TCP / UDP

TCP & UDP

- Transport (전송) 계층에서 사용하는 프로토콜.
- 전송계층에서 사용하는 주소 = Port
- port = service = process
- PID : 로컬에서 동작하는 프로세스인 서비스의 ID
- Port : 리모트에서 사용하는 프로세스 ID

L2	L3	L4	L7
Ethernet	IP	TCP	HTTP, SSH, Telnet, FTP
Ethernet	IP	UDP	DHCP, DNS, SNMP

TCP vs UDP

	Reliable	Best-Effort
Protocol	**TCP**	**UDP**
Connection Type	Connection-oriented	Connectionless
Sequencing	Yes	No
Uses	E-mail File sharing Downloading	Voice streaming Video streaming Real-time services

https://www.learncisco.net/courses/icnd-1/building-a-network/tcpip-transport-layer.html

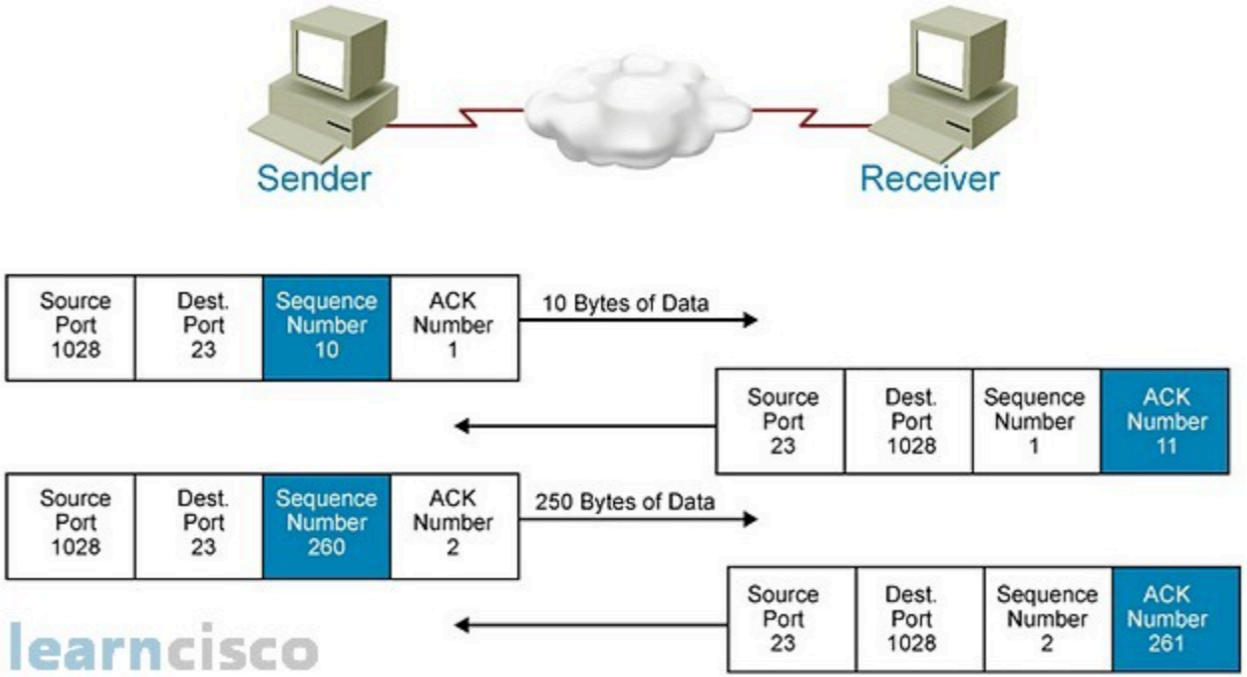
TCP가 신뢰성을 보장하는 방법

1. Stateful - TCP Flag

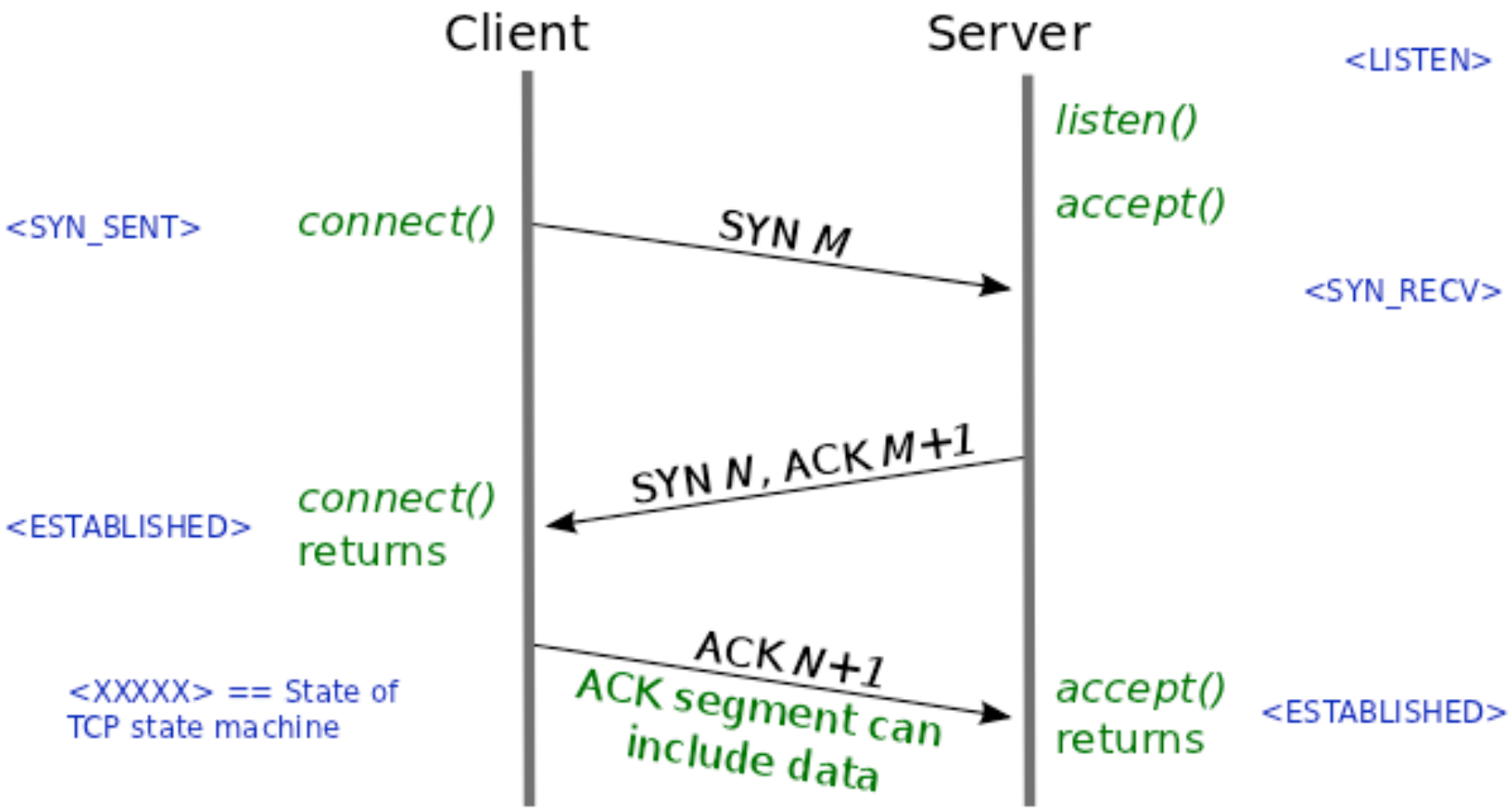
   Urgent
   Acknowledgement
   Push
   Reset
   Synchronize
   Finish

