

10 전송 계층

가천대학교 - 2019학년도 1학기 -

❖ 전송 계층 프로토콜

- UDP : 비연결형 서비스를 지원하는 프로토콜.
 - 데이터 신뢰성 문제로 일반 응용 프로그래머가 덜 선호함
 - 작고 가벼워 빠른 전송이 필요한 환경에서 사용
 - 특히, 데이터를 실시간으로 전송하는 환경에서 TCP보다 유리한 점이 많음
- RTP 프로토콜 : 실시간 서비스를 제공하는 프로토콜

Contents

❖ 학습목표

- 비연결형 서비스를 제공하는 UDP의 헤더와 데이터 전송 방법을 이해한다.
- 실시간 데이터 전송 프로토콜을 살펴본다.
- RTP의 헤더와 동작 원리를 이해한다.
- OSI TP의 서비스 프리미티브 종류와 동작을 이해한다.

❖ 내용

- UDP 프로토콜
- RTP 프로토콜
- OSI TP 프로토콜

01_UDP 프로토콜

- UDP User Datagram Protocol : 인터넷 프로토콜 중 구조가 가장 간단
- 특징
 - 비연결형 서비스를 제공
 - 헤더와 전송 데이터에 대한 체크섬 기능을 제공
 - Best Effort 전달 방식을 지원
빠른 시간내의 전달을 최우선으로 하는 전달 방식. 네트워크는 분실되거나 손상된 패킷을 복구하는 등의 복잡한 기능은 제공하지 않는 것으로 이러한 기능을 제거함으로써 네트워크를 효율적으로 운용할뿐만 아니라 사용자에게는 신속한 전달을 보장할 수 있게 된다.
- 신뢰성이 떨어지지만 프로토콜을 처리하는 기능이 작아 TCP보다 데이터 처리가 빠르므로 데이터 전송 시간에 민감한 응용 환경에서는 UDP를 사용하는 것이 유리

01_UDP 프로토콜

❖ UDP 헤더 구조

- 프로토콜의 오버헤드가 작은 편임



그림 10-1 UDP 헤더의 구조

- Source Port/Destination Port(송신 포트/수신 포트) : 송수신 프로세스에 할당된 네트워크 포트 번호
- Length(길이) : 프로토콜 헤더를 포함한 UDP 데이터그램의 전체 크기
 - 단위는 바이트. 헤더의 최소 길이가 8바이트이므로 Length의 최소값은 8
 - 이론상 최대 65,535($2^{16}-1$) 바이트까지 가능하나, 응용프로그램 및 다른 계층과의 연관성 문제를 고려하여 일반적으로 8,192바이트를 넘지 않게 사용함
- Checksum(체크섬) : 프로토콜 헤더와 데이터에 대한 체크섬 값을 제공
 - 수신 프로세스는 체크섬 오류가 발견되면 해당 데이터그램을 버림
 - 체크섬 기능은 옵션으로 필드 값이 0이면 송신 프로세스가 체크섬을 계산하지 않았다는 의미

❖ UDP의 데이터그램 전송

- 비연결형 서비스를 이용하여 데이터그램을 전송
- 각 데이터그램은 전송 과정에서 독립적으로 중개됨
- 데이터그램이 목적지까지 도착하는 것을 보장하지 않음
- 흐름 제어 기능이 없어 버퍼 오버플로(Buffer Overflow)에 의한 데이터 분실 오류가 발생할 수 있음
- 오류 유형
 - 데이터가 목적지에 도착하지 못하는 데이터그램 분실과
 - 데이터그램의 도착 순서가 바뀌는 도착 순서 변경

