

Semana 3

PATRON SINGLETON

ENLACE TCP

IP Privadas

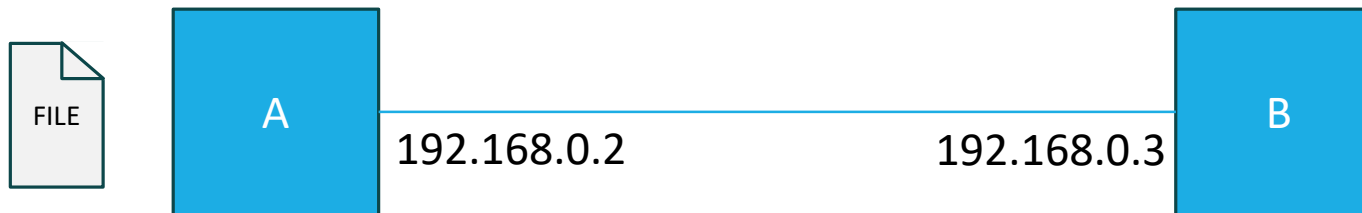
| Nombre | Rango | Número de direcciones | Tipo de clase | Bloque mayor |
|-------------------|-------------------------------|-----------------------|---------------------|----------------|
| Bloque de 24 bits | 10.0.0.0 – 10.255.255.255 | 16'777.215 | Clase A | 10.0.0.0/8 |
| Bloque de 20 bits | 172.16.0.0 – 172.31.255.255 | 1'048.576 | 16 Clases B juntas | 172.16.0.0/12 |
| Bloque de 16 bits | 192.168.0.0 – 192.168.255.255 | 65.335 | 256 Clases C juntas | 192.168.0.0/16 |

PROTOCOLO IP

| | Desde | A |
|---------|--|--|
| Clase A | 0.0.0.0 Identificador de red Identificador de estación | 127.255.255.255 Identificador de red Identificador de estación |
| Clase B | 128.0.0.0 Identificador de red Identificador de estación | 191.255.255.255 Identificador de red Identificador de estación |
| Clase C | 192.0.0.0 Identificador de red Identificador de estación | 223.255.255.255 Identificador de red Identificador de estación |
| Clase D | 224.0.0.0 Dirección de grupo | 239.255.255.255 Dirección de grupo |
| Clase E | 240.0.0.0 Indefinido | 247.255.255.255 Indefinido |

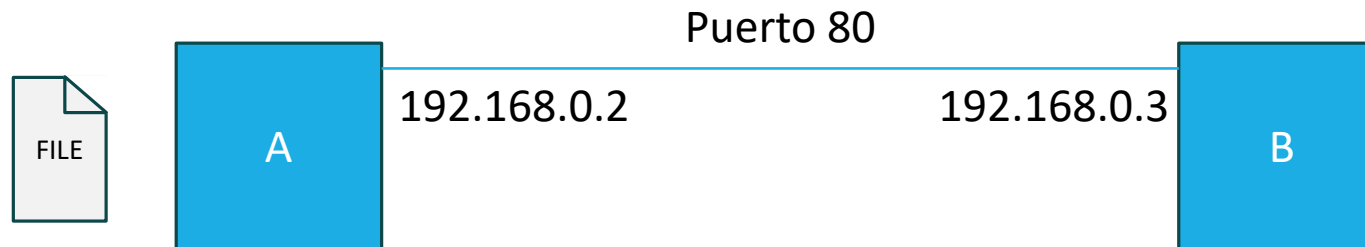
Fuente: Alejandro Llagua

TRANSMISIÓN DE DATOS



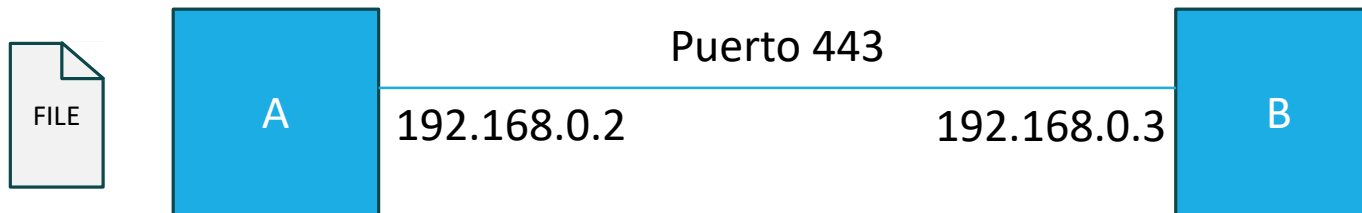
Usando TCP y UDP NO basta saber las direcciones IP, sino también el puerto por el que se envía la información.

TRANSMISIÓN DE DATOS



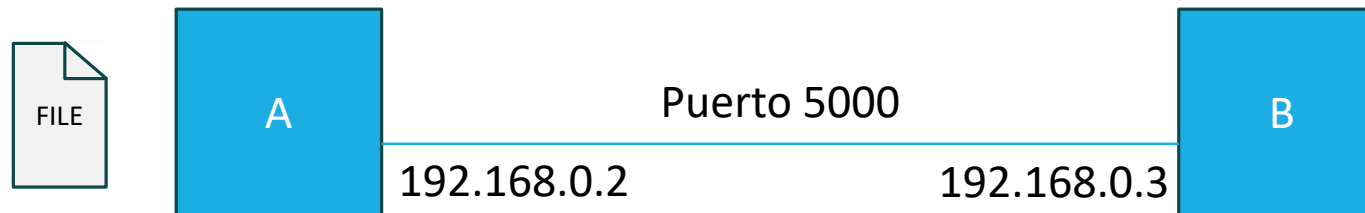
Usando TCP y UDP NO basta saber las direcciones IP, sino también el puerto por el que se envía la información.

TRANSMISIÓN DE DATOS



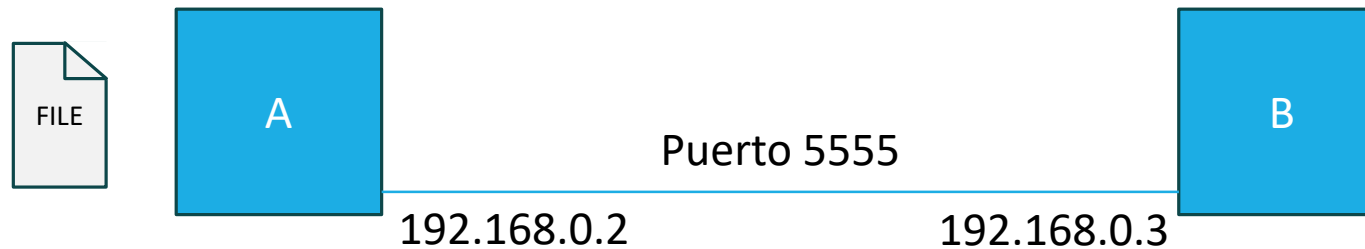
Usando TCP y UDP NO basta saber las direcciones IP, sino también el puerto por el que se envía la información.

TRANSMISIÓN DE DATOS



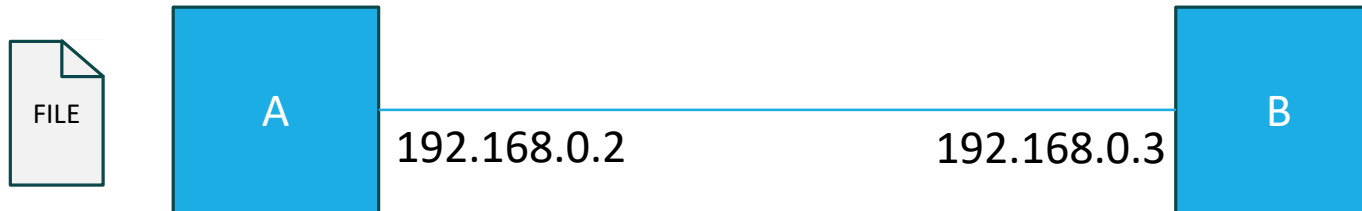
Usando TCP y UDP NO basta saber las direcciones IP, sino también el puerto por el que se envía la información.

TRANSMISIÓN DE DATOS



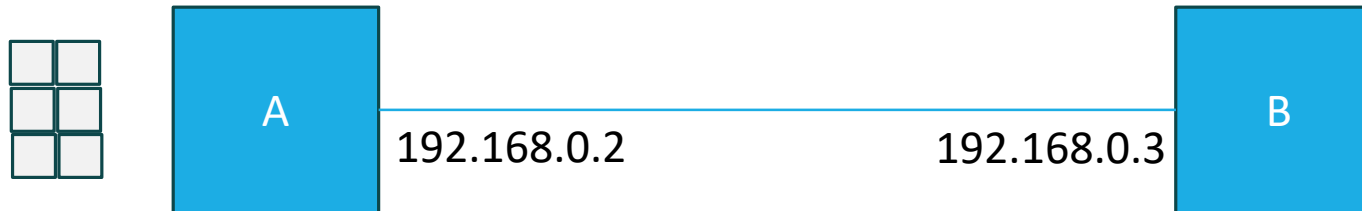
Usando TCP y UDP NO basta saber las direcciones IP, sino también el puerto por el que se envía la información.

TRANSMISIÓN DE DATOS



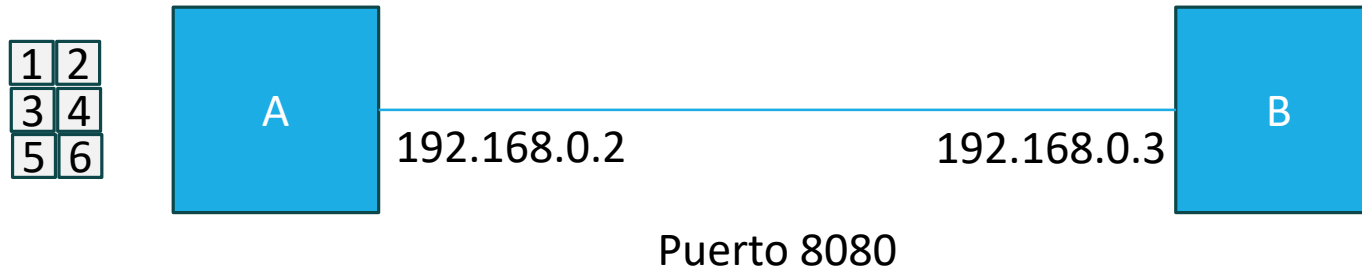
Esto permite establecer distintos servicios y separarlos por funciones. Algunos servicios son Transmisiones en vivo, Web, Web segura, Transferencia de archivos, e-mail.

