

## 1. 프로토콜 스택의 내부 구성

여기에서는 OS에 내장된 네트워크 제어용 소프트웨어(프로토콜 스택)와 네트워크용 하드웨어(LAN어댑터)가 브라우저에서 받은 메시지를 서버에 송출하는 동작을 살펴본다.

### - 프로토콜 스택

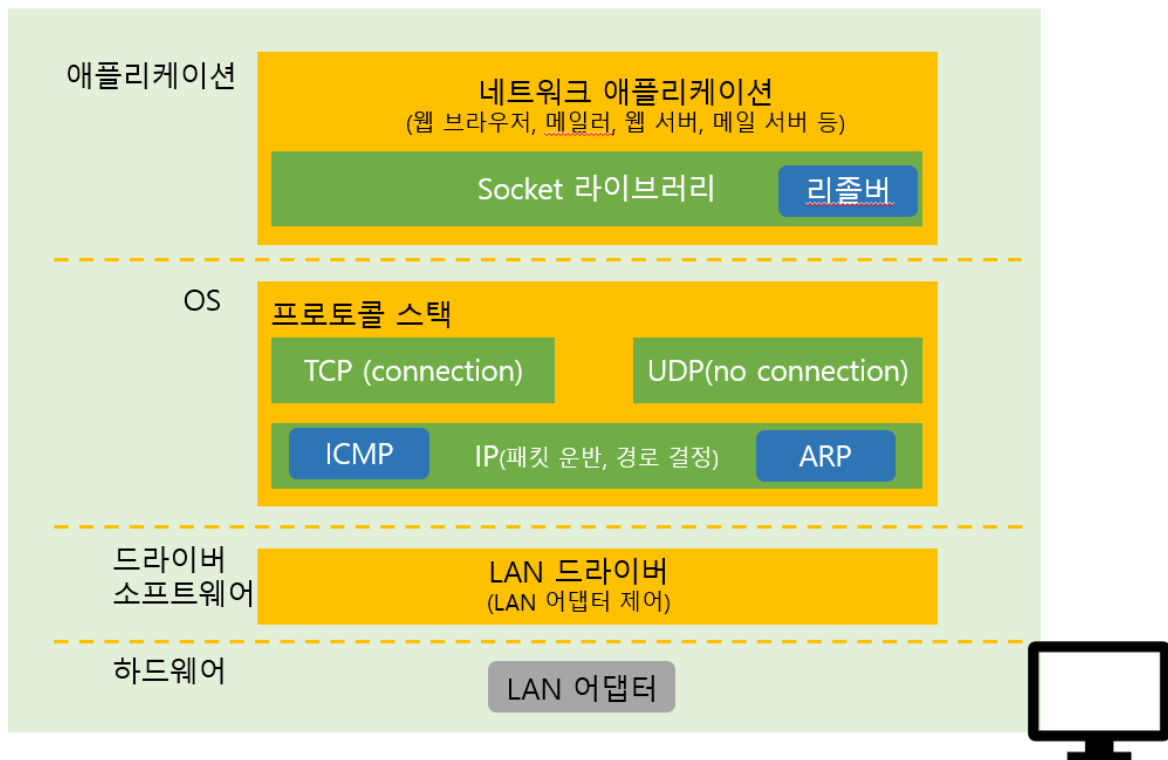


그림 1. TCP/IP 소프트웨어 계층 구조

TCP : Transmission Control Protocol

UDP : User Datagram Protocol

IP : Internet Protocol

ICMP : Internet Control Message Protocol

ARP : Address Resolution Protocol

## - TCP vs. UDP 사용 목적

TCP : 브라우저나 메일 등의 일반적인 애플리케이션이 데이터를 송수신 할 경우

UDP : DNS 서버에 대한 조회 등에서 짧은 제어용 데이터를 송수신 할 경우

- ➔ 인터넷에서 데이터를 운반할 때는 데이터를 작게 나누어 **패킷**이라는 형태로 운반한다.
- ➔ 패킷을 통신 상대까지 운반하는 것이 IP주소의 역할이다.
- ➔ IP안의 ICMP는 패킷을 운반할 때 발생하는 오류를 통지하거나 제어용 메시지를 통지할 때 사용한다.
- ➔ IP안의 ARP는 IP주소에 대응하는 이더넷의 MAC주소를 조사할 때 사용한다.
- ➔ LAN드라이버는 LAN 어댑터의 하드웨어를 제어한다.
- ➔ LAN어댑터는 실제 송수신 동작을 실행한다.