01_UDP 프로토콜

■ UDP에서의 데이터그램 분실

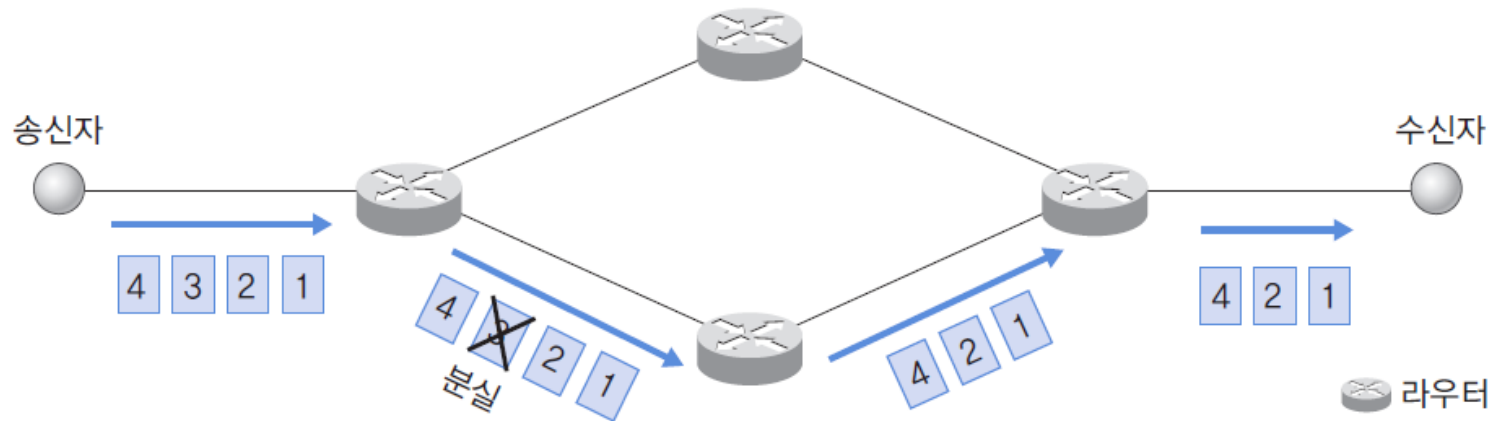


그림 10-2 데이터그램 분실

- 분실 오류를 복구하는 기능을 수신하지 않으므로 데이터 분실을 인지하지 못함
- UDP의 데이터그램 분실 오류는 상위 계층 스스로 데이터 분실을 확인해 복구해야 함
- 데이터의 순서 번호 기능이 없으므로 응용 프로그램에서 순서 번호와 유사한 기능을 프로그램 내부에서 구현해야 함

01_UDP 프로토콜

- UDP에서의 데이터그램 도착 순서 변경
 - 데이터의 순서 번호 기능이 없으므로 응용 프로그램에서 순서 번호와 유사한 기능을 프로그램 내부에서 구현해야 함

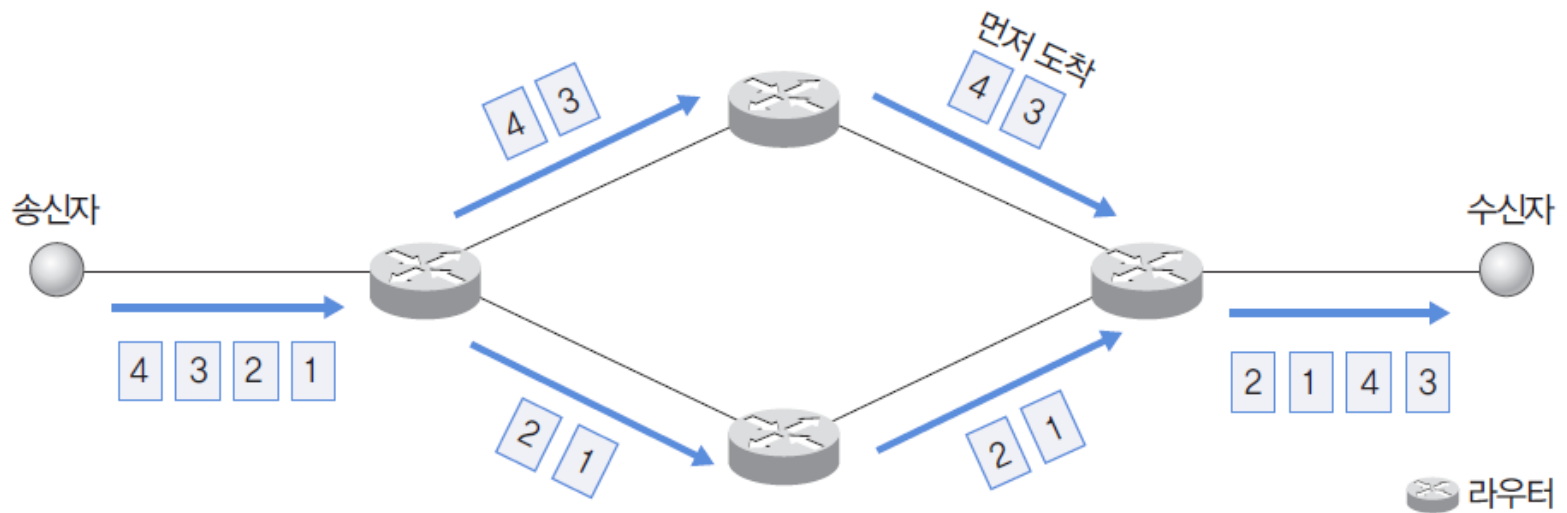


그림 10-3 도착 순서 변경

02_RTP 프로토콜

❖ 인터넷 멀티미디어 서비스

- 비디오, 오디오 파일 전체를 하나의 단위로 다운로드 하여 재생
=> 실시간으로 다운로드하며 재생
- TV, 라디오, 영화 등 동영상 서비스에도 일반화
- 인터넷 음성전화 서비스 보급, 기존 전화망 연동 서비스

❖ 실시간 요구 사항

- 데이터 변형/분실 오류를 복구하는 기능이 상대적으로 덜 중요함
- 데이터그램의 도착 순서, 패킷의 지연 간격 분포의 균일성, 데이터 압축에 의한 정보 전송량의 최소화 등이 중요
- TCP : 패킷의 순서와 신뢰성이 지나치게 강조되어, 재전송 기능, 복잡한 흐름 제어 기능으로 인해 실시간 환경에 부적합
- UDP : 기능이 단순하여 빠른 데이터 전송을 지원하지만, 데이터그램의 순서를 보장하지 못함
- 가장 현실적인 방법 중 하나 : UDP에 데이터의 순서 번호 기능 추가

