**간단한 TCP/IP에 대한 검색 – 네이버 백과사전 TCP/IP**

**https://terms.naver.com/entry.nhn?docId=2274898&cid=42171&categoryId=51118**

TCP? 전송할 파일을 작은 조각(패킷)으로 나누어 전송, 받은 조각들을 원래의 메시지로 조립

IP? 조각들의 주소 부분을 처리해 패킷들이 목적지에 정확하게 도달하도록 함

**정석용, 『정석용의 TCP / IP 소켓 프로그래밍』, 프리렉(2009), 106-108**

**<TCP/IP 4계층>**

TCP/IP 4계층? 미국 국방성이 개발한 ARPAnet의 통신 프로토콜로 인터넷의 기본 프로토콜로 사용되면서 실질적인 표준으로 자리 매김 함

프로토콜? 통신 규약

4. 응용 계층: OSI 7계층의 세션 계층, 표현 계층, 응용 계층.

텔넷, FTP, SMTP와 같은 TCP와 UDP 기반의 응용 프로그램 구현할 때 사용

3. 전송 계층: OSI 7계층의 전송 계층. 통신 노드 간의 연결을 제어, 자료의 송수신을 담당.

프로토콜로는 스트림 형태의 연결형 서비스인 TCP, 데이터그램 형태의 비연결형 서비스인 UDP가 있음

2. 인터넷 계층: OSI 7계층의 네트워크 계층. 통신 노드 간의 IP 패킷을 전송하는 기능과 라우팅 기능.

프로토콜로는 IP, ICMP, ARP, RARP가 있음. IP는 데이터그램 방식의 비연결형 서비스만 제공

1. 네트워크 액세스 계층: OSI 7계층의 물리 계층, 데이터 링크 계층. LAN, X.25, 패킷망, 위성 통신, 다이얼업 모뎀 등에 사용. 특히 이더넷에서는 CSMA/CD MAC 프로토콜을 사용하며 IEEE 802.3 MAC 표준으로 규정되어 있음.

이러한 통신 프로토콜 계층 구조를 기반으로 하여 프로그램으로 각 계층별 자료를 제어하면서 통신할 수 있는 것. 소켓 프로그램은 전송 계층에서 제공하는 통신 함수를 호출해서 자료를 송수신하게 됨

**정석용, 『정석용의 TCP / IP 소켓 프로그래밍』, 프리렉(2009), 108-109**

**<소켓 프로그램>**

대표적인 전송 계층(3계층)의 프로그램, TCP, UDP 기반의 전송 계층 프로그램. 소켓 API함수를 이용해 통신 노드 간의 패킷 단위의 자료 전송을 제어함, 운영체제에 따라 유닉스 BSD 소켓, 리눅스 소켓, 윈도우 소켓 등이 있음

소켓 모듈은 자료 전송을 위해 전송 계층에 있는 TCP 모듈 혹은 UDP 모듈의 함수를 호출함.

TCP 모듈, UDP 모듈은 발신지나 최종 목적지의 컴퓨터에서만 동작함, 라우터 같은 중간에서 자료를 전달하는 컴퓨터에서는 동작하지 않음

TCP, UDP 모듈은 다음 경유지로 자료를 전송하기 위해 IP모듈의 함수를 호출함. IP 모듈에서는 NIC를 통해 다음 경유지로 자료를 전송하기 위해 디바이스 드라이버의 함수를 호출함.

디바이스 드라이버에서는 컴퓨터에 연결된 NIC를 제어하고 자료를 실제 케이블을 통해 전송함.

NIC? Network Interface Card

**정석용, 『정석용의 TCP / IP 소켓 프로그래밍』, 프리렉(2009), 109-111**

**<TCP>**

응용 프로그램 입장에서 TCP는 배송물을 목적지까지 전달하는 이삿짐 센터의 역할.

배송물에 주소를 적어 의뢰하기만 하면 TCP가 처리해줌

응용 프로그램은 이삿짐 센터가 배송물을 목적지까지 전달하기 위해 어떤 경로를 채택하는지 알 필요가 없음, TCP는 전송 계층의 프로토콜로 통신하는 양쪽 노트 간 자료의 전송과 제어를 책임지기 때문

TCP는 배송물을 목적지까지 손실 없이 전달해줌, 즉 신뢰성이 담보되어 있음. 이를 위해 TCP는 자료가 목적지에 잘 전달되었는지 확인하고, 그렇지 못하면 미리 보관하고 있던 사본을 다시 전송하는 방식을 취함.

