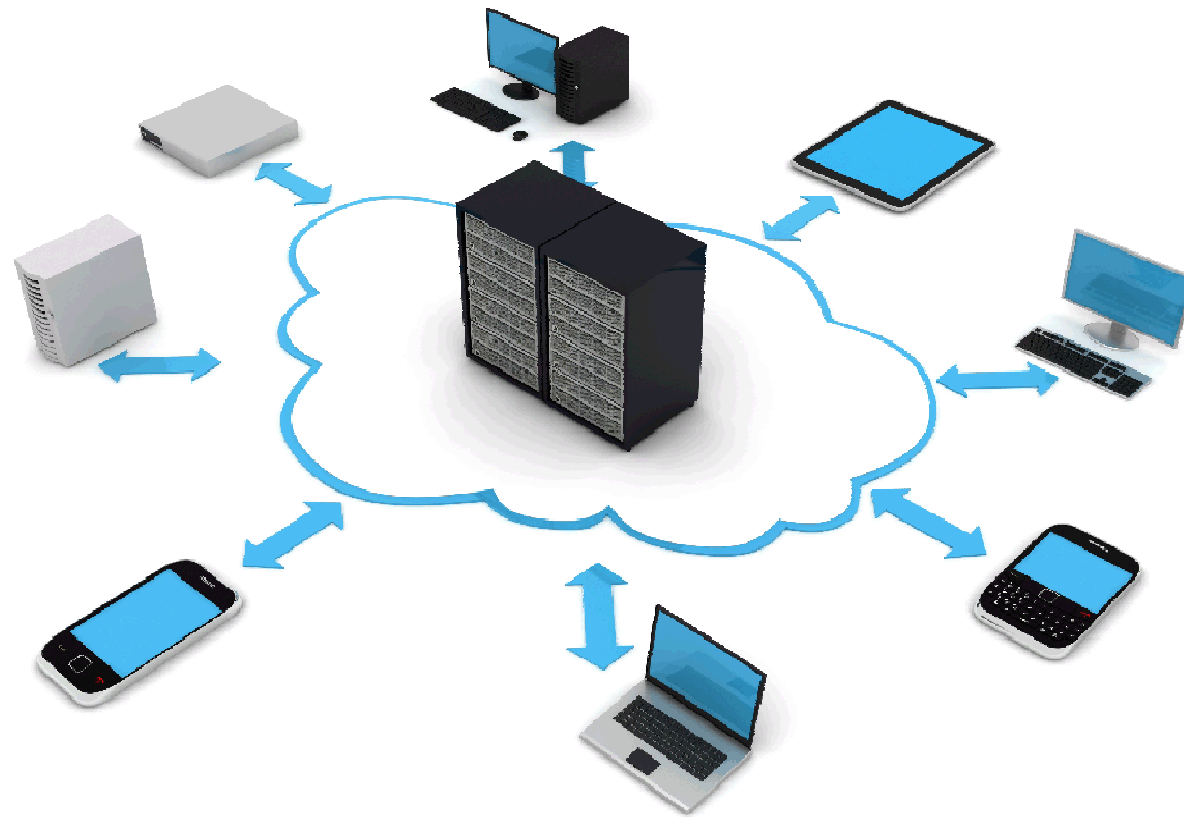


네트워크 (Network)

▶ 네트워크(Network)

여러 대의 컴퓨터를 통신 회선으로 연결한 것
(홈 네트워크, 지역 네트워크, 인터넷 등이 해당)



▶ 네트워크(Network)

✓ 서버와 클라이언트

네트워크로 연결된 컴퓨터간의 관계를 역할(role)로 구분한 개념

- 서버 : 서비스를 제공하는 프로그램으로 클라이언트의 연결을 수락하고 요청 내용을 처리 후 응답을 보내는 역할
- 클라이언트 : 서비스를 받는 프로그램으로 네트워크 데이터를 필요로 하는 모든 어플리케이션이 해당 됨

✓ IP주소

네트워크 상에서 컴퓨터를 식별하는 번호로

네트워크 어댑터(랜카드) 마다 할당 되어 있음

✓ 포트(Port)

같은 컴퓨터 내에서 프로그램을 식별하는 번호로

클라이언트는 서버 연결 요청 시 IP주소와 포트 번호를 알아야 함

▶ InetAddress 클래스

IP주소를 다루기 위해 자바에서 제공하는 클래스

메소드	설명
byte[] getAddress()	IP주소를 byte배열로 리턴
static InetAddress[] getAllByName(String host)	도메인 명에 지정된 모든 호스트의 IP주소를 배열에 담아 반환
static InetAddress getByAddress(byte[] addr)	byte배열을 통해 IP주소를 얻는다.
static InetAddress getByName(String host)	도메인 명을 통해 IP주소를 얻는다.
String getCanonicalHostName()	FQDN(Full Qualified Domain Name)을 얻는다.
String.getHostAddress()	호스트의 IP주소를 반환한다.
String.getHostName()	호스트의 이름을 반환한다.
static InetAddress getLocalHost()	지역호스트의 IP주소를 반환한다.
boolean isMulticastAddress()	IP주소가 멀티캐스트 주소인지 알려준다.
boolean isLoopbackAddress()	IP주소가 loopback 주소(127.0.0.1)인지 알려준다.

