나, 마이크로 프로토콜로 나누기 위한 인터페이스의 제약으로 확장성의 한계가 존재하고 종단 간 프로토콜에 대한 사전 협상 기능이 존재하지 않는 문제가 있었다[2].

II. 전송 계층의 분리

전송 프로토콜의 선택은 응용 프로그램의 효율적인 수행을 위해서 중요하다. 그러나 현재 일반적으로 사용되고 있는 전송 프로토콜은 TCP와 UDP이나 이는 최근의 다양한 네트워크 환경이나 서비스의 요구를 충족시키기에는 한계가 있다. SCTP가 새로운 전송프로토콜로 주목을 받았지만 그 활용은 네트워크 장비들의 SIGTRAN, Diameter 등과 같은 시그널링 프로토콜을 위한 용도에 치우치고 있다.

또한 전송 프로토콜의 데이터 전송 성능은 네트워크 특성에 많은 영향을 받게 된다. 이러한 문제로 전송 프로토콜의 다양한 흐름제어 및 혼잡제어 알고리즘이 제안되어왔지만 단일 네트워크 속성에 특정되어 있었다. 현재의 이동 단말들은 두 개 이상의 인터페이스를 가지고 각기 다른 네트워크 특성을 지니고 있기 때문에 단일 프로토콜로 적용하기에는 한계가 있다.

따라서 본 논문에서는 이러한 문제를 해결하기 위해 네트워크 특성에 따라 각기 다른 전송 프로토콜을 사용할 수 있도록, 다수의 연결을 사용할 수 있는 전송 계층 구조를 제안하였다. 기존의 전송 계층을 연결 제어 계층, 데이터 전송 계층, 그리고 스트림 제어 계층으로 나누었다.

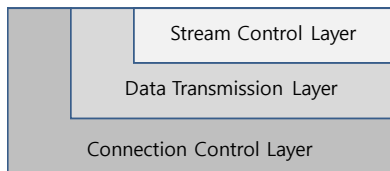


그림 1. 전송 계층의 분리

A. 연결 제어 계층

본 논문은 데이터 전송에 다수의 연결을 사용할 수 있도록 하기 위해, 기존의 전송 프로토콜에서 전송 부분과 연결 제어 부분을 분리하였다. 연결 제어 계층은 데이터 전송을 위한 다수의 연결을 수립하거나 해제하는 기능을 담당한다.

1) 가용 데이터 전송 프로토콜 협상

본 논문에서의 데이터 전송 프로토콜은 특화된 용도에 사용됨을 목적으로 하므로, 다양한 종류의 프로토콜들이 정의될 수 있다. 따라서 종단 호스트 간의 협상을 통해 연결에 사용이 가능한 프로토콜을 선택하는 과정이 필요하다. 연결 제어 계

층은 연결 수립이전에 사용가능한 데이터 전송 프로토콜 리스트를 협상한다.

2) 연결 수립 및 해제

연결 제어 계층은 데이터 전송 프로토콜에 대한 프레임워크를 제공하는 것 또한 목표로하고 있기 때문에, 데이터 전송을 위한 기능을 제외한 부분은 연결 제어 계층을 통한 통일화가 필요하다.

기존의 전송 프로토콜을 연결 제어 부분과 데이터 전송 부분을 분리하여, 각 연결 별로 다른 데이터 전송 프로토콜을 지정하여 사용할 수 있도록 하였다. 연결 수립 과정에서는 연결에 사용하는 데이터 전송 프로토콜을 지정하고, 연결 식별자 생성 및 초기 순서 번호를 교환한다.

B. 데이터 전송 계층

데이터 전송 계층은 특정 응용이나 네트워크 특성에 적합한 데이터 전송 기능을 담당한다. 따라서 이 계층에는 특정 요구조건에 특화된 전송 프로토콜이 존재하게 될 것이다.

기존의 전송 계층이 목표로하는 신뢰성, 흐름 제어, 혼잡제어 등의 기능을 제공한다.

C. 스트림 제어 계층

기존의 전송 프로토콜들은 데이터 스트림이 단일 연결에 종속되어 있었지만, 본 논문에서는 데이터 스트림을 단일 연결에 종속되지 않도록 하기 위해 다른 계층으로 분리 하였다. 이 계층은 아래 그림과 같이 스트림 데이터를 연결에 매핑을 시켜주는 역할을 수행한다. 이 계층을 통해 단일 스트림에 다수의 연결을 활용하면 다중 경로의 효과, 이동성 지원 또한 가능하다.