02_RTP 프로토콜

❖ RTP Real Time Protocol

- 실시간 멀티미디어 데이터의 전송을 지원
- 유니캐스팅뿐 아니라 멀티캐스팅도 지원
- 특징
 - 불규칙한 데이터의 순서를 정렬하기 위해 타임스탬프(Timestamp) 방식을 사용
 - 응용 프로그램의 라이브러리 형태로 구현되는 ALF(Application Level Framing) 방식으로 응용 환경이 요구하는 알고리즘에 따라 버퍼 크기를 개별적으로 조절 가능
 - 실시간 응용 서비스에 유용하지만, 자원 예약이나 QoS 보장과 같은 기능을 제공하지 못하므로 실시간 동영상 서비스를 지원하기에는 한계가 있음

❖ 실시간 요구사항

- 전송 시간이 중요 : 송신 프로세스가 전송한 데이터의 전송 간격 유지, 대부분 특정 데이터가 정해진 시간 안에 반드시 도착하도록 요구
- 버퍼의 역할
 - 네트워크에서 데이터의 시간 간격이 불규칙적으로 변함
 - 수신 프로세스의 버퍼를 이용하여 시간 간격이 일정하도록 보정

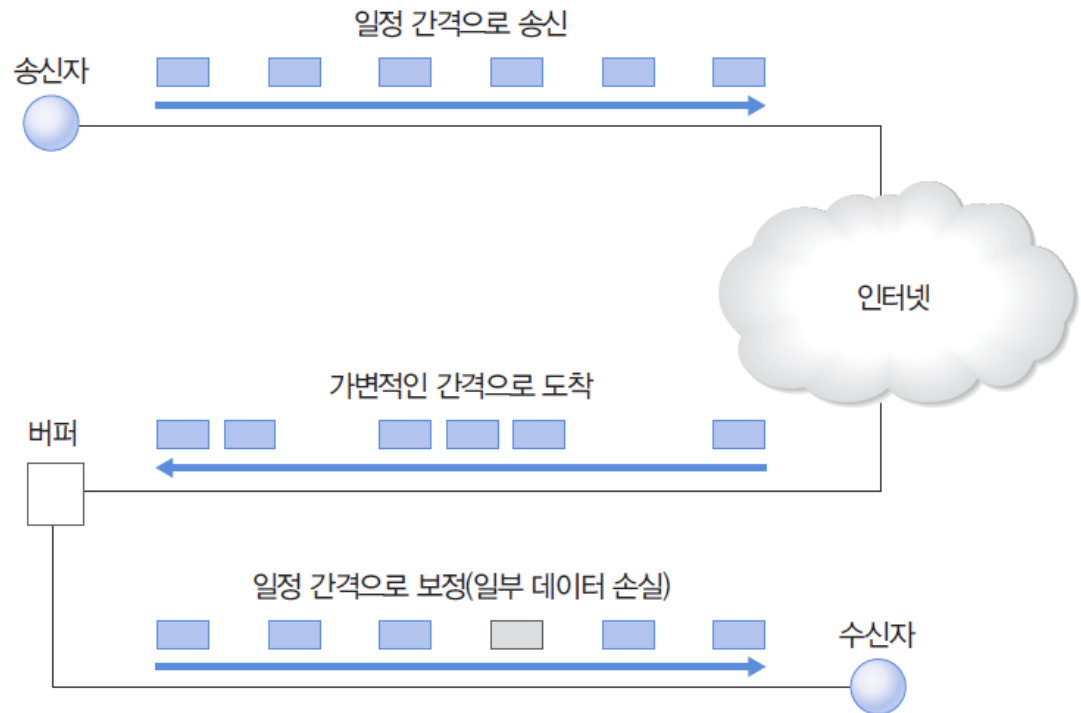


그림 10-4 실시간 전송

02_RTP 프로토콜

■ 지터

- 지터^{Jitter} 분포 : 데이터그램의 도착 시간을 측정하였을 때 각 데이터그램의 도착 시간이 불규칙적으로 도착하는 정도를 나타냄

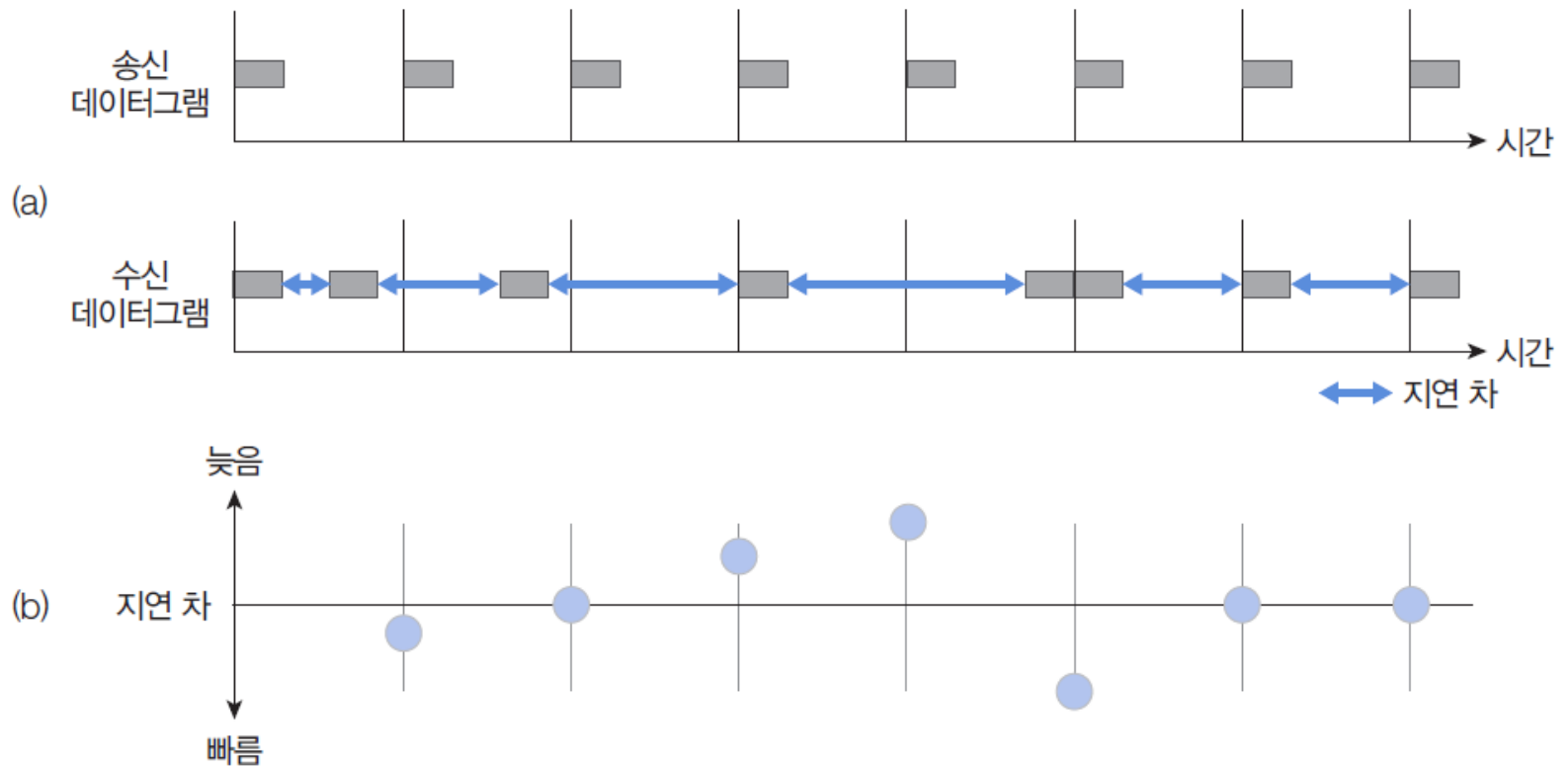


그림 10-5 지터 분포

02_RTP 프로토콜

❖ RTP의 데이터 전송

- 실시간 서비스를 위해 작고 빠른 전송 기능을 제공하는 UDP 위에서 구현
 - 데이터그램 분실이나 도착 순서 변경과 같은 전송 오류는 RTP 자체에서 해결
 - 송수신 프로세스 간의 연결을 관리(UDP의 포트 번호 사용)
- 하나의 완전한 프로그램 단위로 구현되지 않고, 기능별로 개별로 구현
- 다수의 사용자가 하나의 세션에 참여, 서로 실시간 데이터 전송을 지원

