



10 전송프로토콜

전송 프로토콜과 UDP

TCP프로토콜

1

2

3

4

5

6

7

8

9

10



◆ TCP의 기능

» TCP는 프로세스간 통신, 스트림 전달 서비스, 전이중 통신, 연결 지향 서비스, 신뢰성 있는 서비스를 제공

- 연결설정과 해제과정이 존재
- 데이터의 안전한 도착을 확인하는 **ACK를 사용**하여 신뢰성 있는 전송을 제공

» TCP는 프로세스 간의 통신을 위해 **포트 번호를 사용**

◆ TCP의 기능

》 Well-known ports used by TCP

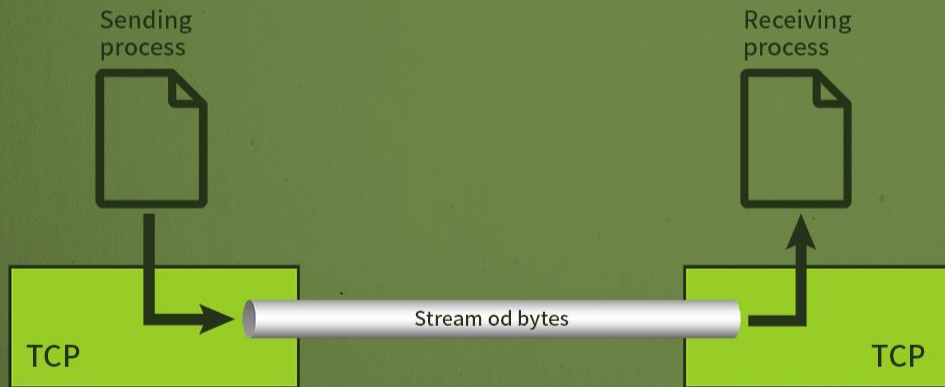
Port	Protocol	Description
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20	FTP,Data	File Transfer Protocol(Data connection)
21	FTP, Control	File Transfer Protocol(Control connection)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Service
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	HyperText Transfer Protocol
111	RPC	Remote Procedure Call



◆ TCP의 기능

» 스트림 전달 서비스

- TCP는 데이터를 **바이트의 나열로 전달**
 - 효율성을 위해 **여러 바이트를 블록으로 구성된 세그먼트를 만들어 전송**





◆ TCP의 기능

» TCP 특징

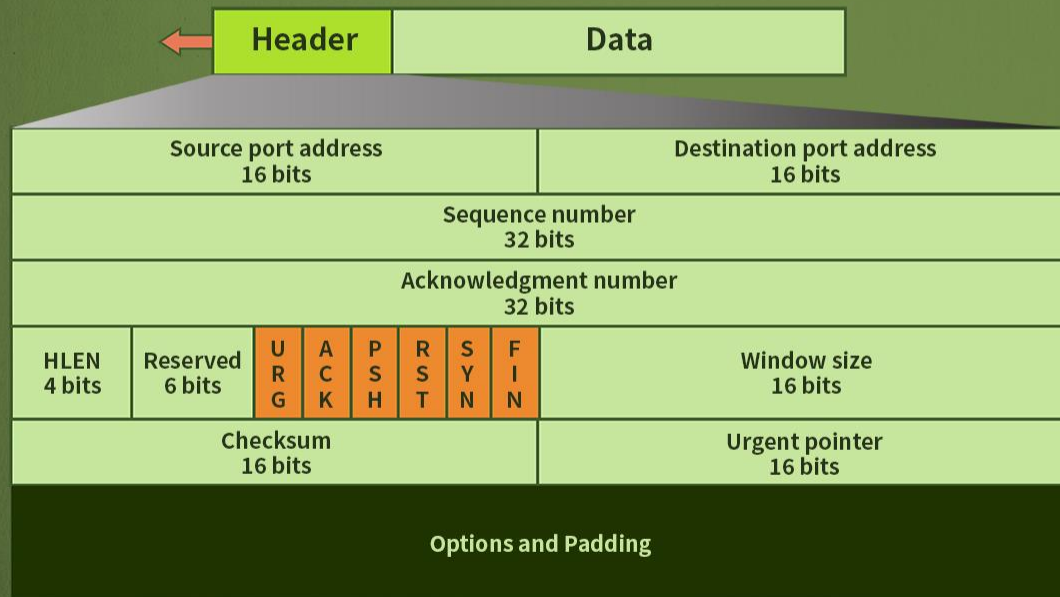
- TCP는 모든 바이트에 번호를 부여

- 시작번호는 0 ~ 232 - 1 사이의 임의의 번호를 갖음
- 번호는 흐름제어와 에러제어에 사용
- 세그먼트의 순서번호는 해당 세그먼트가 나르는 데이터의 첫 번째 바이트 번호

- ACK 번호는 수신해야 할 다음 바이트 번호

- ACK 번호는 누적 값으로 해당 번호 이전의 모든 바이트를 안전하게 받았음을 의미
- 예를 들어, 1234가 ACK번호라면, 1233바이트까지 모두 받았음을 의미