TRANSMISIÓN DE DATOS



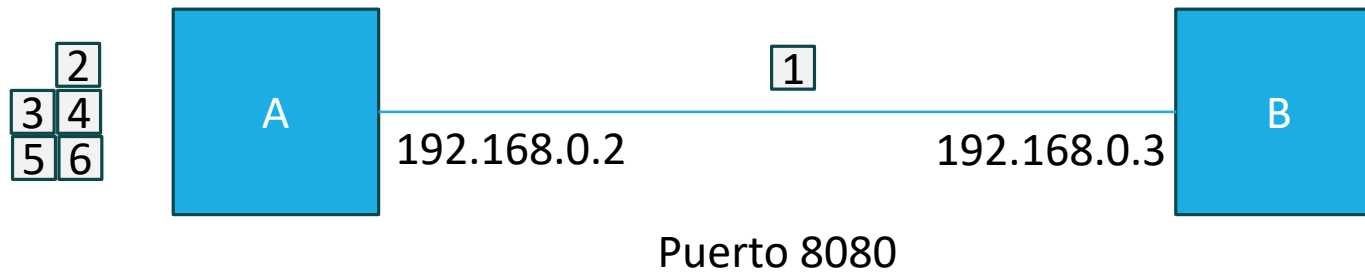
El protocolo IP funciona fragmentando la información en ***paquetes***

TRANSMISIÓN DE DATOS



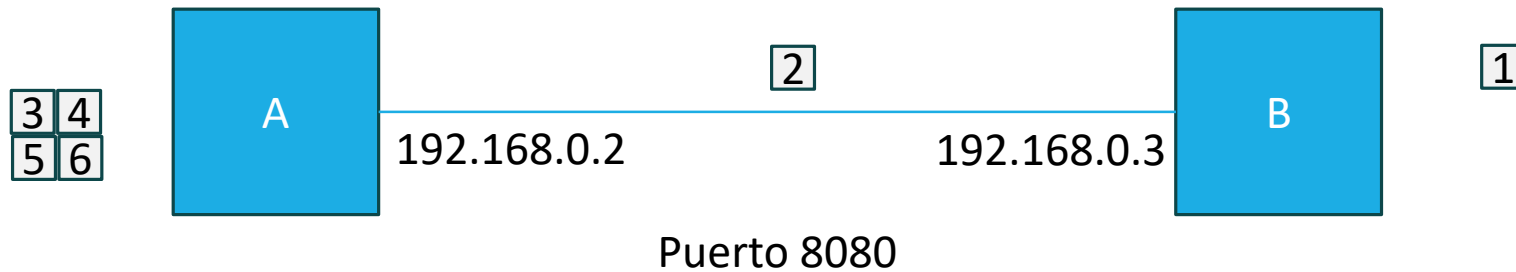
El protocolo IP funciona fragmentando la información en ***paquetes***

TRANSMISIÓN DE DATOS



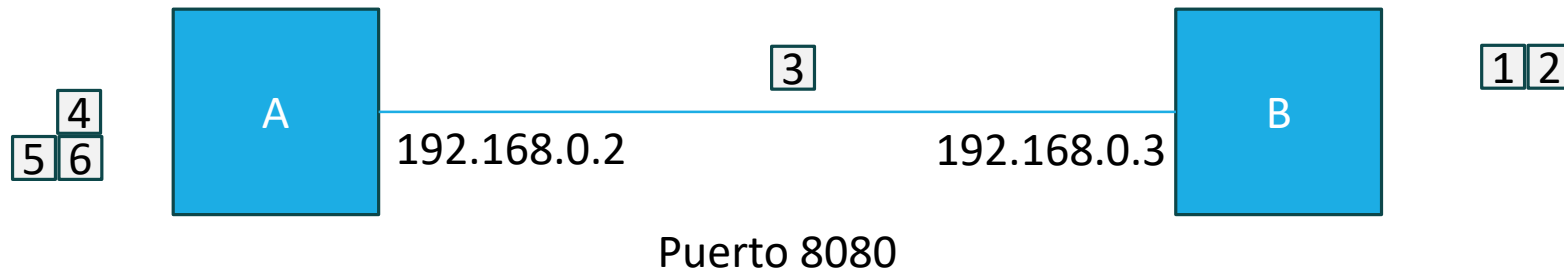
El protocolo IP funciona fragmentando la información en ***paquetes***

TRANSMISIÓN DE DATOS



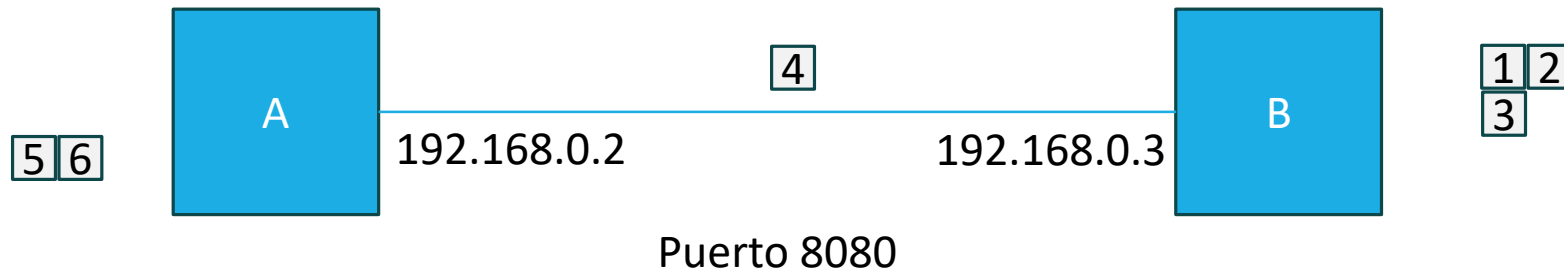
El protocolo IP funciona fragmentando la información en ***paquetes***

TRANSMISIÓN DE DATOS



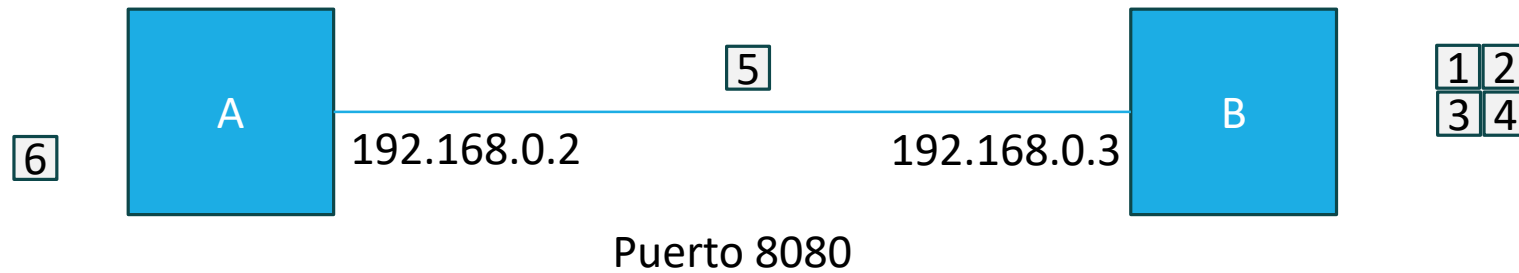
El protocolo IP funciona fragmentando la información en ***paquetes***

TRANSMISIÓN DE DATOS



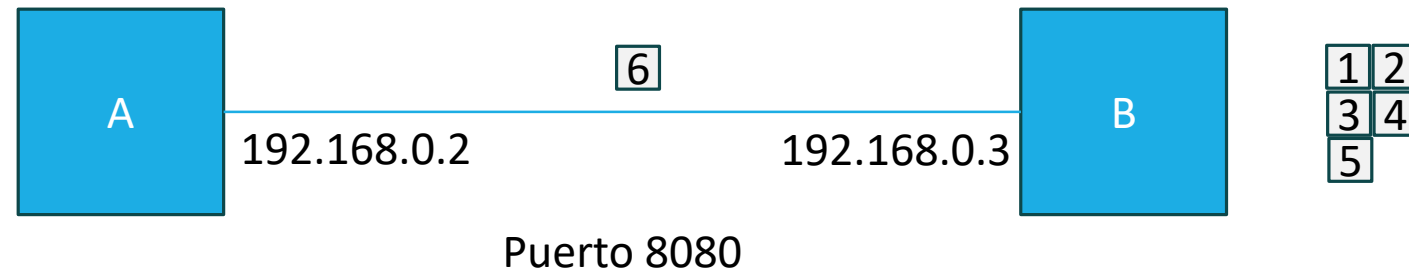
El protocolo IP funciona fragmentando la información en ***paquetes***

TRANSMISIÓN DE DATOS



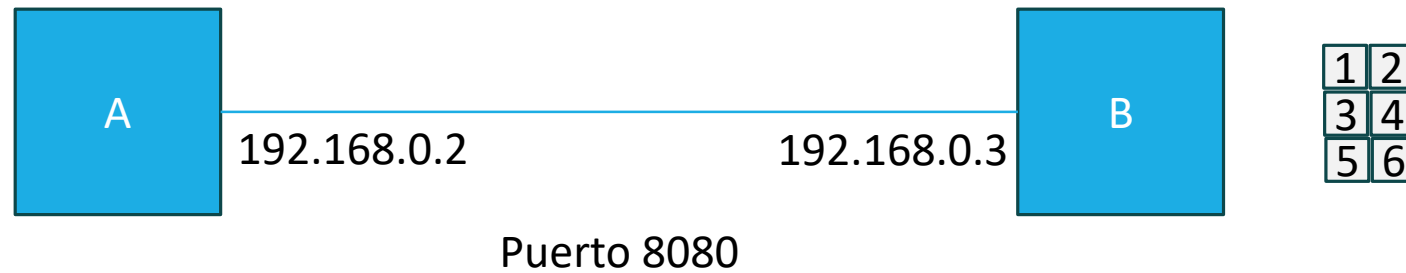
El protocolo IP funciona fragmentando la información en ***paquetes***