## 2. 소켓의 실체는 통신 제어용 제어 정보

프로토콜 스택은 내부에 제어 정보를 기록하는 메모리 영역을 가지고 있으며 여기에 통신 동작을 제어하기 위한 제어정보를 기록한다.

이 정보가 곧 소켓의 실체라고도 볼 수 있다.

프로토콜 스택은 이 제어 정보를 참조하면서 동작한다.

cmd -> netstat -ano 명령어로 현재 소켓 내용을 확인 할 수 있다.

## 3. Socket을 호출했을 때의 동작

소켓을 만들 때 한 개의 메모리 영역을 미리 확보하고 초기 상태라는 것을 이 영역에 기록한다.

소켓이 만들어 지면 소켓을 나타내는 디스크립터를 애플리케이션에 알려준다.

디스크립터를 받은 애플리케이션은 이후 프로토콜 스택에 데이터를 송신하는 동작을 의뢰할 때

디스크립터를 통지한다.

따라서 디스크립터가 어느 소켓인지를 나타내면 필요한 정보는 전부 프로토콜 스택에서 알 수 있다.

이렇게 함으로써 애플리케이션에서 일일이 통신상대의 정보를 통지받을 필요가 없어진다.

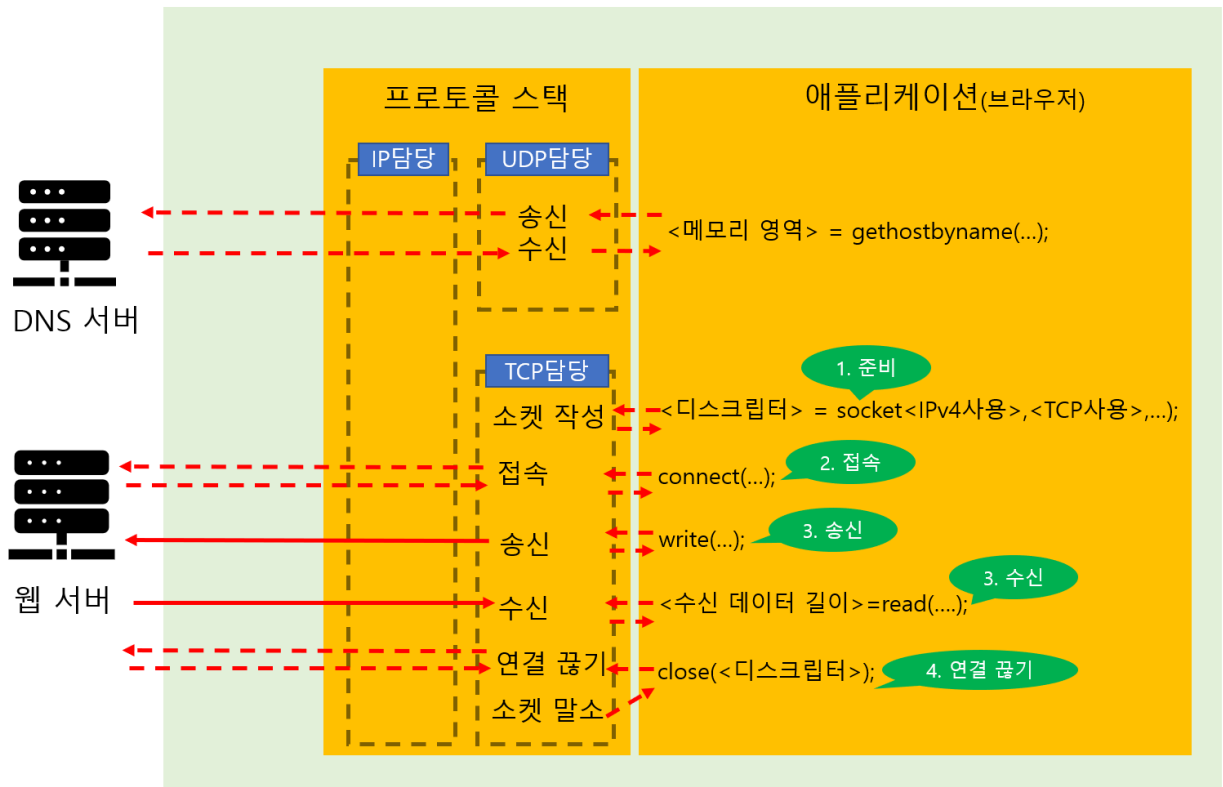


그림2. 메시지 송수신 동작