→ TCP는 재전송을 통한 오류 제어 등으로 신뢰성 있는 자료 전송을 보장함

TCP는 목적지로 자료를 전송할 때 수신 측에서 수용할 수 있는 만큼의 자료만을 전송함, 즉 수용 가능한 자료만 전송하는 흐름제어(Flow Control)를 통해 자료의 안전성을 보장함. 이를 위해 양쪽 시스템은 수신 측 용량에 대한 별도의 정보를 서로 교환함.

TCP를 통해 자료를 전송하는 경우, TCP는 우선 자료가 목적지에 도달할 때까지 경유할 경로를 결정함. 자료는 전송 단위인 패킷으로 쪼개서 번호를 매김, 미리 결정한 경로를 통해 순서대로 전송함. 이렇게 하면 모든 자료가 동일한 경로를 거쳐 전송되어 자료의 순서가 뒤바뀌지 않음

→연결형 서비스

정리: 통신하는 컴퓨터 간에 미리 경로를 결정해서 자료를 송수신하고, 이것이 끝나면 연결을 종료하는 통신 방식. 한번의 많은 양의 자료를 전송할 수도 있고, 동일한 경로를 통해 목적지에 전달되기 때문에 자료의 순서가 보장됨

TCP와 UDP 장단점: TCP가 UDP보다 안전하게 자료를 전송함, 하지만 UDP보다 많은 비용을 치러야 함. 자료 전송 이전에 경로를 결정해야 하고, 자료 전송 여부를 확인해야 하며, 손실이 생기면 사본을 다시 전송해야 하는 등의 추가적 비용을 지불해야 하기 때문

전송 오류가 적거나, 패킷 순서가 바뀌지 않는 환경이거나, 한 번의 패킷 송수신으로 서비스가

완성되는 서비스, 예를 들면 LAN, DNS, time서비스, NFS 등에서는 빠른 UDP를 사용하는 것이 유리

**정석용, 『정석용의 TCP / IP 소켓 프로그래밍』, 프리렉(2009), 111-125**

**<TCP로 자료 송수신>**

소켓은 TCP/UDP를 이용하는 창구 역할, 응용 프로그램은 소켓을 통해 자료를 송수신 할 수 있음.

응용 프로그램에서 TCP/UDP를 이용해 원격지 컴퓨터로 자료를 전송하려면 우선 TCP/IP의 창구 역할을 하는 소켓을 생성하고, 운영체제에서 제공하는 소켓 API를 사용해 자료를 송수신하면 됨.

Ex) 전화망을 이용해 음성 통신을 할 때 전화기가 필요한 것처럼, TCP/IP 통신망을 이용해 소켓 통신을 하려면 소켓이 필요함. 즉, 소켓 = 전화기

소켓은 연결형 서비스를 제공하는 TCP용 소켓, 비연결형 서비스를 제공하는 UDP용 소켓 두 유형이 있음, 소켓을 통해 자료를 송수신할 때 소켓 기술자를 이용함. 파일 기술자를 이용하면 파일을 읽거나 쓸 때뿐만 아니라 소켓을 이용해 자료를 송수신할 때도 사용

서로 통신할 때면 두 개의 소켓을 생성해 자료를 송수신함. 두 소켓이 연결되면 소켓을 통해 자료를 송수신할 수 있게 됨

**정석용, 『정석용의 TCP / IP 소켓 프로그래밍』, 프리렉(2009), 153-154**

**<주소의 구성>**

TCP/IP 통신망에서는 소켓 통신을 위해 상대 컴퓨터와의 접속점으로 IP 주소와 포트 번호를 사용함

IP 주소: 문자열(도메인)이거나, 202.131.30.81 과 같은 10진수, 네트워크 바이트 순서로 된 이진 값으로 변환해야 함

포트 번호: 2바이트로 구성, TCP/IP 프로토콜에서는 빅 엔디안 방식을 채택. 그러므로 2바이트의 포트 번호를 빅 엔디안 방식으로 변환해야 함