TRANSMISIÓN DE DATOS



El protocolo IP funciona fragmentando la información en ***paquetes***

TRANSMISIÓN DE DATOS



El protocolo IP funciona fragmentando la información en ***paquetes***

Puerto de red

- Los puertos están añadidos en la capa de transporte (Capa 4 del modelo OSI). Permite enviar y recibir mensajes simultáneamente de diversas aplicaciones.
- Para poder determinar de qué aplicación se trata, el encabezado y el número de puerto de red están definidos, según la aplicación.
- Los puertos tienen 2 bytes de extensión, por cual hay 65536 posibles puertos.
- Los puertos inferiores al 1024, son los **puertos bien conocidos**, usados aplicaciones del sistema. Entre 1024 y 49151 son **puertos registrados** por la IANA. El resto son **puertos dinámicos** usados para conexiones P2P

Puertos de red

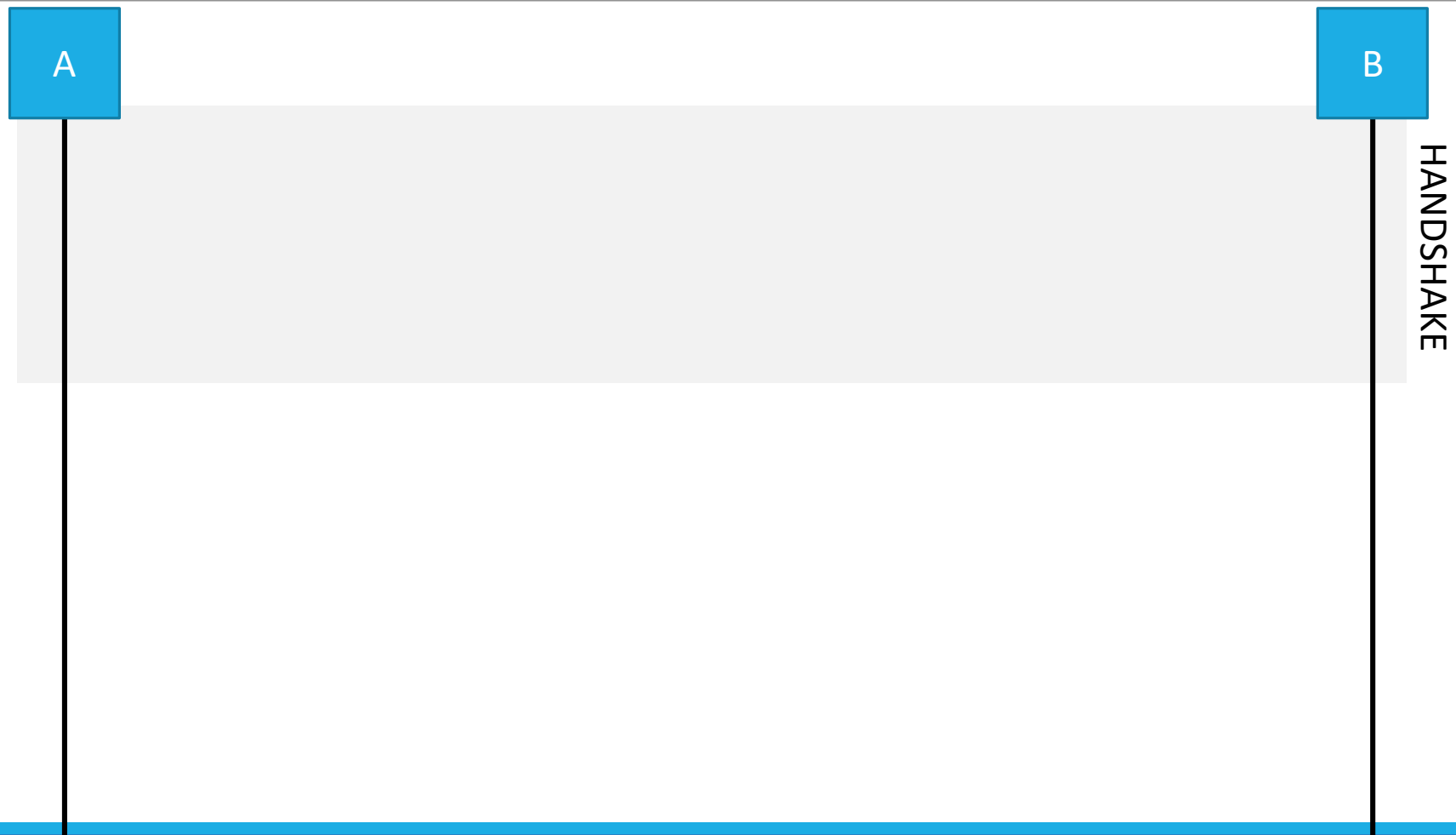
| Número | Protocolo o aplicación |
|--------|---------------------------------------|
| 1 | TCP port service multiplexer (TCPMUX) |
| 20 | FTP – Data |
| 21 | FTP – Control |
| 22 | SSH |
| 23 | Telnet |
| 53 | DNS |
| 80 | HTTP |
| 443 | HTTPS |
| 546 | DHCP Client |
| 547 | DHCP Server |
| 3306 | MySQL |

