

Network Socket Communication

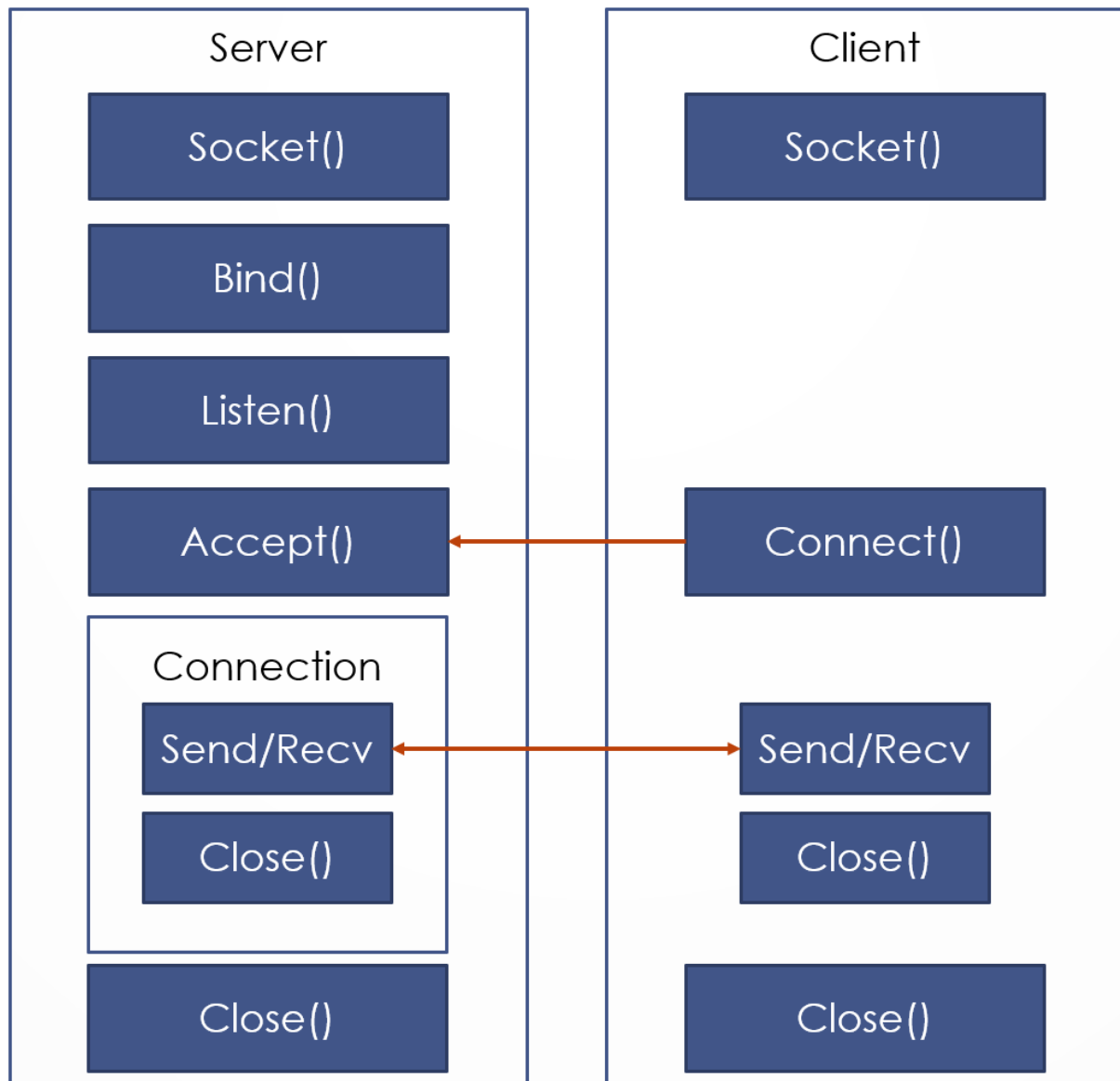
▼ Index

- [1. 통신이란?](#)
- [2. Socket Communication Protocol](#)
- [3. Synchronous & Asynchronous](#)

1. 통신이란?

- PC와 PC 즉, Program과 Program간 데이터를 주고 받는 것을 통신이라고 한다.
- 전송하는 패킷(데이터)이 컴퓨터의 NIC → Lan Cable 전송됨
- 전송된 데이터는 DNS와 라우터 등의 네트워크 장비들을 거치며 도달하고자하는 PC의 NIC로 도착하며, 목표로 하는 프로그램에서 패킷(데이터)를 읽어 서로 간에 데이터를 주고 받는다.