빅 엔디안 방식? 시작 주소에 데이터의 최상위 비트가 오도록 저장

소켓 통신을 하려면 자료를 교환할 상대방의 주소(IP주소, 포트번호)를 알고 연결해야 함

**정석용, 『정석용의 TCP / IP 소켓 프로그래밍』, 프리렉(2009), 173-176**

**<TCP 소켓 프로그래밍 - 서버>**

1. 연결 준비 단계

네트워크 통신을 위해 소켓 생성→연결할 주소(서버의 주소) 준비→소켓을 프로토콜 주소에 연결→소켓으로 들어오는 연결 요청 수신

2. 서비스 처리 단계

클라이언트에서 네트워크 통신을 위해 소켓 생성→클라이언트에서 서버로 연결 요청→서버가 클라이언트의 연결 요청을 기다리다가 접수→서버가 연결 소켓으로 클라이언트와 자료를 송수신하며 서비스 제공→클라이언트와의 연결 소켓을 종료

3. 서버 종료 단계

듣기 소켓을 종료하면 더 이상 클라이언트로부터의 연결 요청을 받아들일 수 없음, 즉 서버 프로그램이 종료됨

**정석용, 『정석용의 TCP / IP 소켓 프로그래밍』, 프리렉(2009), 201-202**

**<TCP 소켓 프로그래밍 - 클라이언트>**

1. 연결 준비 단계

네트워크 통신을 위해 소켓 생성→연결할 서버의 프로토콜 주소 준비

2. 서비스 처리 단계

소켓을 서버에 연결→서버로부터 자료를 읽어들임(자료 송수신) →서버와의 연결 소켓 종료

**잠깐 요약!**

TCP는 연결형 서비스를 제공하며, 도착한 자료의 순서를 보장하기 위해 전송 자료에 순번을 매김.

TCP는 자료 수신측에 ACK 패킷을 요구해 자료의 전송 오류, 중복, 손실, 파손 등이 확인되면 재전송으로 자료를 복구함.

한 번에 전송하는 자료의 크기에 제한이 없음, 수신측의 자료 처리 능력과 네트워크 상황에 따라 크기를 조절하는 흐름 제어를 수행함

**정석용, 『정석용의 TCP / IP 소켓 프로그래밍』, 프리렉(2009), 394-396**

**<TCP 상태 변화 – 1. 연결 준비 단계>**

TCP는 연결형 서비스이므로 TCP로 통신하려는 두 개의 호스트는 별도의 제어 정보를 주고받음으로써 호스트 간의 연결을 이루어냄. 클라이언트의 응용 프로그램에서 서버와 연결하기 위해 connect 함수를 호출하면 특정 제어 비트를 1로 설정한 TCP 패킷을 서로 교환함으로써 이루어짐. 패킷 3개를 서로 교환하기 때문에 3방향 핸드 셰이킹이라고 부름

-1단계(SYN 패킷 전송): 우선 클라이언트는 제어 비트 중 SYN 비트를 1로 설정한 TCP 제어 패킷(SYN 패킷)을 서버로 전송해 연결을 시도함. TCP는 송수신되는 자료의 바이트 순서를 관리함, SYN 패킷에 기록된 일련번호와 확인번호 필드의 값으로 순서를 추적함. 처음 SYN 패킷에는 클라이언트가 사용할 일련번호의 시작값(ISN)을 일련 번호 필드에 기록해 전송함. 즉, SYN 패킷의 일련 번호 필드에는 연결 후 자료를 송수신할 때 사용할 ISN이 있음. 보안상의 이유는 ISN은 임의의 숫자로 시작해야 하지만 0도 자주 사용됨

-2단계(SYN\_ACK 패킷 전송): 서버는 제어 비트 중 SYN 비트와 ACK 비트를 1로 설정한 SYN\_ACK 패킷을 클라이언트로 전송함으로써 클라이언트에게 SYN 패킷을 잘 받았음을 알림. 역시 서버가 사용할 ISN을 SYN\_ACK 패킷의 일련 번호 필드에 기록해서 전송함

-3단계(ACK 패킷 전송): 클라이언트는 2단계의 SYN\_ACK 패킷을 서버로부터 받은 후 두 호스트 간에 연결이 되었다는 것을 확인하고, 제어비트의 ACK 비트를 1로 설정한 TCP 제어 패킷(ACK 패킷)을 서버로 보냄. 두 호스트 간에 송수신하는 자료의 각 바이트는 3방향 핸드셰이킹의 연결 과정에서 교환한 ISN에서부터 순서대로 번호를 매김. 따라서 TCP 제어 패킷의 일련 번호 필드의 값을 통해 해당 패킷의 자료가 전체 자료 중 몇 번째 바이트를 송수신했는지에 대해 알 수 있음