FLUJO TCP

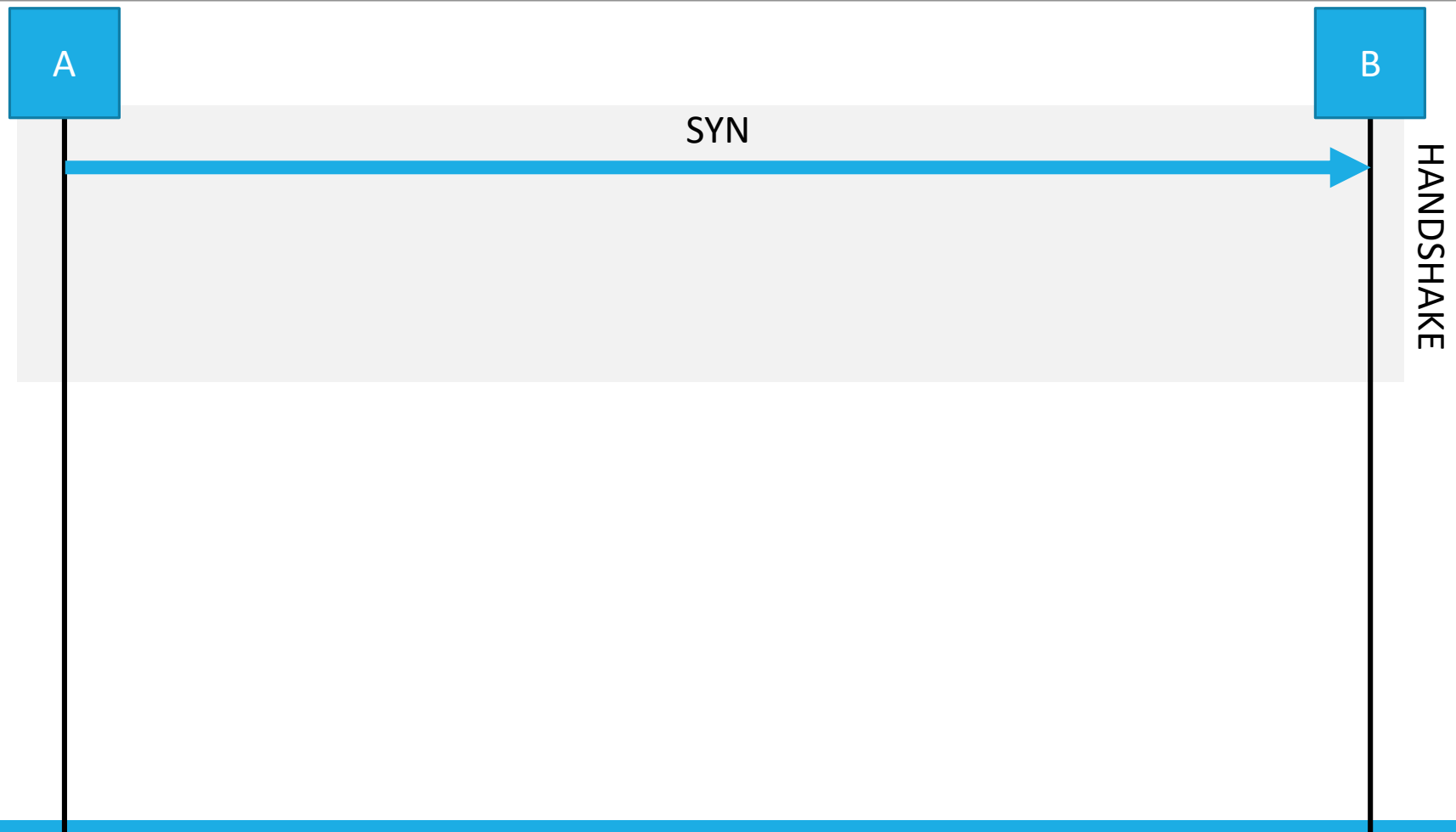
TCP



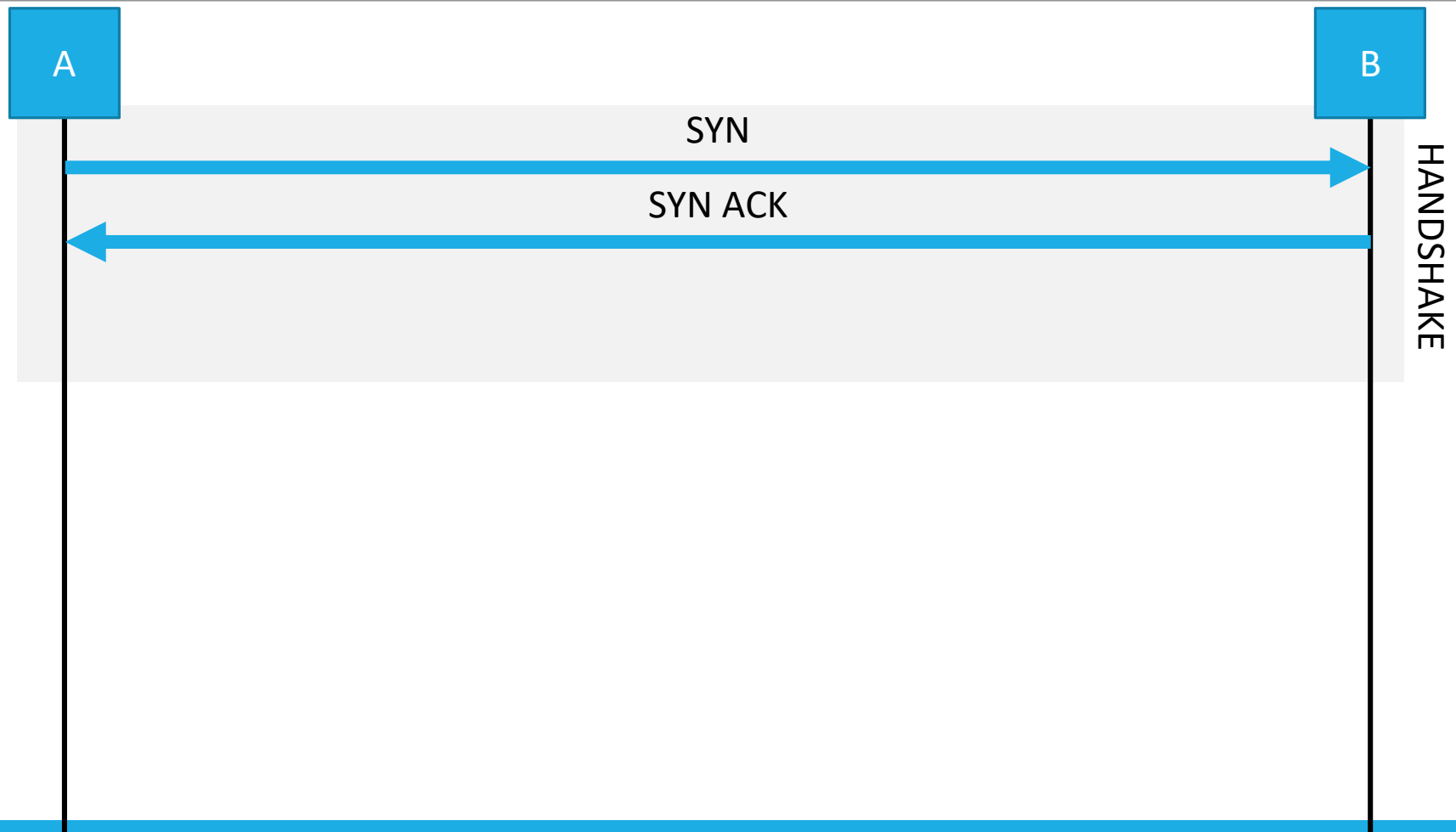
TCP



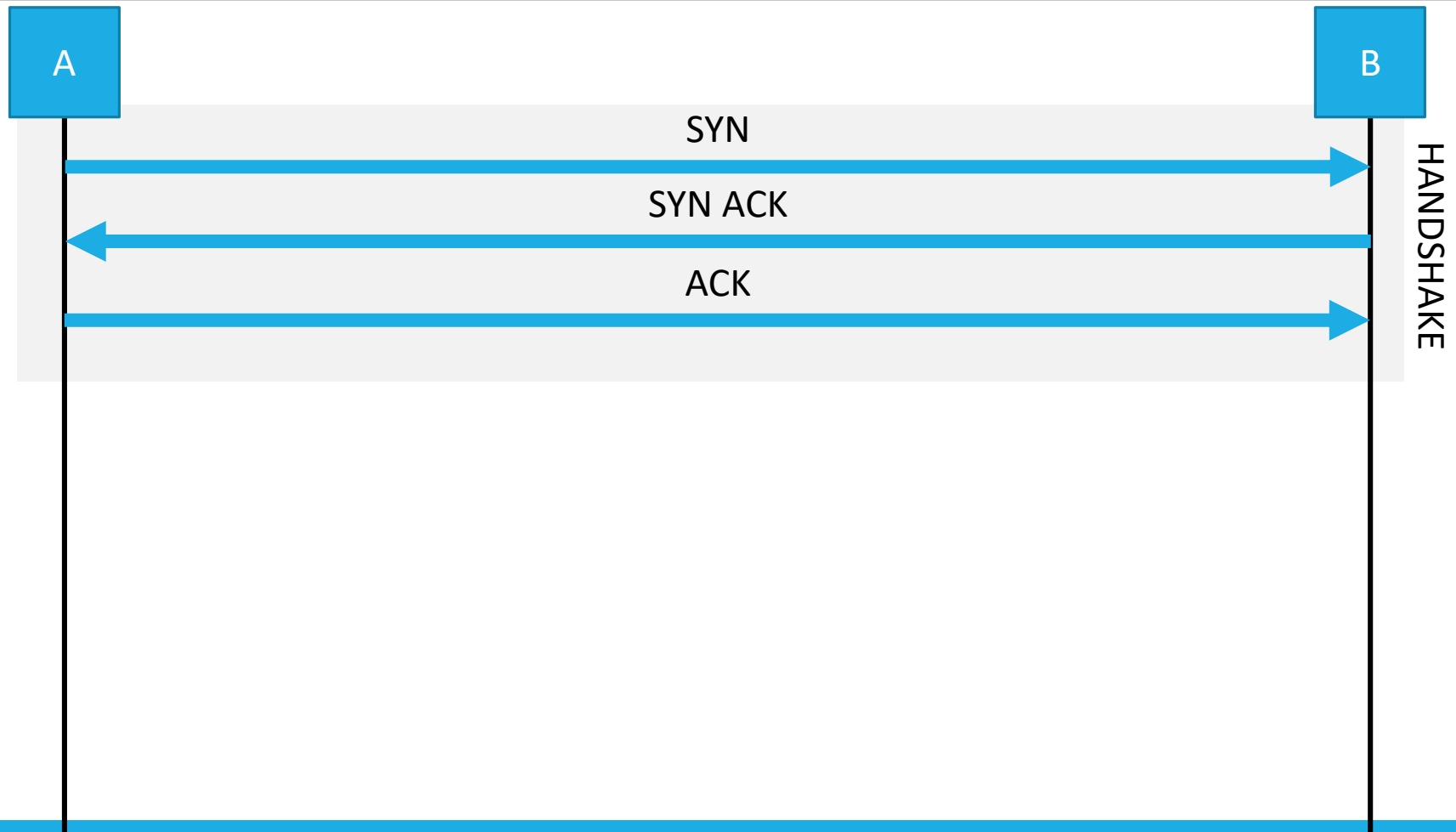
TCP



