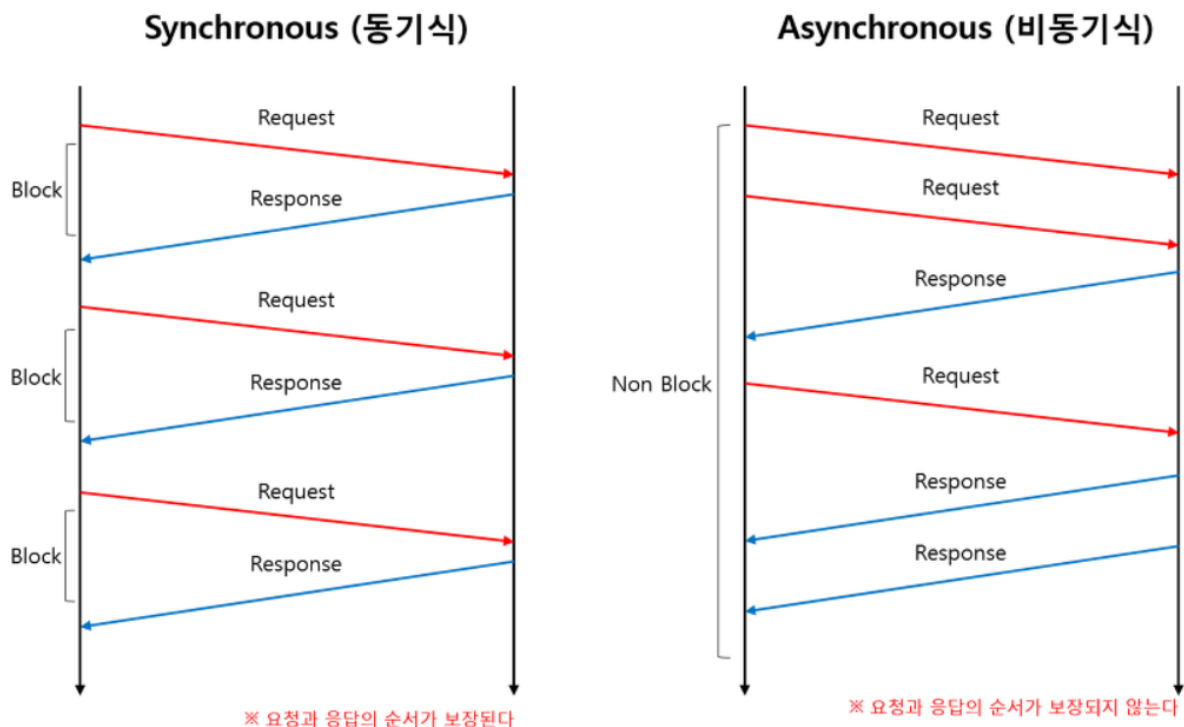


동기 통신과 비동기 통신



Synchronous (동기) 통신

- request를 보내면 시간이 얼마나 걸리든 그 자리에서 response를 받는다
- 두 서버 사이의 transaction을 맞춘다
- request를 보낸 thread는 response가 도착하기 전까지는 아무것도 하지 못하는 **block** 상태가 된다
- 요청과 응답값의 순서를 보장한다

장점

- request에 대한 처리 결과 값을 보장받을 수 있다
- 설계가 매우 간단하고 직관적이다

단점

- 결과가 주어질 때까지 아무것도 못하고 대기해야 한다
- response가 지연되게 된다면 request를 보낸 thread는 해당 response를 무작정 기다리는 상태가 되어 뒤에 들어오는 request들은 connection 가능한 thread가 없

어 연결을 맺지 못하는 성능적인 이슈가 발생할 수 있다

Asynchronous (비동기) 통신

- request를 보내고 response를 언제 받아도 상관없다
- response를 기다리지 않아 request를 보낸 thread는 다른 일을 할 수 있는 **non block** 상태이다
- response 순서를 보장하지 않는다

장점

- response를 기다리지 않고 non block 상태로 계속 자기 일을 하기 때문에 동기식 방식에 비해 성능적으로 좋다
- 결과가 주어지는데 시간이 걸리더라도 그 시간동안 다른 작업을 할 수 있으므로 자원을 효율적으로 사용할 수 있다

단점

- response에 대한 처리 결과를 보장받고 처리해야 되는 서비스에는 적합하지 않다

처리결과에 의해 처리하여 서비스의 질을 높일 수 있다 → synchronous

처리결과에 의존하지 않고 성능적으로 빠른 처리가 가능하다 → asynchronous