ACK 패킷은 현재까지 자료를 얼마나 전송받았고, 또 얼마나 받을 수 있는지 알려줌. 즉, ACK 패킷의 확인번호 필드의 값은 수신하려고 하는 다음 바이트의 일련 번호로 확인 번호 이전까지의 바이트는 모두 수신하였음을 확인해 줌. 또한 윈도우 필드의 값은 수신자가 전송받을 수 있는 바이트의 크기를 설정함. 송신자는 수신자가 설정한 윈도우 필드의 값 내에서 자료를 송신할 수 있음. 즉 흐름 제어 기능을 수행함

**정석용, 『정석용의 TCP / IP 소켓 프로그래밍』, 프리렉(2009), 404-405**

**<TCP 상태 변화 – 2. 자료 송수신 단계>**

자료의 송수신은 클라이언트와 서버 소켓 모두 ESTABLISHED 상태에서만 가능함. TCP에서는 세그먼트 하나로 전송할 수 있는 세그먼트의 최대 크기(MSS)를 설정하여 사용함.

실제 응용 프로그램에서 원격지로 자료를 전송하면 자료의 송수신은 커널의 송수신 버퍼를 통해 이루어짐

원격지로 자료를 전송할 때 응용 프로그램의 버퍼의 내용은 커널의 TCP 송신 버퍼로 복사됨. 응용 프로그램의 버퍼의 내용 모두가 커널의 TCP 송신 버퍼에 복사될 때까지 응용 프로그램은 차단됨. TCP 송신 버퍼의 크기는 옵션 SO\_SNDBUF로 지정. 응용 프로그램의 버퍼의 크기가 크거나 송신 버퍼에 비어 있지 않아서 모두 복사하지 못하면 응용 프로그램이 차단됨.

TCP는 송신 버퍼의 내용을 MSS만큼 IP를 통해 원격지로 전송함. MSS는 connect 함수 호출 시에 3방향 핸드셰이킹을 통해 상대방에게서 받음.

IP는 해당 내용을 데이터 링크의 출력 큐에 넣어 전송함. 큐가 가득 차면 TCP로 재전송을 요구함. TCP는 응용 프로그램에 알리지 않고 커널의 송신 버퍼의 내용을 재전송함. 상대방이 전송한 ACK 패킷을 받으면 송신 버퍼의 내용을 폐기함.

**정석용, 『정석용의 TCP / IP 소켓 프로그래밍』, 프리렉(2009), 406**

**<TCP 상태 변화 – 3. 연결 종료 단계>**

\* 클라이언트가 먼저 close 함수를 호출해서 연결 종료를 시작하는 것으로 가정함 \*

양쪽 호스트가 연결을 종료하기 위해서 FIN 패킷과 ACK 패킷의 제어 비트를 1로 설정하고 서로 교환함. 우선 클라이언트가 연결 종료를 시작한 것으로 가정하였기 때문에 클라이언트에서 서버로의 연결이 다음 1, 2단계를 거쳐서 종료됨. 양방향 연결에서 한쪽 방향의 연결만 종료되기 때문에 반쪽 종료가 이루어짐

-1단계(FIN 패킷 전송): 우선 클라이언트는 제어비트의 FIN 비트를 1로 설정한 TCP 제어 패킷(FIN 패킷)을 서버로 전송해서 연결 종료를 시작함

-2단계(ACK 패킷 전송): 서버는 FIN 패킷을 전송받고서 수신한 일련 번호(ISN)+1의 ACK 패킷을 클라이언트로 전송함

다음은 서버가 모든 작업을 종료하고 close 함수를 호출해 나머지 한 방향의 연결을 종료하는 단계임. 3, 4단계를 거치면 서버에서 클라이언트로의 연결이 반쪽 종료됨

-3단계(FIN 패킷 전송): 서버는 자신의 작업을 완료한 후에 연결 종료를 위해 close 함수를 호출함. 그러면 클라이언트로 제어 비트의 FIN 비트 1로 설정한 TCP 제어 패킷(FIN 패킷)을 전송함

-4단계(ACK 패킷 전송): 클라이언트는 서버에서 수신한 일련 번호(ISN)에 1을 더한 ACK 패킷을 서버로 전송함.

이처럼 TCP 연결 종료는 각 방향에 대해 독립적으로 이루어짐