▶ 소켓 프로그래밍

소켓을 이용한 통신 프로그래밍

✓ 소켓

프로세스 간의 통신에 사용되는 양쪽 끝 단

✓ TCP

데이터 전송 속도가 느리지만 정확하고 안정적으로 전달할 수 있는
연결 지향적 프로토콜

✓ UDP

데이터 전송 속도가 빠르지만 신뢰성 없는 데이터를 전송하는
비연결 지향적 프로토콜

▶ TCP 소켓 프로그래밍

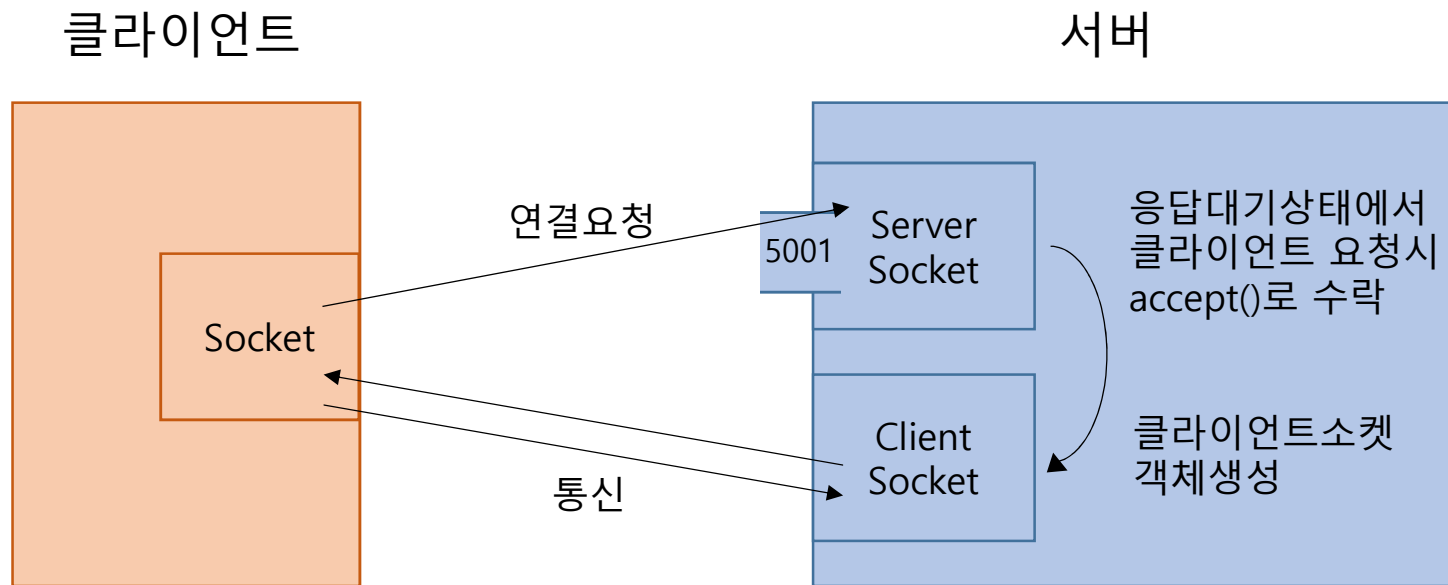
클라이언트와 서버간의 1:1 소켓 통신

서버가 먼저 실행 되어 클라이언트의 요청을 기다려야 하고

서버용 프로그램과 클라이언트용 프로그램을 따로 구현해야 함

자바에서는 TCP 소켓 프로그래밍을 위해 java.net패키지에서

ServerSocket과 Socket클래스 제공



▶ TCP 소켓 프로그래밍

✓ 서버용 TCP 소켓 프로그래밍 순서

1. 서버의 포트번호 정함
2. 서버용 소켓 객체 생성
3. 클라이언트 쪽에서 접속 요청이 오길 기다림
4. 접속 요청이 오면 요청 수락 후 해당 클라이언트에 대한 소켓 객체 생성
5. 연결된 클라이언트와 입출력 스트림 생성
6. 보조 스트림을 통해 성능 개선
7. 스트림을 통해 읽고 쓰기
8. 통신 종료