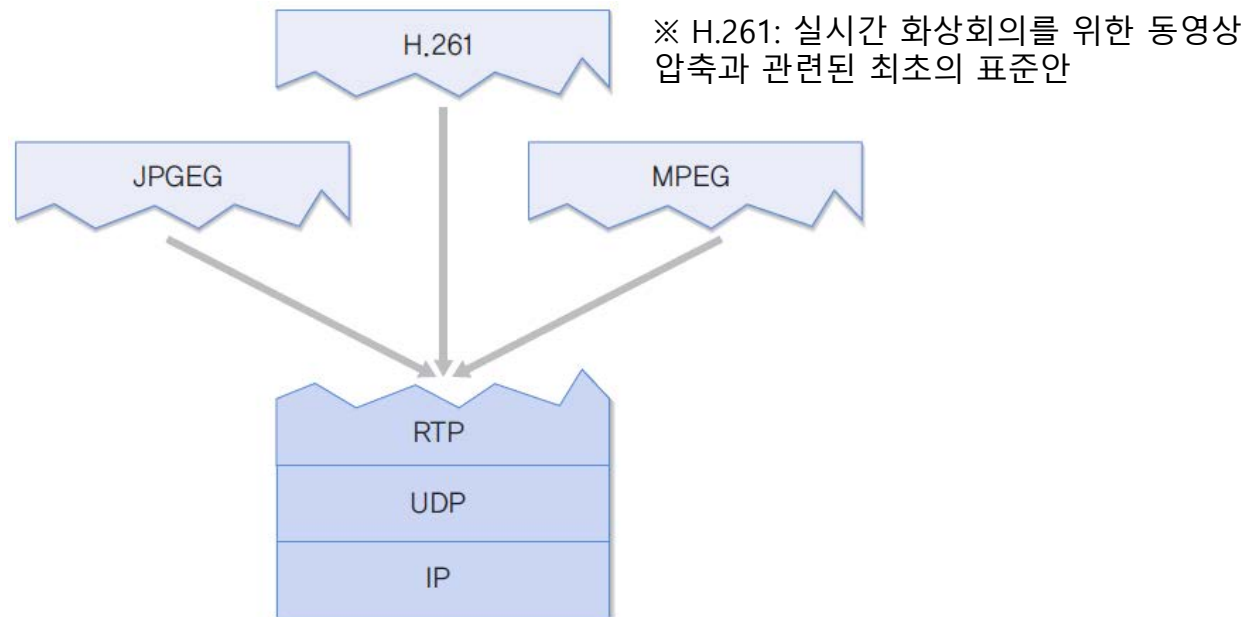


그림 10-6 RTP 구조

02_RTP 프로토콜

■ 두 종류의 RTP 릴레이(RTP Relay)를 지원

- 릴레이는 데이터 전송 과정에서 송수신 프로세스가 데이터를 직접 전송할 수 없는 상황이 발생했을 때, 데이터를 중개하는 기능
 - 예) 송수신 프로세스 사이에 방화벽이 설치되거나 데이터 형식이 다를 경우, 릴레이가 필요한 처리를 수행, 데이터 전송이 가능하도록 지원
- 믹서(Mixer) : RTP 데이터그램 스트림을 받아 이들을 적절히 조합하여 새로운 데이터그램 스트림을 생성
 - 데이터 형식이 변하거나 믹싱 기능이 수행될 수 있음
 - 여러 송신 프로세스로부터 수신한 데이터의 시간 관계 조절을 위해 조합된 데이터그램 스트림에 시간 정보 제공 및 시간 정보(Synchronization) 제공여부 표시
- 트랜슬레이터(Translator) : 입력된 각 RTP 데이터그램을 하나 이상의 출력용 RTP 데이터그램으로 만들어주는 장치
 - 데이터 형식 변경 가능
 - 예1: 임의의 수신자 그룹에서 특정 수신 프로세스가 고해상도 비디오 신호를 처리할 능력이 없는 경우 트랜슬레이터가 저해상도 신호로 변환하여 수신 프로세스가 처리할 수 있도록 지원
 - 예2: 입력된 멀티캐스트 RTP 데이터그램을 복사하여 다수의 유니캐스트 수신 프로세스에 전송

02_RTP 프로토콜

❖ RTP 헤더 구조

- 기본 헤더에 응용 환경과 관련된 가변 크기의 헤더 추가 가능
- CSRC 구분자 목록은 믹서에 의해 추가되는 경우에 사용
- 멀티캐스트 전송 가능
 - RTP 데이터 형식에 송신 구분자 Source Identifier 필드 존재 : 멀티캐스트 전송을 위해 멀티캐스트 그룹에서 누가 데이터를 전송했는지 확인
 - Timestamp 필드 지원 : 수신프로세스에서 지연 버퍼를 사용해 타이밍 관계 조절