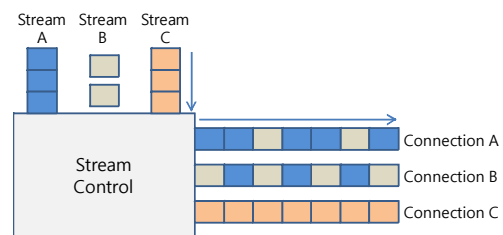


그림 2. 스트림 제어 계층

III. ETP 설계

A. 프로토콜 개요

본 논문에서는 연결제어 계층에 해당하는 ETP(Extensible Transport Protocol)를 제안한다. ETP에서는 데이터 전송을 위한 다수의 연결을 관리하기 위해서 ETP association을 두었다. ETP association에는 상대 호스트와 사용이 가능한 데이터 전송 프로토콜 리스트 정보, 그리고 현재 연

결의 상태 정보를 가진다.

ETP에서의 데이터 전송 과정은 아래와 같은 단계를 따른다.

- 초기 연결 수립 단계
- 데이터 전송 단계(필요 시 추가 연결 수립)
- 연결해제 단계

새로운 전송 계층의 배포에 있어서 큰 장애물은 NAT와 같이 널리 사용되고 있는 미들박스들이다. 미들박스는 제조사에 따라 동작방식이 조금씩 다르며, 경우에 따라서는 새로운 전송 프로토콜 패킷을 폐기하기도 한다. ETP는 이 문제를 해결하기 위해 UDP헤더를 포함하도록 하여, 미들박스에서는 UDP의 처리와 동일하게 하도록 하였다.

B. 패킷 포맷

1) 공통 헤더

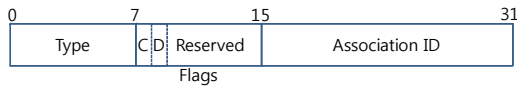


그림 3. 공통 헤더

ETP 패킷은 위 그림과 같이 공통 헤더를 가진다. Type 필드는 ETP 메시지를 식별하기 위한 필드로 메시지 별로 값은 아래와 같다.

- INIT(1), INIT_ACK(2)
- CONN_EST(3), CONN_EST_ACK(4),
- UPDATE_COOKIE(5), UPDATE_COOKIE_ACK(6),
- DATA(7),
- FINISH(8), FINISH_ACK(9), RESET(10)

현재 정의된 flags는 제어 메시지(C), 데이터 전송 메시지(D)를 구분하기 위해 사용된다. 이 값은 상위 데이터 전송 프로토콜에서 제어 정보와 데이터 정보 포함 여부 또한 반영하도록 한다. Association ID값은 ETP Association을 식별하기 위해 사용하며, 각 호스트 별로 고유의 값을 가진다. 유효 범위는 1~65535이며, 예외적으로 INIT/ACK 메시지에는 유효한 Association ID가 존재하지 않으므로 0을 사용한다.

ETP는 UDP 헤더를 기반으로 하기 때문에 별도의 길이, 포트 정보가 존재하지 않는다.

2) INIT 메시지

INIT 메시지는 프로토콜 정보 필드의 리스트 형태를 가지는데, 프로토콜 정보에는 데이터 전송

프로토콜의 식별자(Protocol), 프로토콜의 버전 (Version), 그리고 프로토콜 속성 파라미터 (Protocol Attributes)가 포함될 수 있다. 프로토콜 파라미터는 데이터 전송 프로토콜에 따라 고유의 속성 포맷이 정해진다.

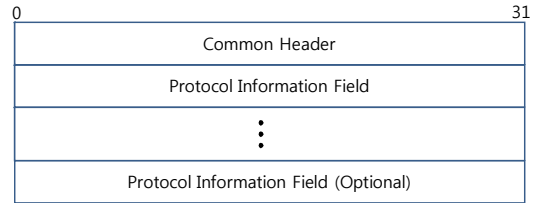


그림 4. INIT 메시지 포맷

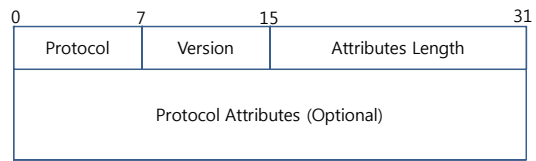


그림 5. 프로토콜 정보 필드

3) INIT_ACK 메시지

ETP는 TCP SYN flooding 공격을 방지하기 위해 SCTP와 동일한 state cookie를 활용한다[3]. INIT_ACK 메시지에 cookie와 cookie lifetime이 포함되며, cookie lifetime 필드는 cookie가 유효한 시간을 초단위로 표현한다. 프로토콜 정보 필드는 INIT 메시지에 포함된 프로토콜 정보 필드 리스트와 일치하는 프로토콜에 대해서만 응답을 한다. 만약 일치하는 프로토콜이 존재하지 않으면 해당 필드가 포함되지 않게 되어 ETP association 생성이 실패하게 된다.