2. Session : Sequence num + Acknowledgement num

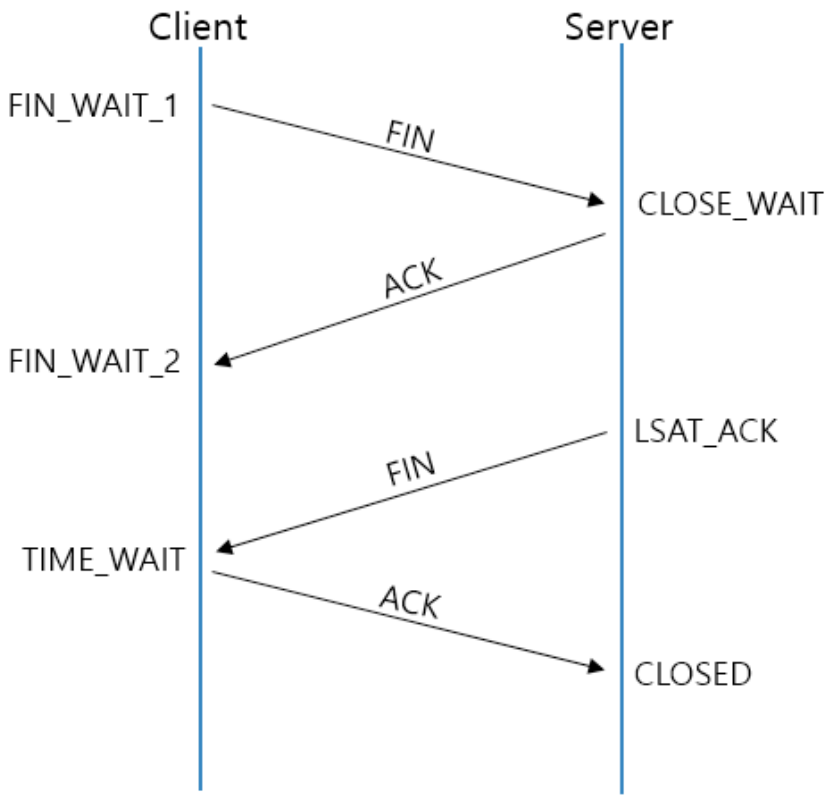
흐름 제어


Sender
Receiver
Source Port 1028
Dest. Port 23
Sequence Number 10
ACK Number 1
10 Bytes of Data
Source Port 23
Dest. Port 1028
Sequence Number 1
ACK Number 11
Source Port 1028
Dest. Port 23
Sequence Number 260
ACK Number 2
250 Bytes of Data
Source Port 23
Dest. Port 1028
Sequence Number 2
ACK Number 261
learncisco

3-Way Handshake



4-Way Handshake



WireShark 실습 - 웹사이트 연결

1. 패킷 캡처를 시작하기 전에 먼저 정보 확인, 쿠키 삭제

```
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=400<CHANNEL_IO>
        ether f0:18:98:77:0c:50
        inet6 fe80::c59:b316:f91:1e9b%en0 prefixlen 64 secured scopeid 0x6
        inet 172.30.1.8 netmask 0xffffff00 broadcast 172.30.1.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

WireShark 실습 - 웹사이트 연결

2. 사이트 접속 후 캡처 종료, 분석 시작

udp.stream eq 15

No.	Time	Source	Destination	Protocol	Length	Info
1191	6.793183	172.30.1.8	168.126.63.1	DNS	73	Standard query 0xefca A www.swu.ac.kr
1193	6.804104	168.126.63.1	172.30.1.8	DNS	126	Standard query response 0xefca A www.swu.ac.kr