▶ TCP 소켓 프로그래밍

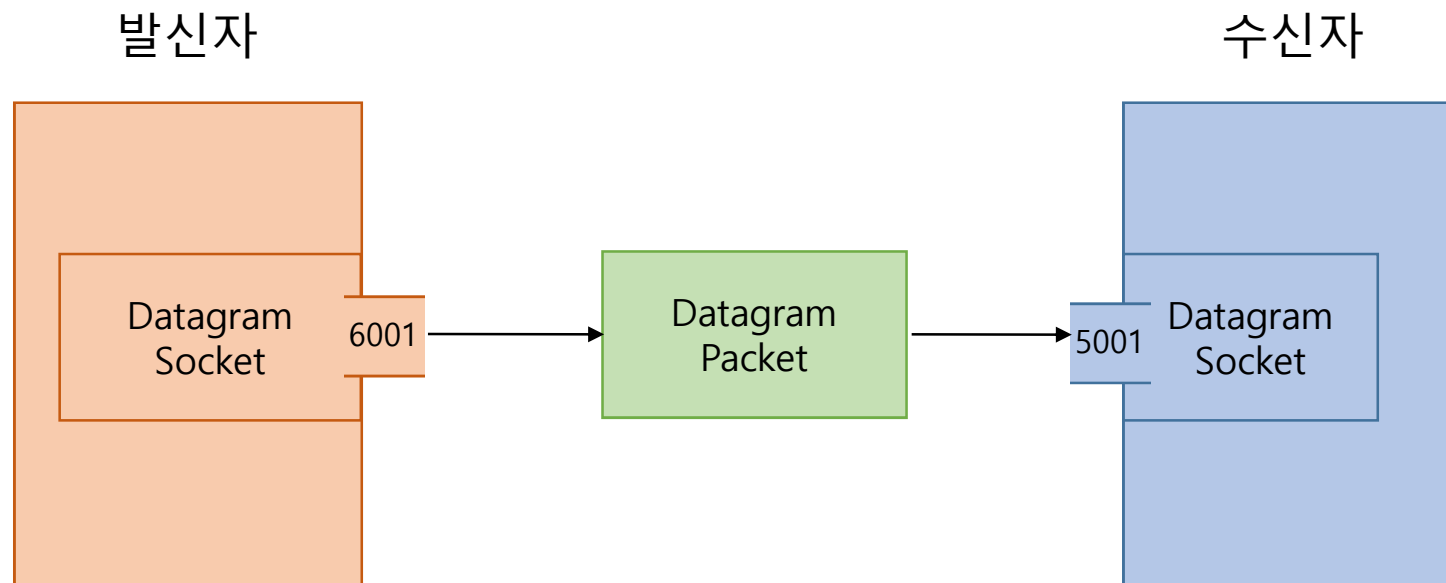
✓ 클라이언트용 TCP 소켓 프로그래밍 순서

1. 서버의 IP주소와 서버가 정한 포트번호를 매개변수로 하여 클라이언트용 소켓 객체 생성
2. 서버와의 입출력 스트림 오픈
3. 보조 스트림을 통해 성능 개선
4. 스트림을 통해 읽고 쓰기
5. 통신 종료

▶ UDP 소켓 프로그래밍

UDP는 연결 지향적이지 않기 때문에 연결 요청을 받아줄
서버 소켓이 필요 없음

java.net패키지에서 제공하는 두 개의 DatagramSocket간에
DatagramPacket으로 변환된 데이터 주고 받음



▶ UDP 소켓 프로그래밍

✓ 서버용 UDP 소켓 프로그래밍 순서

1. 서버의 포트번호 정함
2. DatagramSocket 객체
3. 연결한 클라이언트 IP주소를 가진 InetAddress 객체 생성
4. 전송할 메시지를 byte[]로 바꿈
5. 전송할 메시지를 DatagramPacket 객체에 담음
6. 소켓 레퍼런스를 사용하여 메시지 전송
7. 소켓 닫음

▶ UDP 소켓 프로그래밍

✓ 클라이언트용 UDP 소켓 프로그래밍 순서

1. 서버가 보낸 메시지를 받을 byte[] 준비
2. DatagramSocket 객체 생성
3. 메시지 받을 DatagramPacket 객체 준비
4. byte[]로 받은 메시지를 String으로 바꾸어 출력
5. 소켓 닫음