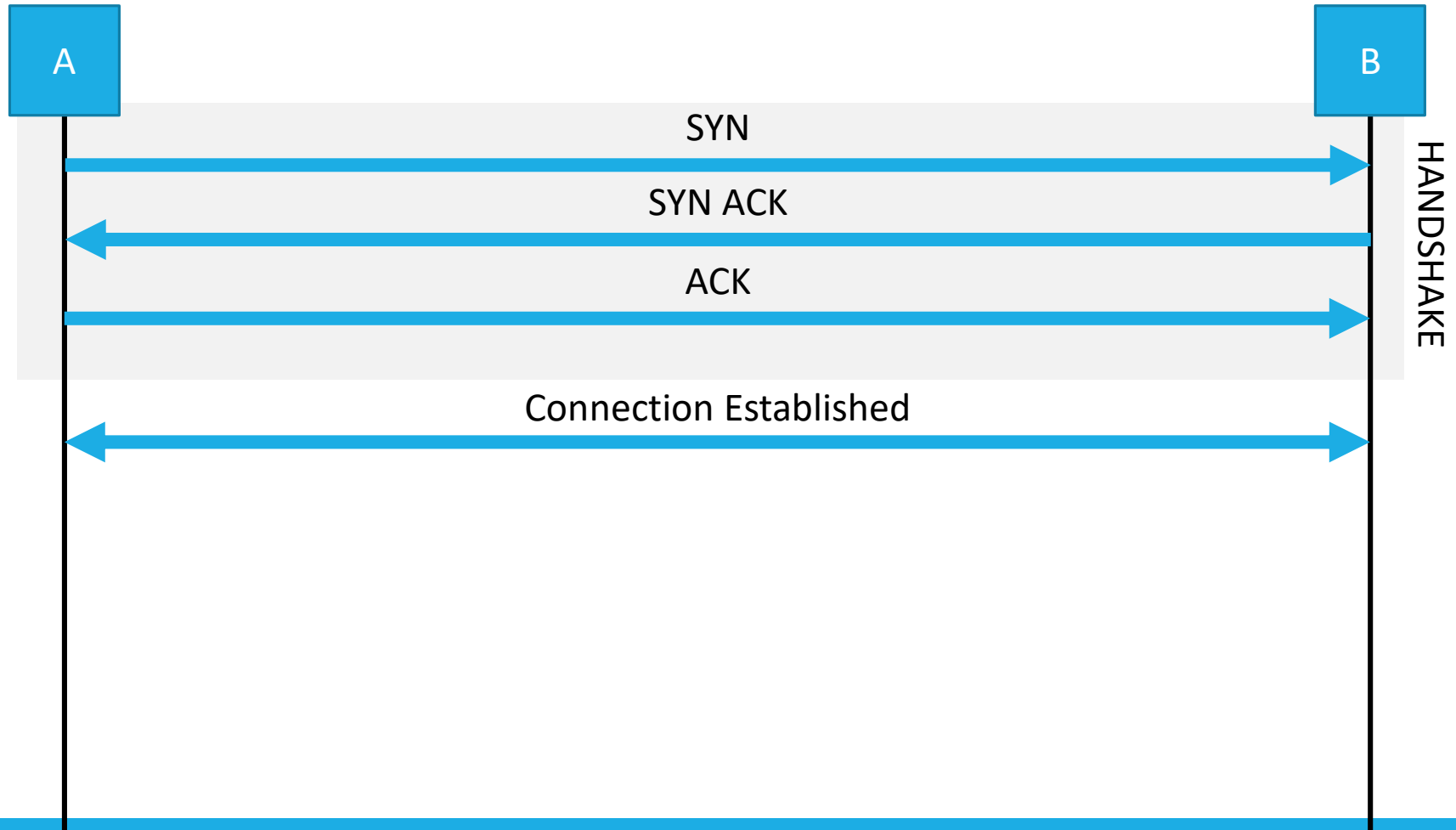
TCP



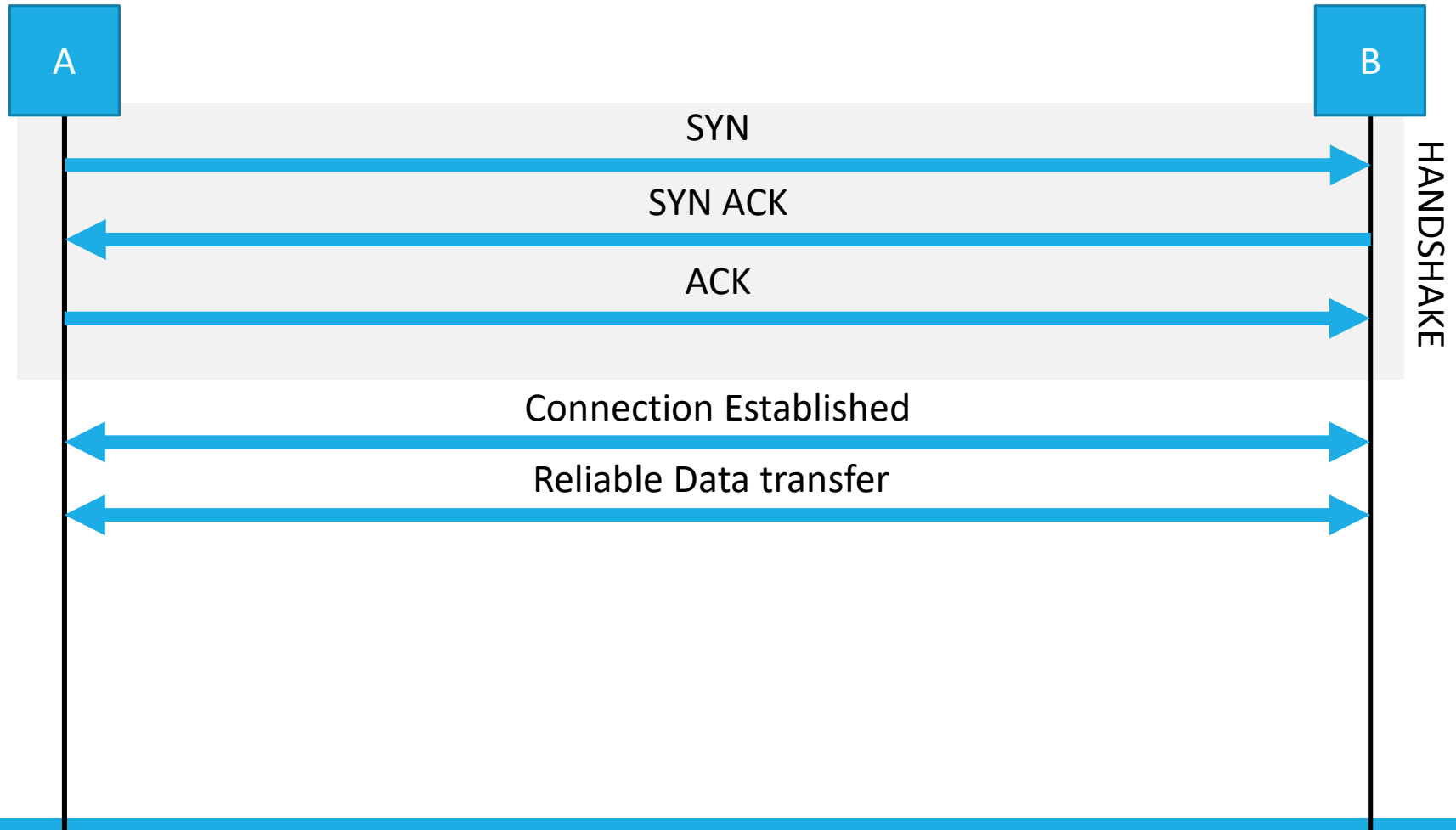
TCP



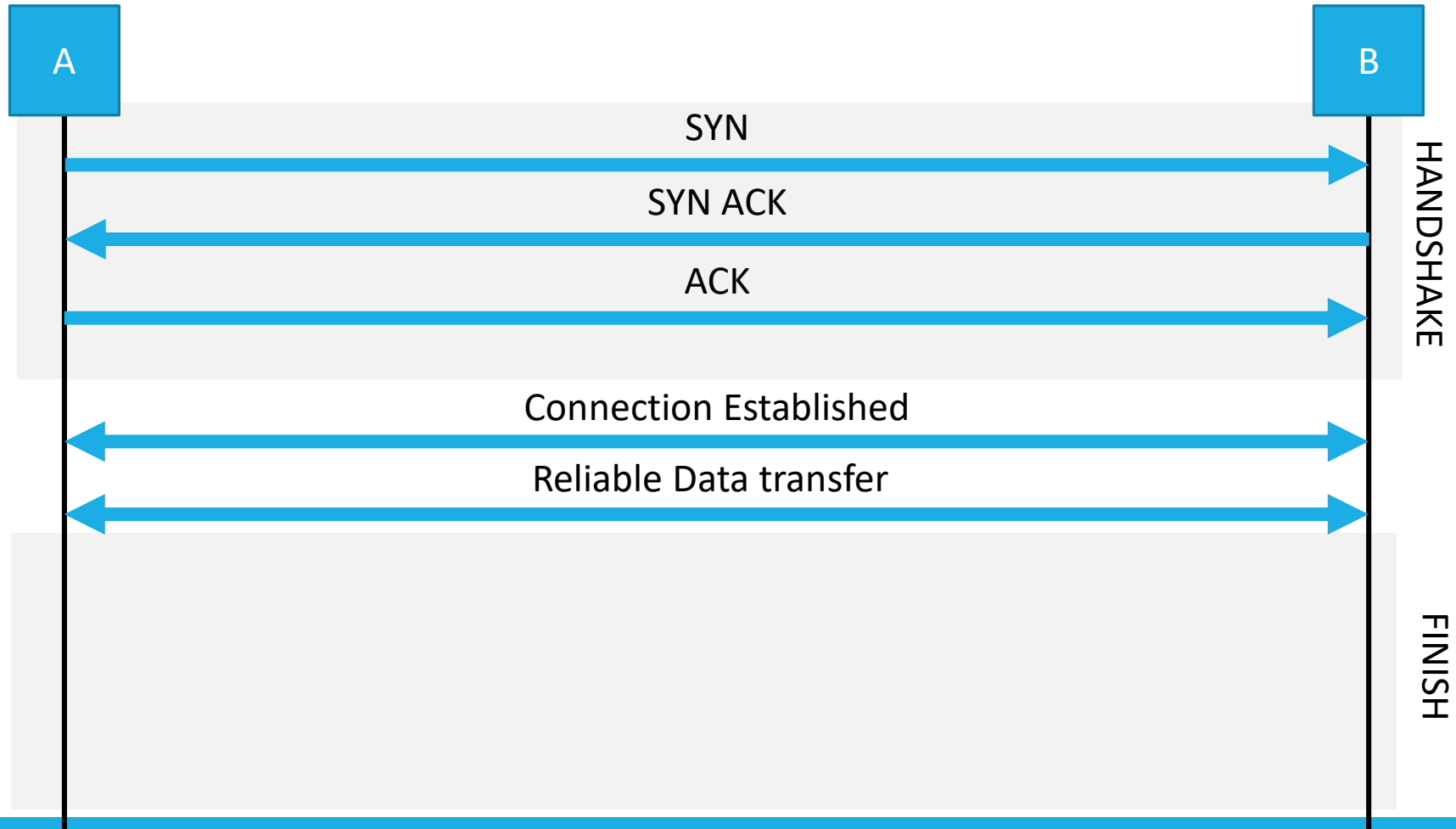
TCP



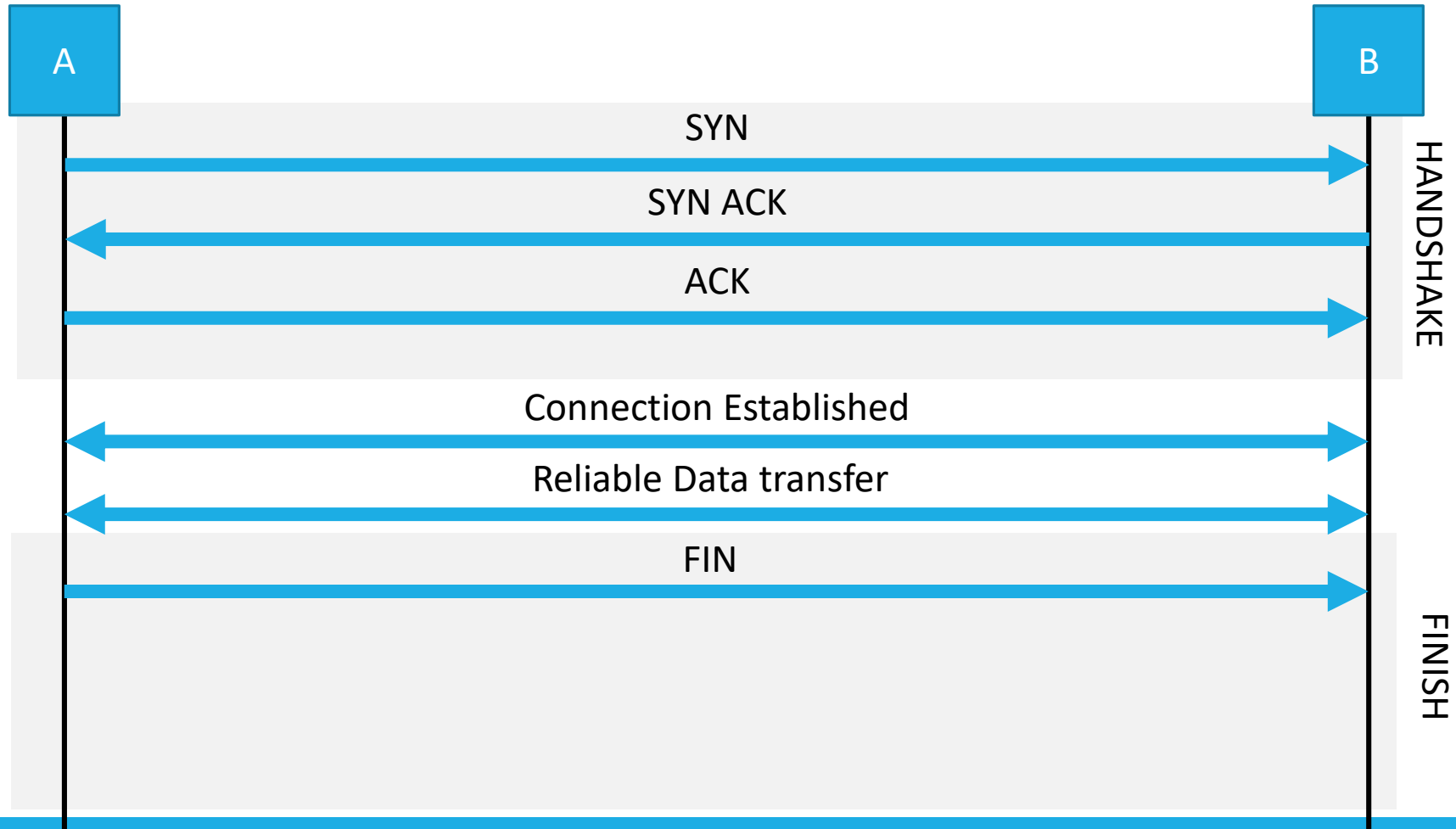