◆ 세그먼트 형태



◆ 세그먼트 형

» 헤더는 20바이트이며, 옵션이 있으면 60바이트까지 존재

- 4바이트 단위로서 표시

» 제어 필드

- 여러 비트가 동시에 설정될 수 있음
- 부정적 ACK(NACK)가 없음

URG : Urgent pointer is valid

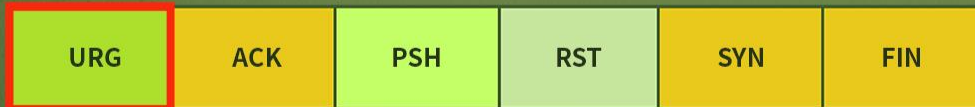
ACK : Acknowledgment is valid

PSH : Request for push

RST : Reset the connection

SYN : Synchronize sequence numbers

FIN : Terminate the connection



◆ 세그먼트 형태

» 긴급 데이터

- 송신자는 수신자가 순서에 관계없이 **우선적으로 데이터가 처리**되길 원할 수 있음
- 긴급데이터를 포함하면 **제어 필드의 PSH와 긴급 포인터(Urgent pointer)가 사용**

» 송신 TCP는 수신 TCP가 버퍼링을 하지 말고 즉시 데이터를 응용에게 전달할 것을 요구

- 제어 필드의 PSH 비트를 설정
- **지연된 전송은 대화식 응용에서는(Interactive applications) 바람직하지 않음**



◆ 세그먼트 형태

» RST(Reset)은 비정상적인 상황이나 오랫동안 통신이 없으면 현재의 연결을 끊음

» 체크섬

- 데이터에 대한 오류를 검사하여 재전송에 의한 복구를 수행
- TCP에서 체크섬은 강제사항
- UDP에서와 같은 가상헤더를 포함하여 계산



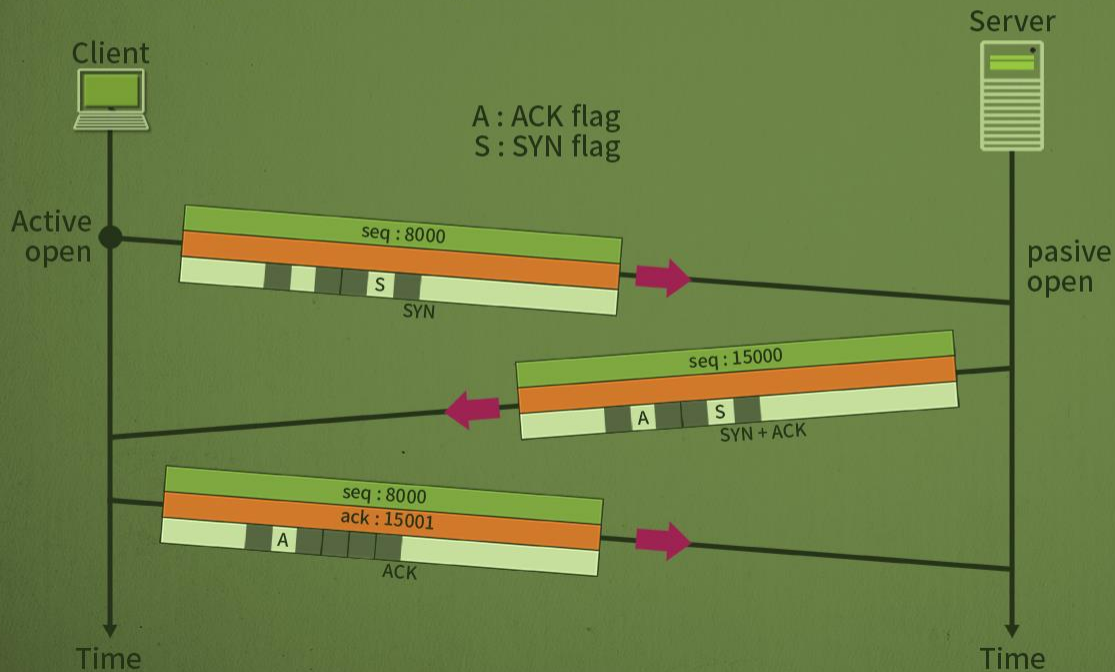
◆ 연결 설정과 해제

» TCP는 3단계 메시지 교환(**Three-way handshake**)을 통해서 연결설정

- 클라이언트는 연결을 요청하는 **SYN** 세그먼트를 전송
- 서버는 **SYN**과 **ACK**를 포함하는 세그먼트로 응답
- 클라이언트는 **ACK**를 보냄