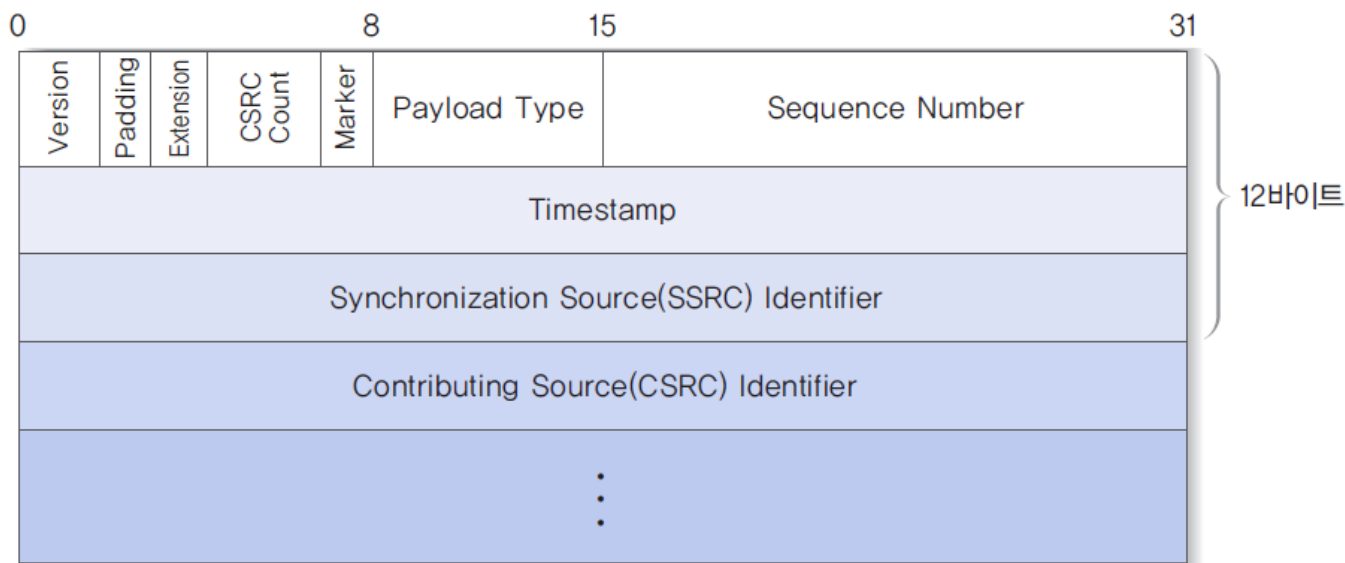


그림 10-7 RTP 고정 헤더의 구조

02_RTP 프로토콜

■ RTP 헤더에 정의된 각 필드의 의미

- Version(버전) : RTP의 버전 번호
현재 2로 지정
- Padding(패딩) : RTP 페이로드의
마지막에 패딩 데이터가 존재하는지 여부
- Extension(확장) : 고정 헤더의 마지막에 확장 헤더가 하나 더 이어짐을 의미
- CSRC Count(CSRC 개수) : CSRC 구분자의 개수를 표시
- Marker(표식) : 임의의 표식^{Marking}, 페이로드 유형에 따라 값의 의미가 결정됨
 - 보통 데이터 스트림의 경계를 표시하는데 사용. 예) 비디오 페이로드에서 프레임의 마지막을 표시하기 위해 1로 지정
- Payload Type(페이로드 유형) : 헤더 다음에 이어지는 RTP 페이로드의 유형
- Sequence Number(순서 번호) : Timestamp 필드 값이 동일한 페이로드에 대해
패킷 손실이나 순서 변경과 같은 오류 검출
 - 일반적으로 동시에 생성된 일련의 연속 패킷들은 동일한 Timestamp 값을 가지며, 순서번호는 RTP 패킷 단위로 1씩 증가
- Timestamp(타임스탬프) : RTP 페이로드에 포함된 데이터의 생성 시기
- SSRC Identifier(SSRC 구분자) : 임의의 세션 내에서 RTP 페이로드의 발신지가
어디인지를 구분하는 고유 번호. 랜덤하게 생성되는 32비트 숫자

0					8			15			31
Version	Padding	Extension	CSRC Count	Marker	Payload Type			Sequence Number			
Timestamp											
Synchronization Source(SSRC) Identifier											
Contributing Source(CSRC) Identifier											
⋮											

02_RTP 프로토콜

표 10-1 RFC 1890에서 권고한 표준 오디오 · 비디오 인코딩

페이로드	인코딩	페이로드	인코딩	페이로드	인코딩
0	PCMU audio	10	L16 audio	28	nv video
1	1016 audio	11	L16 audio	31	H.261 video
2	G.721 audio	12~13	audio	32	MPV video
3	GSM audio	14	MPA audio	33	MP2T video
4	audio	15	G.728 audio	34~71	
5	DV14 audio	16~23	audio	72~76	Reserved
6	DV14 audio	24	video	77~95	
7	LPC audio	25	CelB video	96~127	Dynamic
8	PCMA audio	26	JPEG video		
9	G.722 audio	27			

02_RTP 프로토콜

❖ RTP 제어 프로토콜(RTCP RTP Control Protocol)

■ 주요 기능

- QoS Quality of Service와 혼잡 제어 : 데이터 분배 과정에서 발생하는 서비스 품질에 관한 피드백 기능을 지원. 송수신 보고서에 전송률, 패킷 분실, 지터 등 정보 포함
- Identification(구분자) : RTCP 송신 프로세스에 관한 구분자 정보가 포함, 서로 다른 세션에서 발신된 스트림 정보들을 서로 연관시키는 근거를 제공
- 세션 크기 : 전체 세션 트래픽의 5% 이내로 유지되도록 알고리즘이 동작

표 10-2 패킷의 종류와 역할

종류	역할
Sender Report ^{SR} , Receiver Report ^{RR}	데이터 전송 품질을 피드백하기 위한 용도로 사용된다.
Source Description ^{SDES}	송신 프로세스가 자신에 대한 정보를 더 많이 제공하는 용도로 이용된다.
Goodbye ^{BYE}	송신 프로세스가 더 이상 존재하지 않음을 의미하고, 이는 수신 프로세스가 송신 프로세스를 무한정 기다리지 않도록 한다. 즉, 상대방과의 통신상 문제가 네트워크 오류가 아닌 세션에 더 이상 참가하지 않기 때문임을 알려준다.
오타? Applicationdefined Packet	응용 환경에 따른 기능을 점검하기 위해 제공된다.