TCP



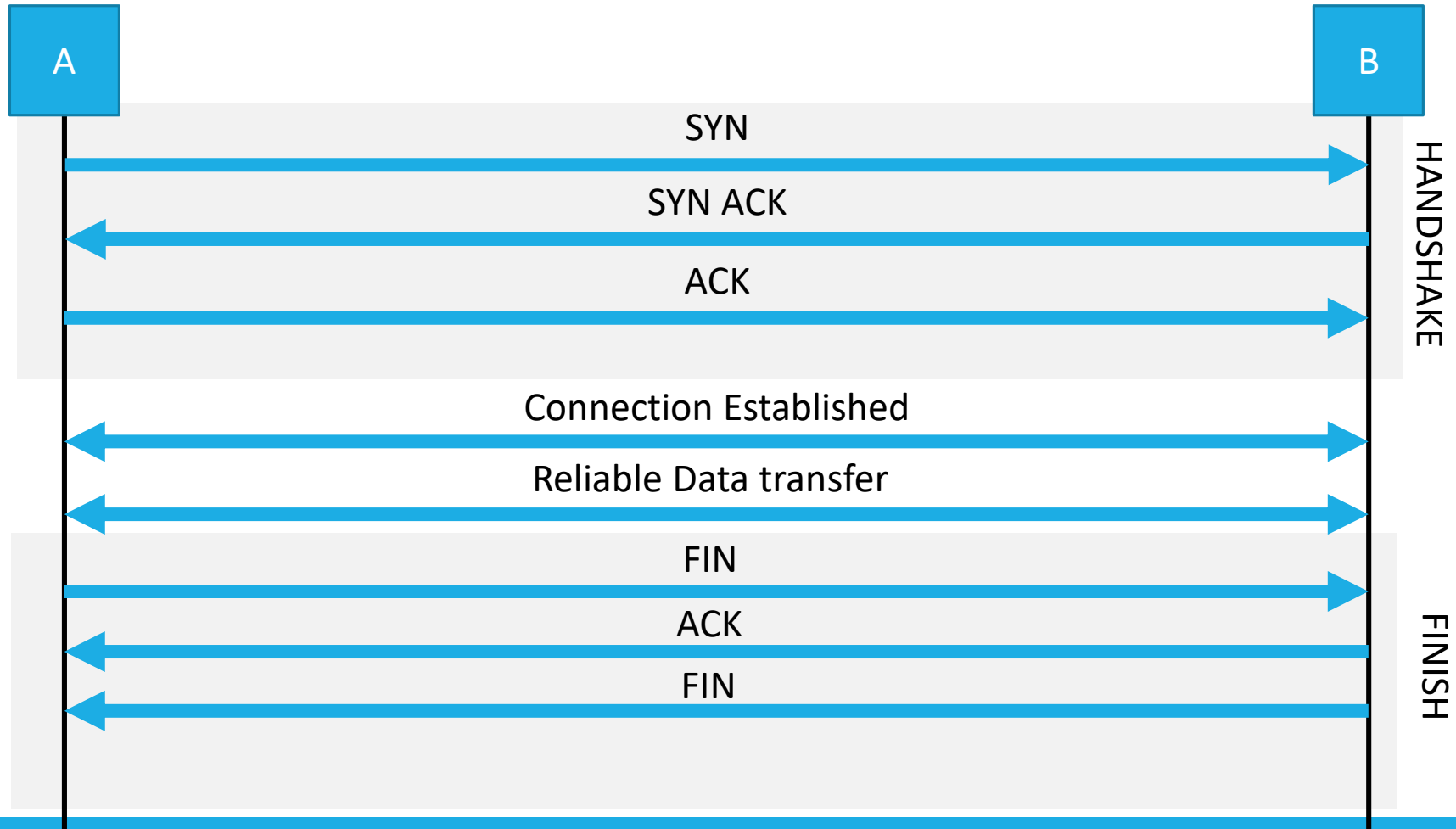
TCP



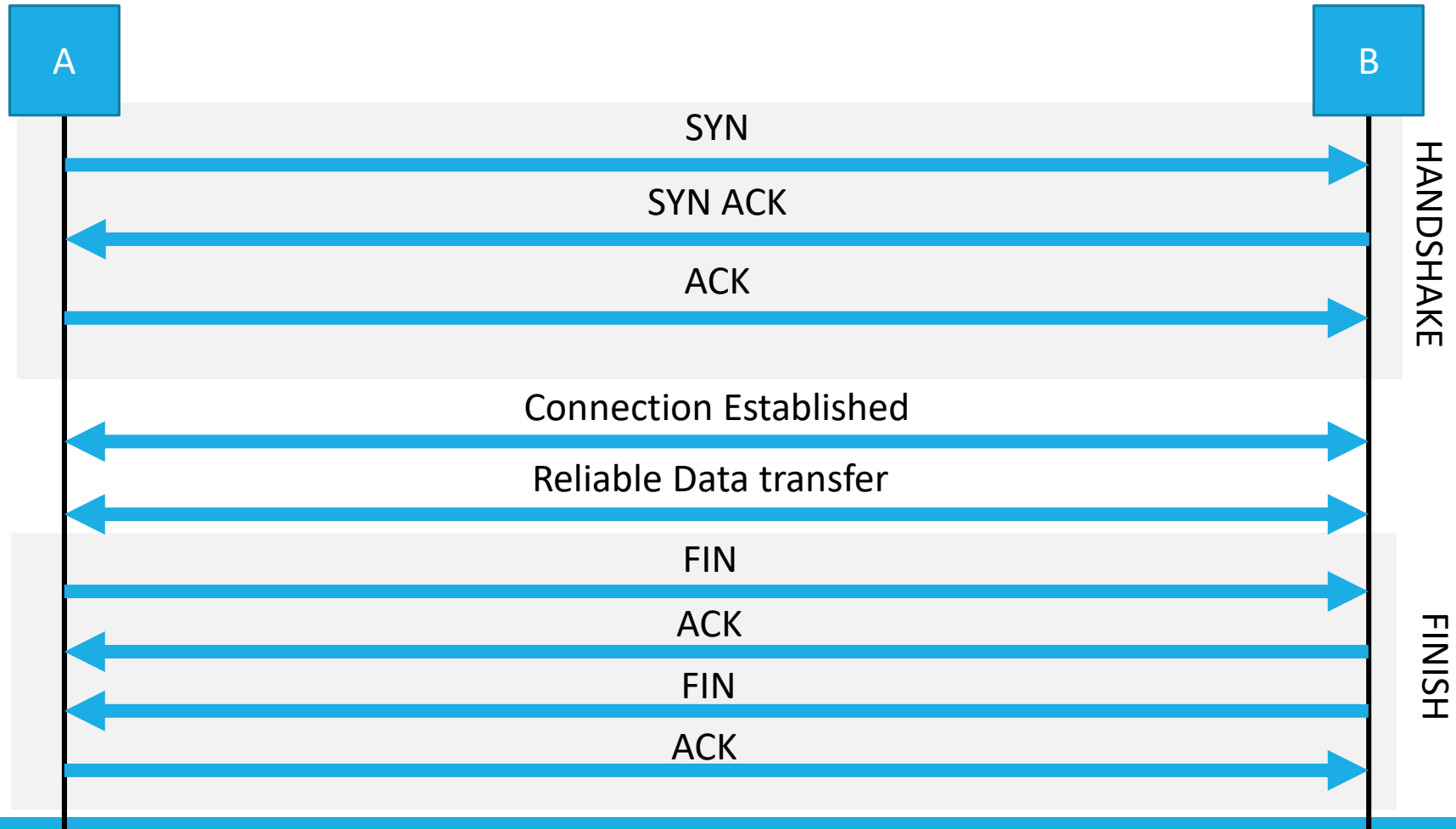
TCP



TCP



TCP

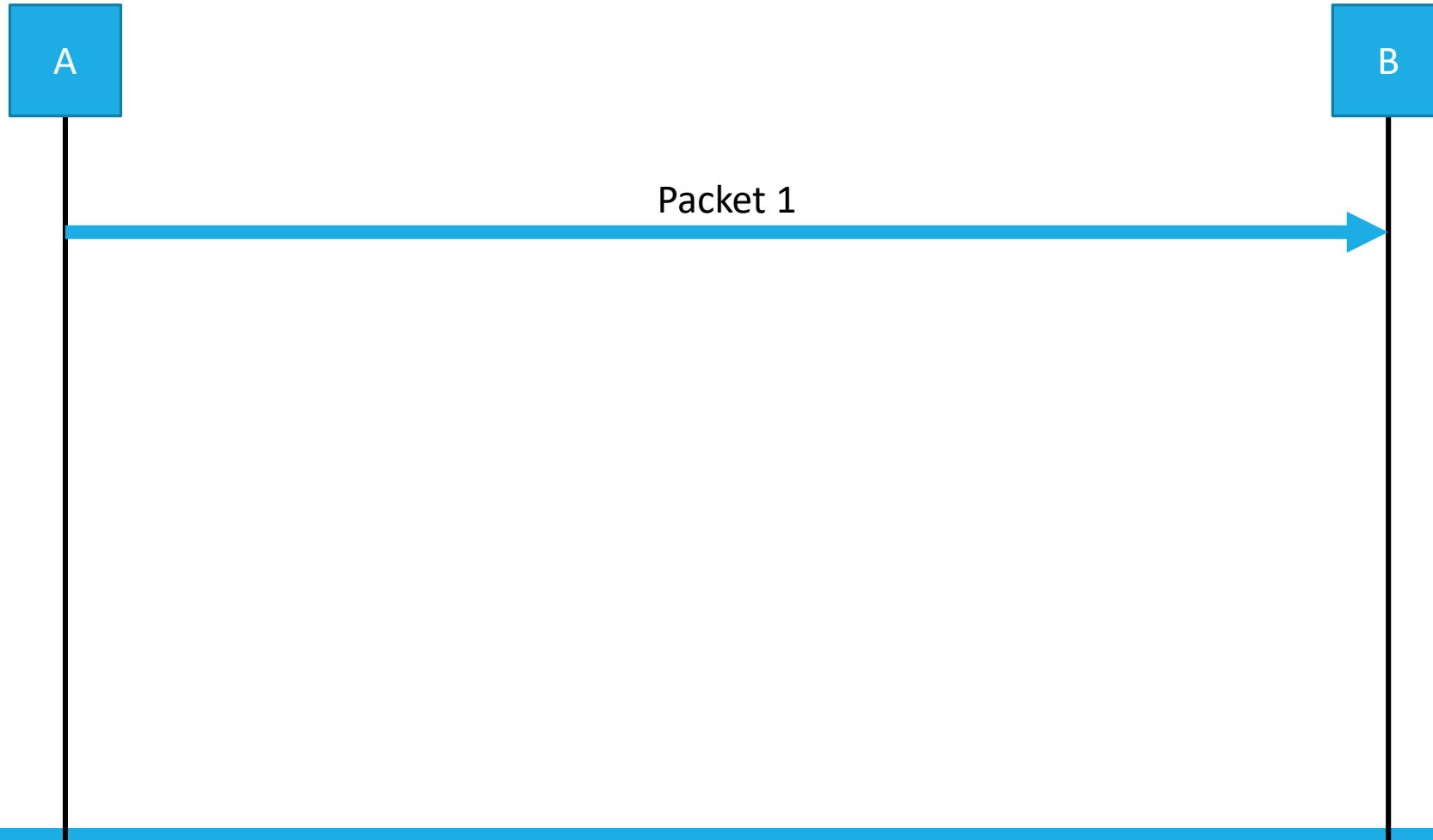