◆ 연결 설정과 해제

» three-way handshaking을 사용한 연결 설정





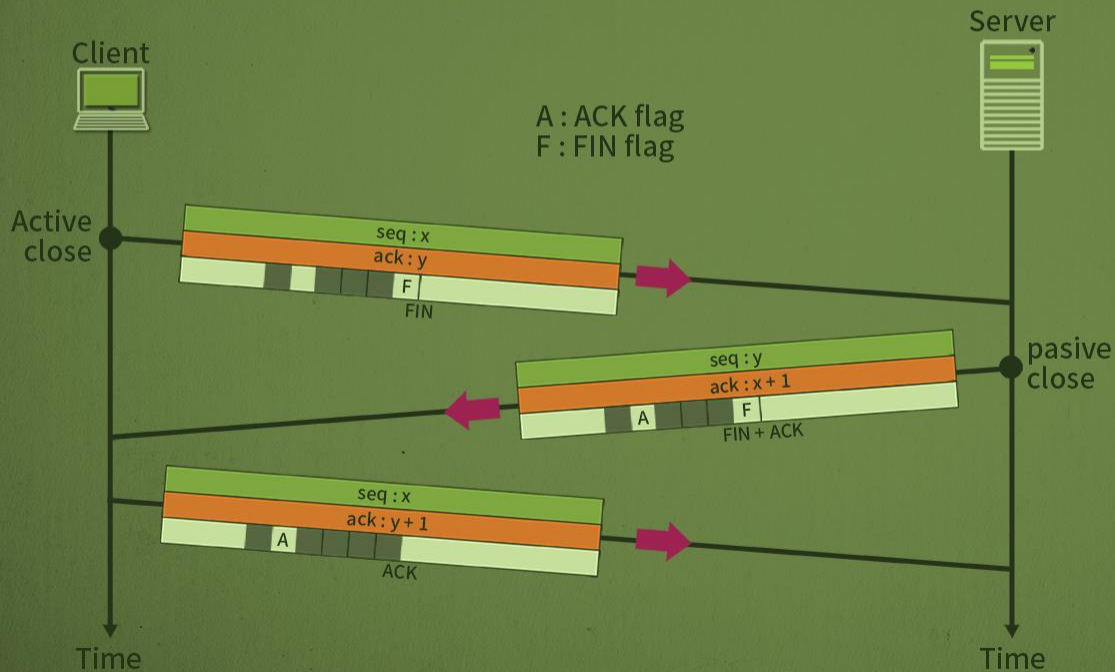
◆ 연결 설정과 해제

» TCP는 연결해제를 위해서 **3단계(three-way handshaking)**와 **4단계(four-way handshaking)** 메시지 교환을 제공

- 제어필드의 FIN을 사용하여 연결 해제
- **4단계 메시지교환을 통한 연결해제는 half-close 상태**를 만들어 수신은 가능하게 함
- **3단계 메시지교환을 통한 연결해제는 즉시 연결을 종료**

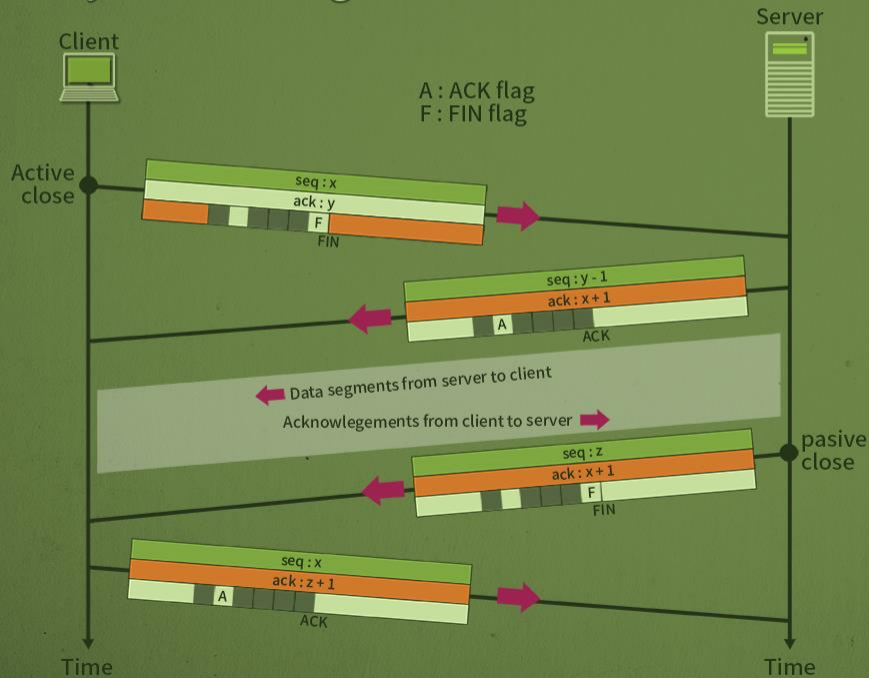
◆ 연결 설정과 해제

» three-way handshaking을 사용한 연결 종료



◆ 연결 설정과 해제

» four-way handshaking을 사용한 연결 종료



흐름제어

» 송신 TCP가 목적지로 부터 **ACK**를 수신하기 전에 보낼 수 있는 **데이터의 양**을 정함

- TCP는 슬라이딩 윈도우 프로토콜을 (sliding window protocol) 이용
- TCP는 바이트 단위로 **윈도우 크기(수신 가능한 데이터 양)**을 명시
- 윈도우 크기는 **시간에 따라서 변할 수 있음**