```
▶ Frame 1193: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits) on interface 0
▶ Ethernet II, Src: Davolink_0e:21:a5 (60:29:d5:0e:21:a5), Dst: Apple_77:0c:50 (f0:18:98:77:0c:50)
▶ Internet Protocol Version 4, Src: 168.126.63.1, Dst: 172.30.1.8
▼ User Datagram Protocol, Src Port: 53, Dst Port: 12879
    Source Port: 53
    Destination Port: 12879
    Length: 92
    Checksum: 0x30fc [unverified]
    [Checksum Status: Unverified]
    [Stream index: 15]
  ▶ [Timestamps]
▼ Domain Name System (response)
    Transaction ID: 0xefca
  ▶ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 1
    Additional RRs: 1
  ▼ Queries
    ▶ www.swu.ac.kr: type A, class IN
  ▼ Answers
    ▶ www.swu.ac.kr: type A, class IN, addr 203.246.40.6
```

tcp.port == 80 && ip.addr == 203.246.40.6

No.	Time	Source	Destination	Protocol	Length	Info
1195	6.804558	172.30.1.8	203.246.40.6	TCP	78	59637 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1396575462 TSecr=0 SACK_PERM=1
1198	6.811498	203.246.40.6	172.30.1.8	TCP	74	80 → 59636 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=4121273503 TSecr
1199	6.811555	172.30.1.8	203.246.40.6	TCP	66	59636 → 80 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=1396575469 TSecr=4121273503
1200	6.811720	172.30.1.8	203.246.40.6	HTTP	514	GET / HTTP/1.1
1201	6.812568	203.246.40.6	172.30.1.8	TCP	74	80 → 59637 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=4121273505 TSecr
1202	6.812625	172.30.1.8	203.246.40.6	TCP	66	59637 → 80 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=1396575470 TSecr=4121273505
1203	6.819305	203.246.40.6	172.30.1.8	TCP	66	80 → 59636 [ACK] Seq=1 Ack=449 Win=15616 Len=0 TSval=4121273509 TSecr=1396575469
1204	6.820092	203.246.40.6	172.30.1.8	HTTP	368	HTTP/1.1 200 OK (text/html)
1205	6.820161	172.30.1.8	203.246.40.6	TCP	66	59636 → 80 [ACK] Seq=449 Ack=303 Win=131456 Len=0 TSval=1396575476 TSecr=4121273511
1220	7.085803	172.30.1.8	203.246.40.6	HTTP	554	GET /index.do HTTP/1.1
1222	7.129213	203.246.40.6	172.30.1.8	TCP	66	80 → 59636 [ACK] Seq=303 Ack=937 Win=16640 Len=0 TSval=4121273822 TSecr=1396575740
1252	8.134708	203.246.40.6	172.30.1.8	TCP	1514	80 → 59636 [ACK] Seq=303 Ack=937 Win=16640 Len=1448 TSval=4121274824 TSecr=1396575740 [TCP se
1253	8.134714	203.246.40.6	172.30.1.8	TCP	1514	80 → 59636 [ACK] Seq=1751 Ack=937 Win=16640 Len=1448 TSval=4121274824 TSecr=1396575740 [TCP s
1254	8.134797	172.30.1.8	203.246.40.6	TCP	66	59636 → 80 [ACK] Seq=937 Ack=3199 Win=128512 Len=0 TSval=1396576782 TSecr=4121274824
1255	8.135630	203.246.40.6	172.30.1.8	TCP	1514	80 → 59636 [ACK] Seq=3199 Ack=937 Win=16640 Len=1448 TSval=4121274824 TSecr=1396575740 [TCP s
1256	8.135637	203.246.40.6	172.30.1.8	TCP	1514	80 → 59636 [ACK] Seq=4647 Ack=937 Win=16640 Len=1448 TSval=4121274824 TSecr=1396575740 [TCP s
1257	8.135639	203.246.40.6	172.30.1.8	TCP	1514	80 → 59636 [ACK] Seq=6095 Ack=937 Win=16640 Len=1448 TSval=4121274824 TSecr=1396575740 [TCP s
1258	8.135641	203.246.40.6	172.30.1.8	TCP	1514	80 → 59636 [ACK] Seq=7543 Ack=937 Win=16640 Len=1448 TSval=4121274824 TSecr=1396575740 [TCP s
1259	8.135643	203.246.40.6	172.30.1.8	TCP	378	80 → 59636 [PSH, ACK] Seq=8991 Ack=937 Win=16640 Len=312 TSval=4121274824 TSecr=1396575740 [T
1260	8.135646	203.246.40.6	172.30.1.8	TCP	1514	80 → 59636 [ACK] Seq=9303 Ack=937 Win=16640 Len=1448 TSval=4121274825 TSecr=1396575740 [TCP s
1261	8.135648	203.246.40.6	172.30.1.8	HTTP	864	HTTP/1.1 200 OK (text/html)
1262	8.135761	172.30.1.8	203.246.40.6	TCP	66	59636 → 80 [ACK] Seq=937 Ack=6095 Win=128128 Len=0 TSval=1396576782 TSecr=4121274824
1263	8.135762	172.30.1.8	203.246.40.6	TCP	66	59636 → 80 [ACK] Seq=937 Ack=8991 Win=125248 Len=0 TSval=1396576782 TSecr=4121274824
1264	8.135762	172.30.1.8	203.246.40.6	TCP	66	59636 → 80 [ACK] Seq=937 Ack=9303 Win=124928 Len=0 TSval=1396576782 TSecr=4121274824
1265	8.135801	172.30.1.8	203.246.40.6	TCP	66	59636 → 80 [ACK] Seq=937 Ack=11549 Win=122688 Len=0 TSval=1396576783 TSecr=4121274825