TRANSFERENCIA TCP

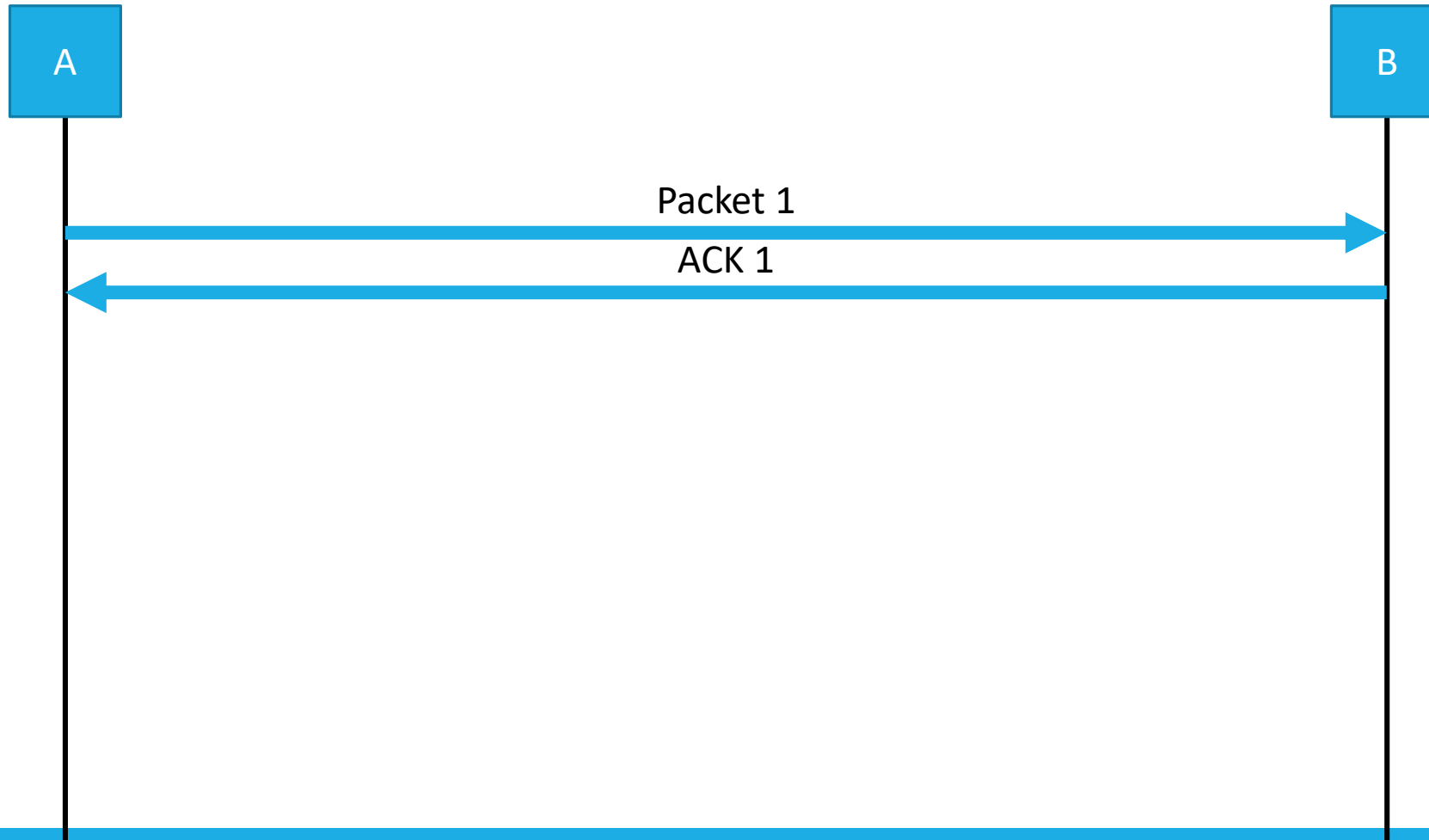
TCP Data transfer



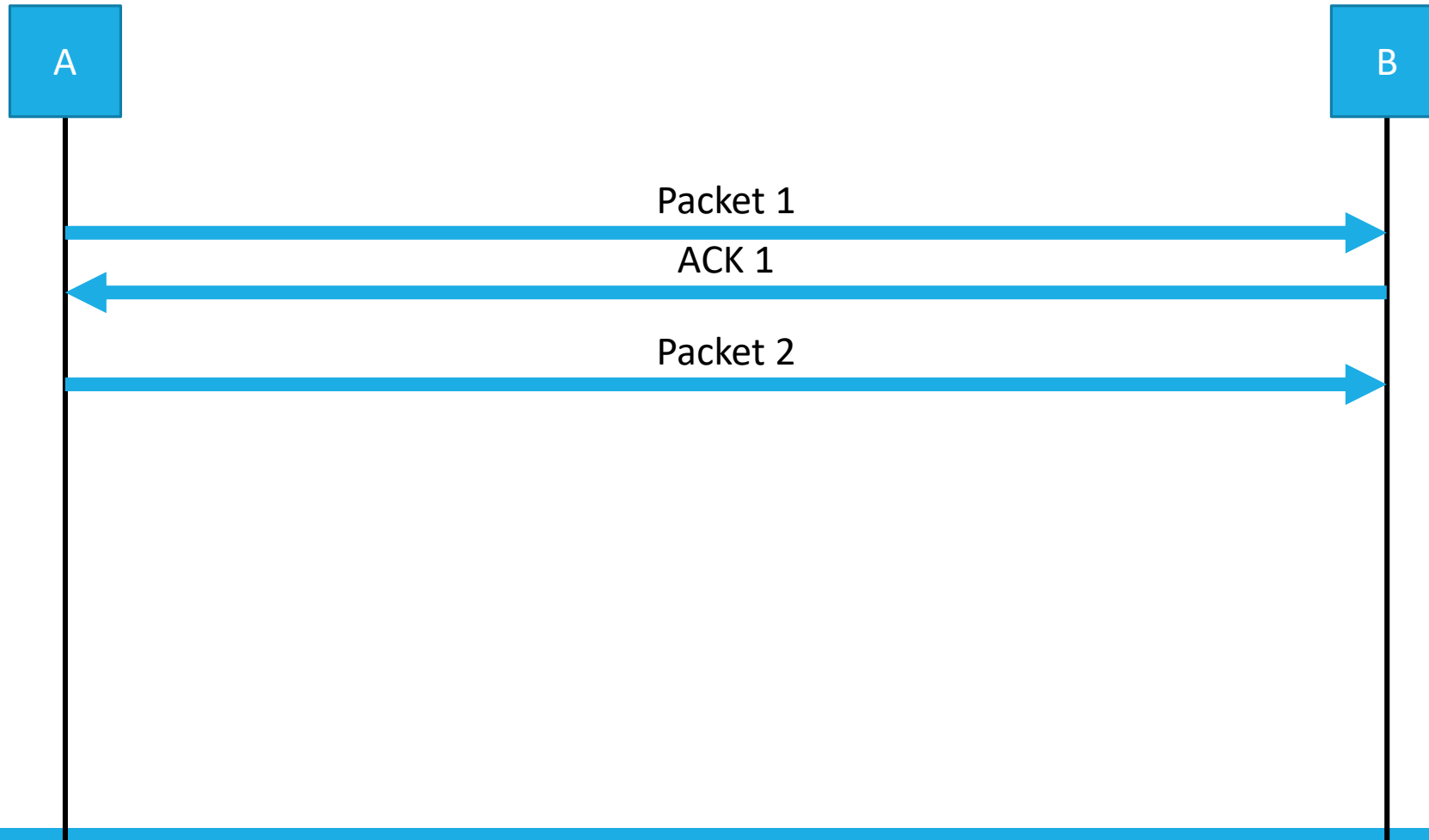
TCP Data transfer



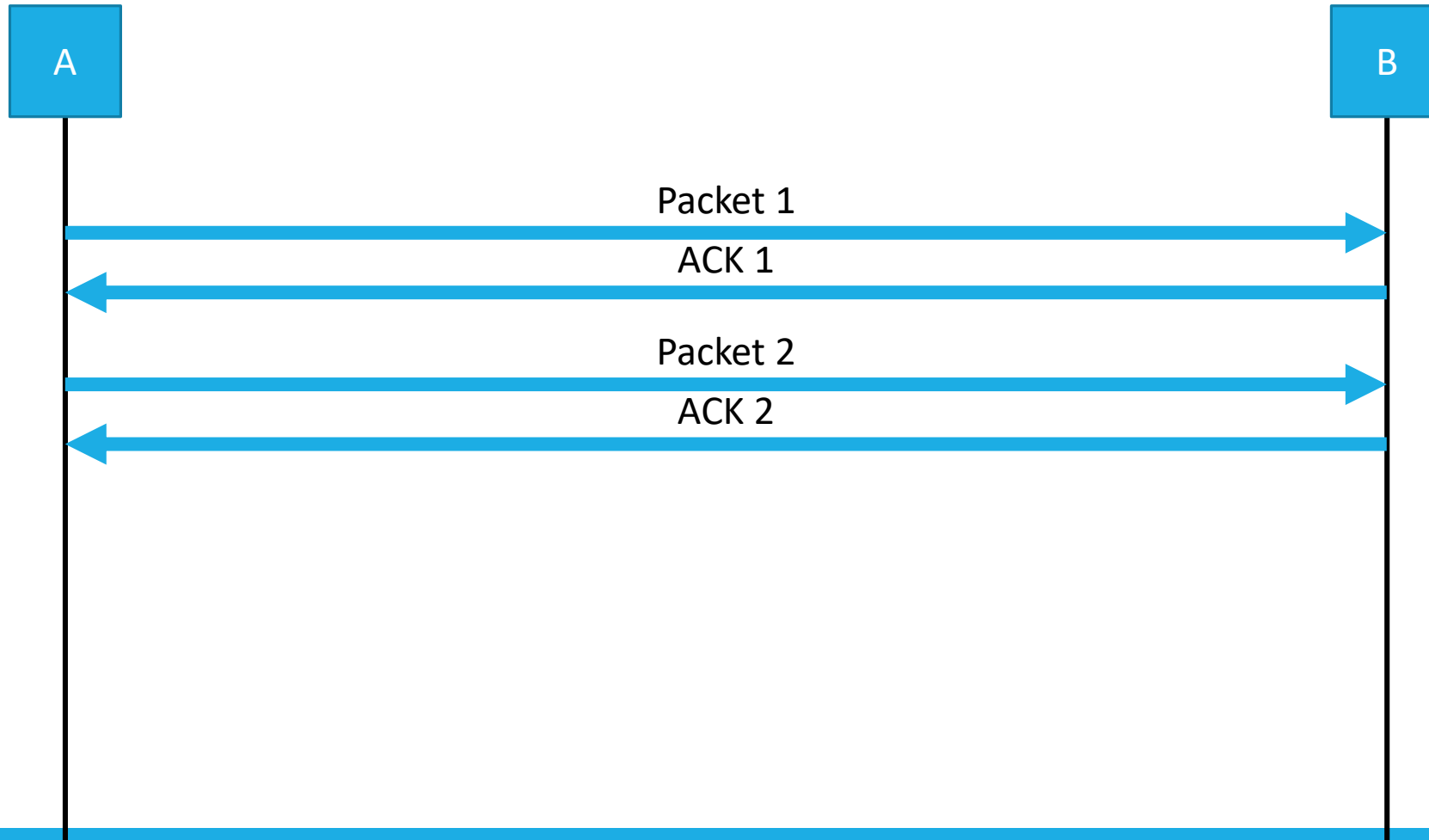