03_OSI TP 프로토콜

- OSI에서 정의한 TP 프로토콜이 제공하는 서비스
 - 클래스 0이 구조가 가장 단순, 클래스 번호가 커질수록 기능이 추가

표 10-3 OSI TP 프로토콜이 제공하는 서비스

클래스	제공하는 서비스
클래스 0	기본 기능
클래스 1	기본 오류 복구 기능
클래스 2	멀티플렉싱 기능
클래스 3	오류 복구, 멀티플렉싱 기능
클래스 4	오류 검출, 오류 복구, 멀티플렉싱 기능

03_OSI TP 프로토콜

❖ OSI TP의 서비스 프리미티브

- 연결형 서비스 : 연결 설정(T-CONNECT), 연결 해제(T-DISCONNECT)
일반 데이터(T-DATA), 긴급 데이터(T-EXPEDITED-DATA)
- 비연결형 서비스 : 데이터 전송을 위한 T-UNITDATA 프리미티브만 존재

표 10-4 OSI TP의 서비스 프리미티브

프리미티브	제공 서비스	프리미티브	제공 서비스
T-CONNECT.request	연결 설정	T-DATA.request	데이터 전송
T-CONNECT.indication	연결 설정	T-DATA.indication	데이터 전송
T-CONNECT.response	연결 설정	T-EXPEDITED-DATA.request	긴급 데이터 전송
T-CONNECT.confirm	연결 설정	T-EXPEDITED-DATA.indication	긴급 데이터 전송
T-DISCONNECT.request	연결 해제	T-UNITDATA.request	비연결형 데이터 전송
T-DISCONNECT.indication	연결 해제	T-UNITDATA.indication	비연결형 데이터 전송

[참고] 계층구조의 개념

❖ 서비스 프리미티브

- 계층 구조 프로토콜에서 하위 계층이 상위 계층에 제공하는 서비스의 종류에는 연결형과 비연결형이 있으며 프리미티브 형태로 구현됨
- 연결형 서비스
 - 연결형^{Connection-oriented} 서비스를 이용하는 3단계

표 2-1 연결형 서비스의 프리미티브 종류

종류	용도
CONNECT	연결 설정
DATA	데이터 전송
DISCONNECT	연결 해제

- 비연결형 서비스
 - 전송할 데이터가 있으면 각 데이터를 독립적으로 목적지 호스트로 전송

[참고] 계층구조의 개념

■ 서비스 프리미티브의 기능

표 2-2 서비스 프리미티브의 기능

기능	설명
Request	클라이언트가 서버에 서비스를 요청함
Indication	서버에 서비스 요청이 도착했음을 통지함
Response	서버가 클라이언트에 서비스 응답을 회신함
Confirm	클라이언트에 응답이 도착했음을 통지함

■ 서비스 프리미티브의 동작 원리 [그림 2-5]

- Request : 연결 설정 요청 `CONNECT.Request`, 데이터 전송 요청 `DATA.Request`,
연결 해제 요청 `DISCONNECT.Request` 등
- Indication : 연결 설정, 데이터 전송, 연결 해제에 대해 `CONNECT.Indication`,
`DATA.Indication`, `DISCONNECT.Indication` 순으로 사용
- Response : 연결 설정 요청은 `CONNECT.Response`, 데이터는 `DATA.Response`,
연결 해제는 `DISCONNECT.Response`로 전달

[참고] 계층구조의 개념

- Confirm : 연결 설정은 CONNECT.Confirm, 데이터는 DATA.Confirm, 연결 해제는 DISCONNECT.Confirm로 전달