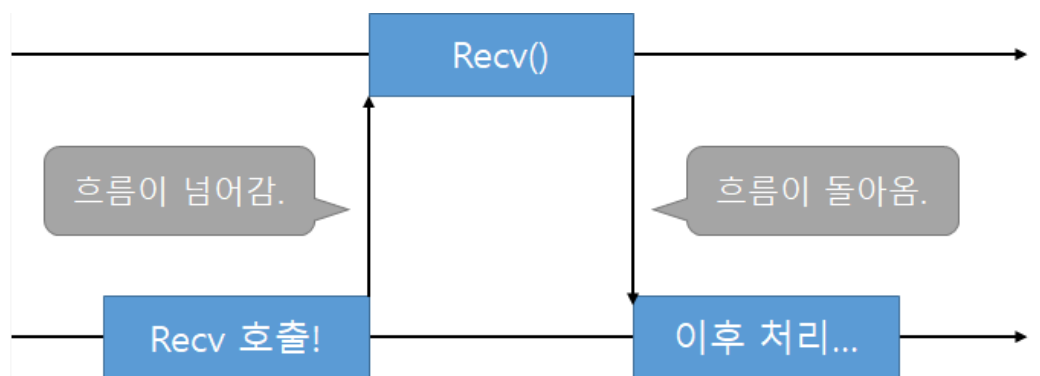
2. Socket Communication Protocol



3. Synchronous & Asynchronous

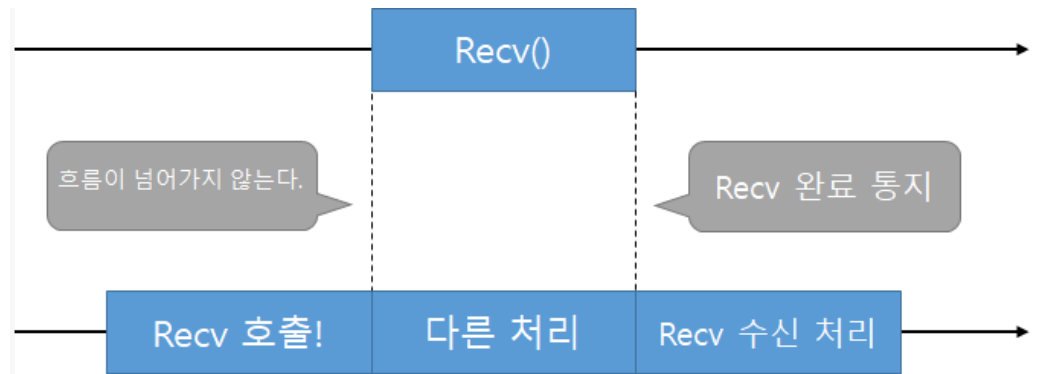
- 우리가 흔히 이야기하는 동기/비동기
 - 동기 : 서버와 클라이언트간 접속 및 데이터 전송, 연결 지속
 - 서버는 누군가가 접속할 때 까지 대기 (Listen)하고 있는 상태이며, 클라이언트는 서버에 접속 요청(Connect) 후 응답할 때까지 대기하고 있는 상태
 - 비동기 : 서버와 클라이언트가 접속 및 대용량 데이터 전송, 연결 종료
- Blocking & Non-Blocking
 - Blocking

- 정의 : 흐름 정지, 동기식으로 설계 시, 응답이 오기 전까지 다음 루틴으로 넘어가는 흐름을 막는 것
 - Blocking Function : 호출 시 응답이 올 때까지 대기하는 함수
- Non-Blocking
 - 정의, 흐름 유지, 비동기식으로 설계 시, 응답이 오든 안오든 상관 없이 다음 루틴으로 넘어가는 흐름 유지
 - Non-Blocking Function : 호출시 응답 여부 상관없이 호출만 하고 다음 루틴으로 넘어감
- IOCP의 동기/비동기
 - > 시스템 내부에서 하나의 쓰레드로 리소스를 유지하냐 안하느냐의 차이
 - 동기 : 서버에 클라이언트가 접속시 하나의 Socket(=Connection)과 Thread를 생성하여 객체를 유지
 - 보통 Vector나 List를 통해 Socket과 Thread 보관
 - 단점
 - 다중 접속시 한정된 자원으로 인한 무한히 Thread와 Socket를 생성할 수 없다.
 - C# 특성상 메모리 해제가 불가능하지만, GC가 존재하는데, GC를 컨트롤 할 수 없기 때문에, 리소스를 마구잡이로 사용하다가 GC 동작 시 서버 Freezing 상태가 될 수 있다.
 - 흐름이 하나로 진행(동기화)된다.



- 비동기 : 서버에 클라이언트가 접속시 접속 시 처리를 위한 이벤트가 발생하고 연결 리소스를 Queue 구조의 IOCP에 쌓습니다.

- Client로부터 데이터 수신 시 IOCP에서 연결 리소스를 꺼내 프로그램으로 알려주는 방식
- 비동기 서버는 접속시마다 Thread를 쌓을 필요가 없기 때문에 리소스를 많이 아낄 수 있다. 또한 Thread를 관리하지 않기 때문에 Thread를 관리하기 위한 ThreadPool를 만들 필요 또한 없다.
- 흐름이 넘어가지 않고 진행된다.



- C# Asynchronous Type
 - TAP : Task-based Asynchronous Pattern
 - EAP : Event-based Asynchronous Pattern
 - APM : Asynchronous Programming Model