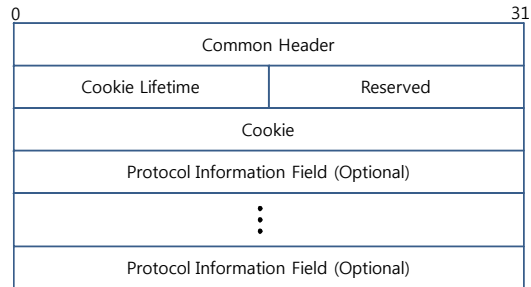


그림 6. INIT_ACK 메시지 포맷

4) CONN_EST 메시지

CONN_EST 메시지는 INIT_ACK에 포함된 cookie의 echo 필드가 포함되며, 연결에 사용될 프로토콜 연결 정보(연결 정보 헤더)가 포함된다. 데이터 전송 계층의 데이터도 포함되는데 MSS(Maximum Segment Size) 그리고 초기 순서 번호와 같은 프로토콜 초기 파라미터에 대한 정

보 교환을 위해 사용된다.

연결 정보 헤더의 connection ID는 ETP association내의 연결을 식별하기 위해 사용되며, 1~65535 유효 범위를 가진다. 0은 사용하지 않는다.

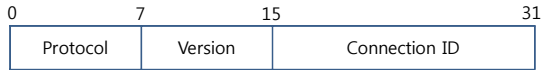


그림 7. 연결 정보 헤더

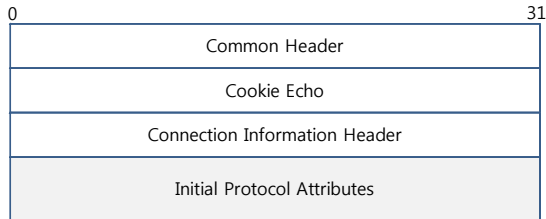


그림 8. CONN_EST 메시지 포맷

5) CONN_EST_ACK 메시지

CONN_EST_ACK 메시지는 cookie 정보를 제외 하고는 CONN_EST 메시지와 동일한 형식을 가진다.

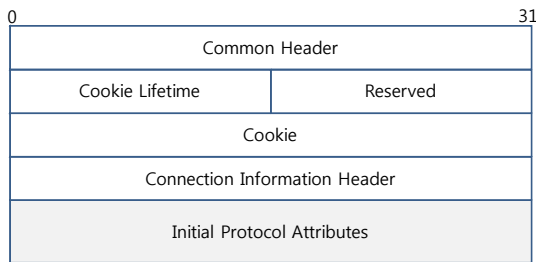


그림 9. CONN_EST_ACK 메시지 포맷

6) UPDATE_COOKIE/ACK 메시지

Cookie lifetime 이후 추가적인 연결 수립을 시도할 경우 이전의 cookie 정보는 유효하지 않게 되므로 실패하게 된다. 이를 방지하기 위해 UPDATE_COOKIE/ACK 메시지를 사용 한다.

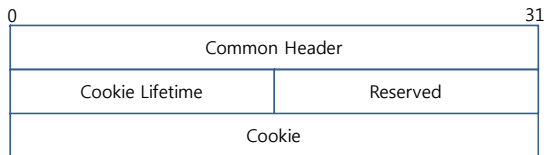


그림 10. UPDATE_COOKIE/ACK 메시지 포맷

7) DATA 메시지

전송 프로토콜의 메시지를 포함하는 메시지며,

포맷은 아래와 같다.

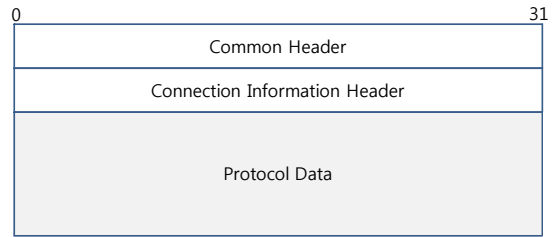


그림 11. DATA 메시지 포맷

8) FINISH/ACK, RESET 메시지

데이터 전송 연결을 해제하기 위해 사용되며, 연결 정보 헤더는 포함되지 않을 수 있다. 연결 정보 헤더가 포함되지 않으면 ETP association 내에 포함된 모든 연결이 해제되며, association 또한 해제된다.

FINISH/ACK 메시지는 단방향 별로 전송을 중단하며, 양방향 중단 시 연결이 해제된다.

RESET 메시지는 유효하지 않는 연결이나 association으로부터 메시지를 수신할 경우 강제로 해제처리하기 위해 사용된다.