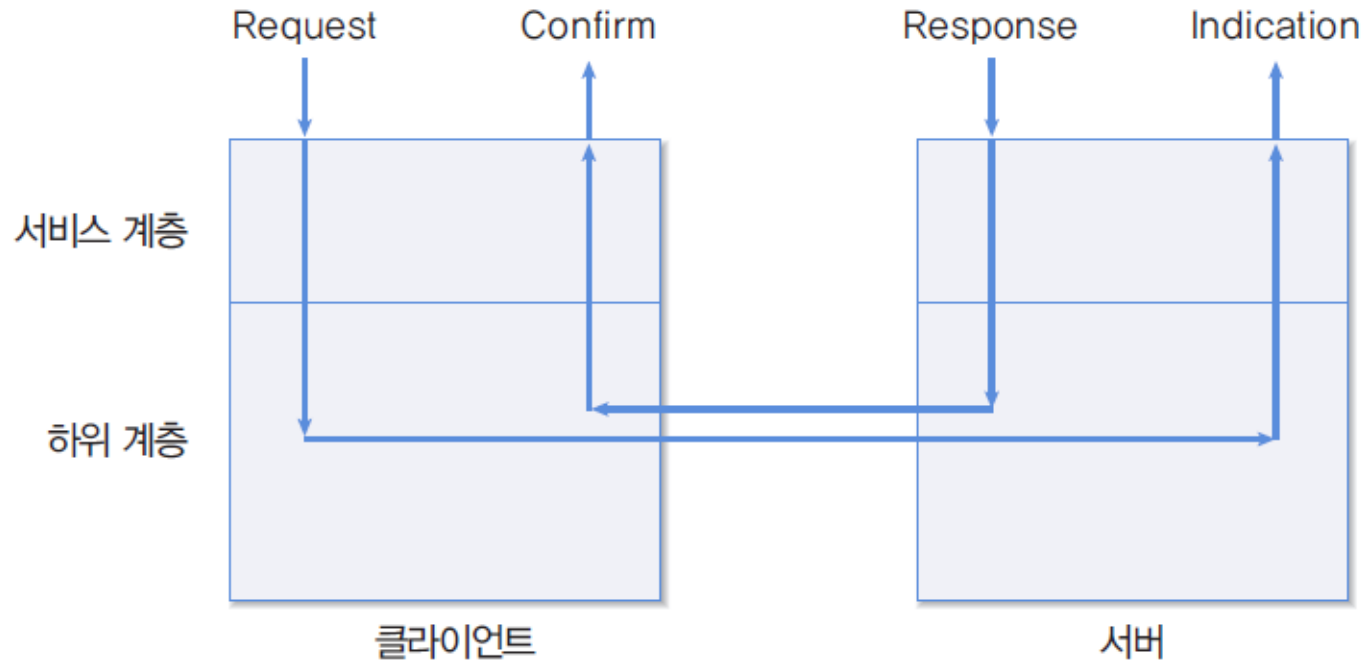


그림 2-5 서비스 프리미티브의 동작 원리

❖ OSI TP의 데이터 전송

- T-DISCONNECT(연결 해제)
 - 어느 한쪽이라도 연결 해제를 원하면 해제
 - 네트워크 내부에 특별한 상황이 발생시 해제

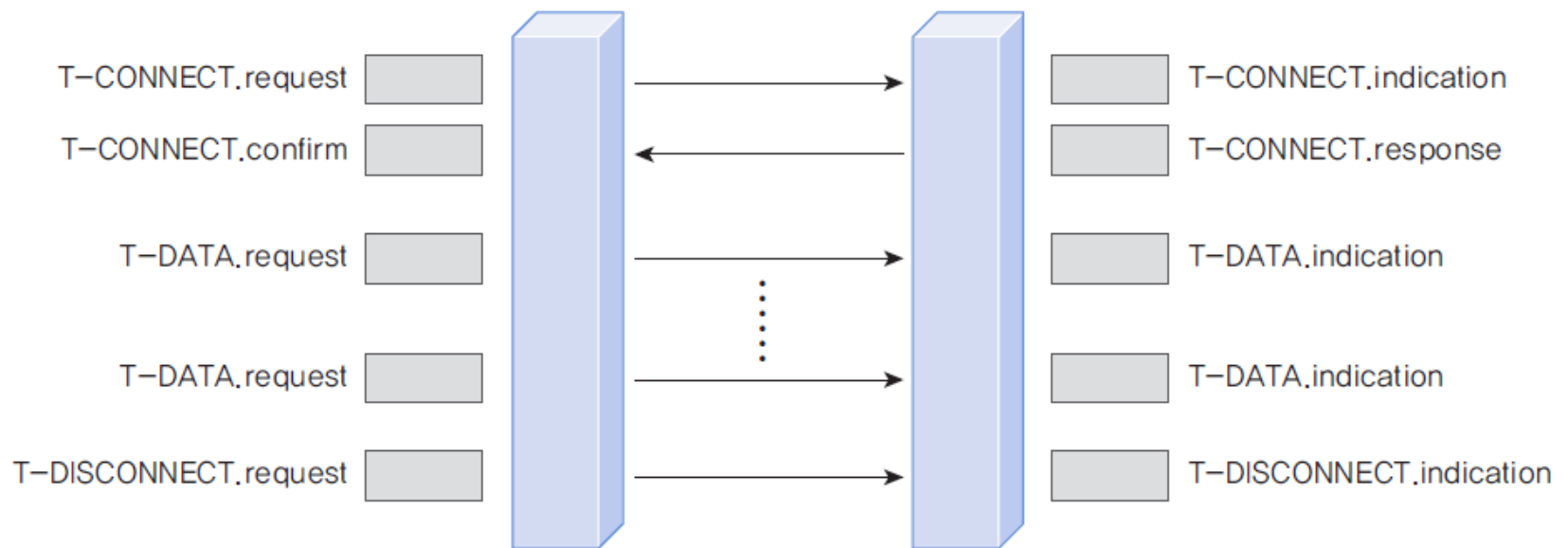


그림 10-8 OSI 프리미티브



Thank You