```
▶ Frame 1194: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0
▶ Ethernet II, Src: Apple_77:0c:50 (f0:18:98:77:0c:50), Dst: Davolink_0e:21:a5 (60:29:d5:0e:21:a5)
▶ Internet Protocol Version 4, Src: 172.30.1.8, Dst: 203.246.40.6
▼ Transmission Control Protocol, Src Port: 59636, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 59636
    Destination Port: 80
    [Stream index: 14]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 0
    1011 .... = Header Length: 44 bytes (11)
  ▶ Flags: 0x002 (SYN)
    Window size value: 65535
    [Calculated window size: 65535]
    Checksum: 0x65c6 [unverified]
```

```
0000  60 29 d5 0e 21 a5 f0 18  98 77 0c 50 08 00 45 00   `)··!··· ·w·P··E·
0010  00 40 00 00 40 00 40 06  99 95 ac 1e 01 08 cb f6   ·@··@·@· ·····
```

Internet Protocol Version 4 (ip), 20 bytes Packets: 23753 · Displayed: 21252 (89.5%) Profile: Default

왜 HTTP는 TCP 세션을 사용하지 않을까?

```
<- 논리적 단위 -> <--- 물리적 단위          ----------------------> <-- 메시지의 교환 -->
세션(세션ID)시작 ->
                    클라이언트 소켓 생성 ->
                                          서버 소켓 연결(Connection)
                                                                  -> 요청(Request)
                                                                  <- 응답(Response)
                                                                    ....
                                                                  -> 요청(Request)
                                                                  <- 응답(Response)
                                       <-   서버 소켓과 연결 닫힘(Close)
                   <- 클라이언트 소켓 닫힘
세션종료
```

왜 HTTP는 TCP 세션을 사용하지 않을까?

- 사용자가 지속적으로 요청할 때마다 Thread가 생성되어 소켓 연결/종료 반복
  → 비효율적
- 연속된 요청과 응답을 위해 세션 ID를 사용하는 웹 세션 사용
- 서버는 클라이언트로 세션 ID를 HTTP Cookie를 통해 전송
- 클라이언트 PC에 세션 ID가 저장되고, 이 ID를 요청과 함께 서버에 보냄

Q&A