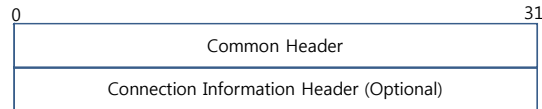


그림 12. FINISH/ACK, RESET 메시지 포맷

C. 초기 연결 수립 절차

ETP는 SYN flooding 공격을 방지하기 위해, 가용 데이터 전송 프로토콜 협상 과정과 초기 연결 수립절차를 통합하여 진행한다. 따라서 ETP association과 연결 정보 생성 시점은 연결 수립 단계에서 이뤄지며, 초기 연결 수립이 실패하게 되면 association 또한 해제된다.

아래 그림에서 Host A는 INIT_ACK 수신 후 ETP association과 연결 정보를 생성하여 해당 정보를 CONN_EST 메시지에 포함하여 전송한다. Host B는 CONN_EST 메시지 수신 시 ETP association과 연결 정보를 생성한다.

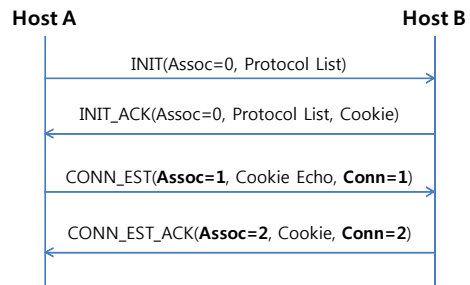


그림 13. 초기 연결 수립 예

D. 추가 연결 수립

추가 연결 수립을 위해서는 CONN_EST와 CONN_EST_ACK 메시지 송수신을 통해 가능하다. 추가 연결이 필요한 응용일 경우 UPDATE_COOKIE 메시지를 통해 항상 유효한 cookie 값을 유지하도록 하여야 한다. 그렇지 않을 경우 추가 연결 시도는 RESET 메시지를 통해 실패하게 된다.

E. 데이터 전송

데이터 전송은 연결 수립 시 지정한 데이터 전송 프로토콜을 통해 전송된다. ETP에서 제공하는 헤더 정보를 통해 수신 측 호스트에서 처리할 데이터 전송 프로토콜이 지정될 수 있도록 한다.

F. 연결 해제

정상적인 연결 해제는 FINISH 메시지를 통해 이뤄지며 방향 별 전송이 중단된다. 아래 그림과 같이 Host A에서 FINISH 처리를 하더라도 Host B에서 송신하는 데이터는 처리가 가능하다. 단, 데이터 전송 계층에서의 ACK 메시지로 유추되는 C flag만 설정된 DATA 메시지는 FINISH 처리가 되더라도 송신이 가능하다.

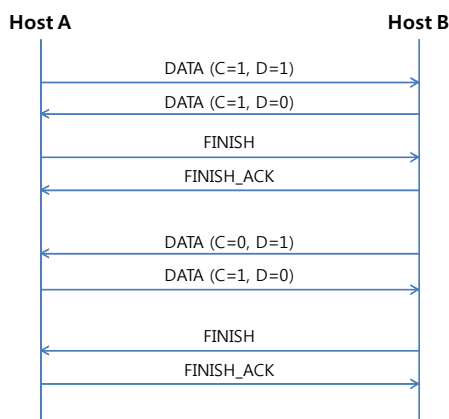


그림 14. 연결 해제 처리

IV. 결론

기존의 전송 프로토콜의 연구는 주로 단일 네트워크 환경에 대한 해결책을 제시하였다. 그러나 최근의 무선 통신 기술의 발달과 클라우드 컴퓨팅의 활성화는 단일 서비스에 대한 경험이 다양한 네트워크를 통한 접근으로 가능하게 되었다. 본 논문에서는 이러한 다양한 네트워크 환경과 요구를 만족시키기 위해 확장이 가능한 전송 계층 구조를 제안하였다.

또한 연결 제어 계층에 해당하는 ETP 프로토콜을 설계하였다. ETP는 UDP를 기반으로 하기

때문에 적용이 쉽고, 응용 계층에서 쉽게 구현이 가능한 장점을 가진다.

향후 추가 연구 과제로 데이터 전송 계층의 프로토콜에 대한 연구가 필요하며, 스트림 제어 계층에서 고속 전송을 위한 스케줄링 그리고 이동성 지원을 위한 알고리즘에 대한 연구가 필요하다.

참고문헌

- [1] Yunhong Gu, Robert L. Grossman, "UDT: UDP-based Data Transfer for High-Speed Wide Area Networks," Computer Networks(Elsevier). Volume 51, Issue 7. May 2007.
- [2] Patrick G. Bridges, et al., "A Configurable and Extensible Transport Protocol," IEEE/ACM Transactions on Networking, Vol.15, No.6, December 2007.
- [3] R. Stewart, et al., "Stream Control Transmission Protocol," IETF RFC4960, September 2007.