TCP Data transfer



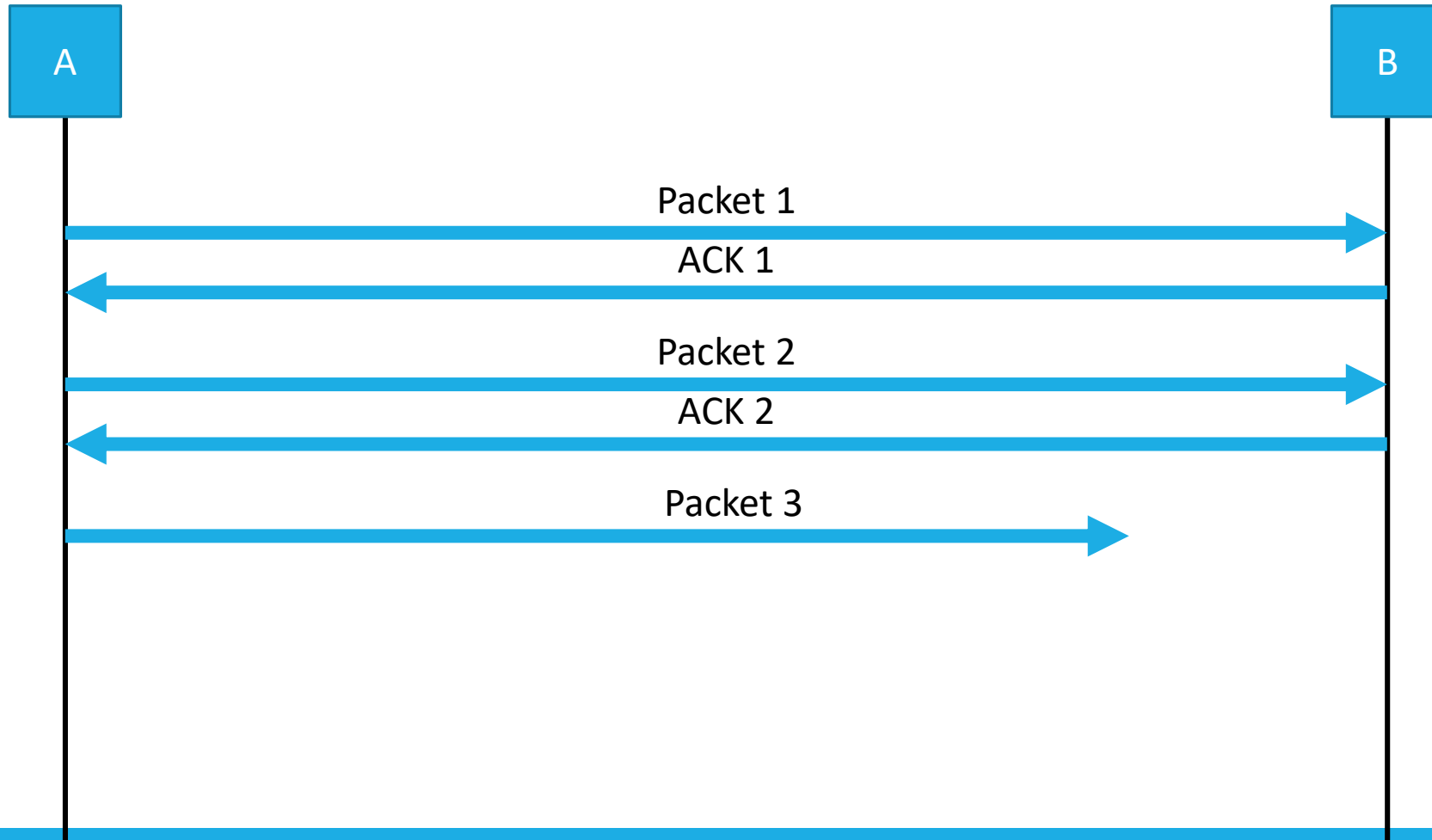
TCP Data transfer



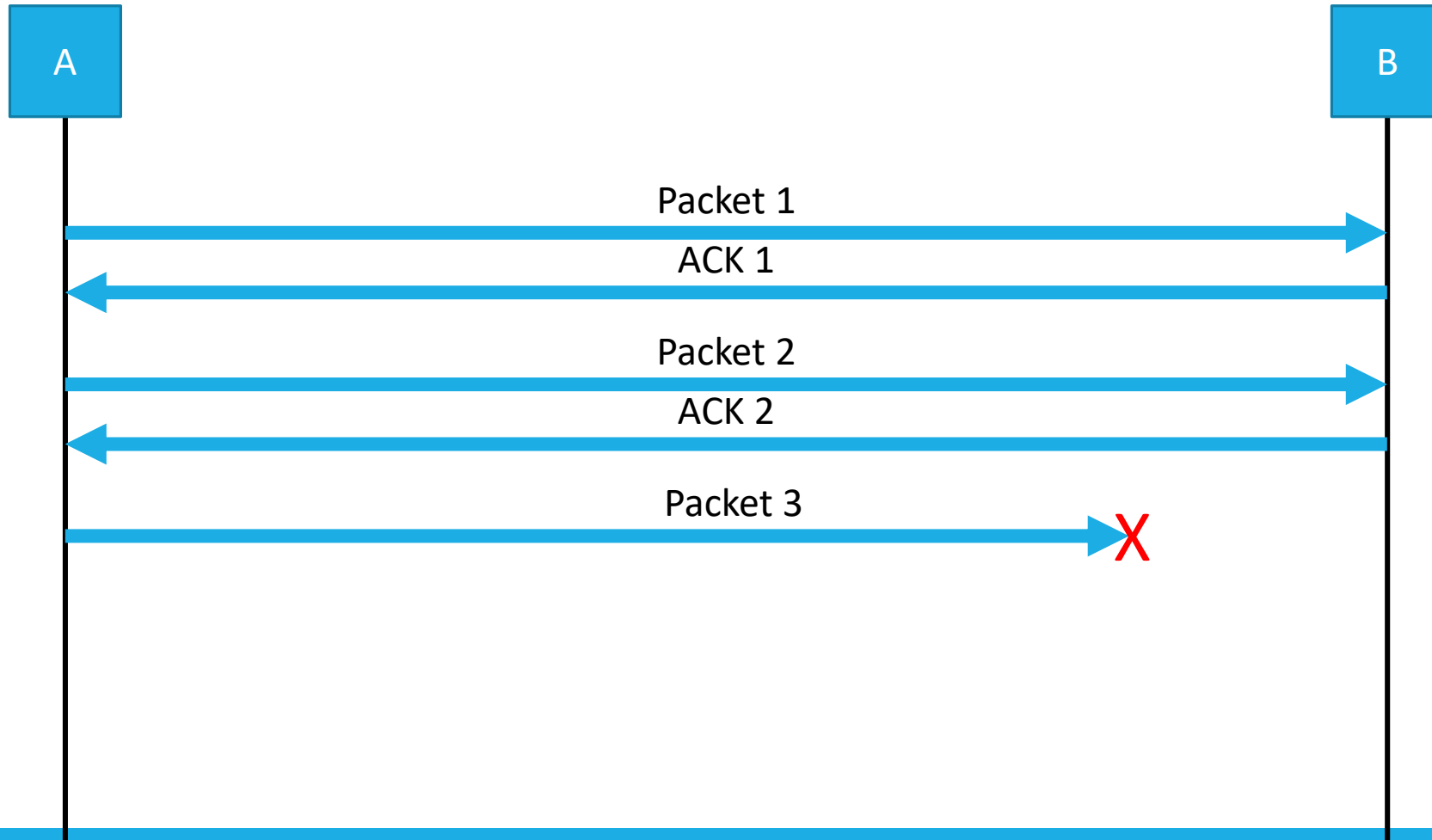
TCP Data transfer



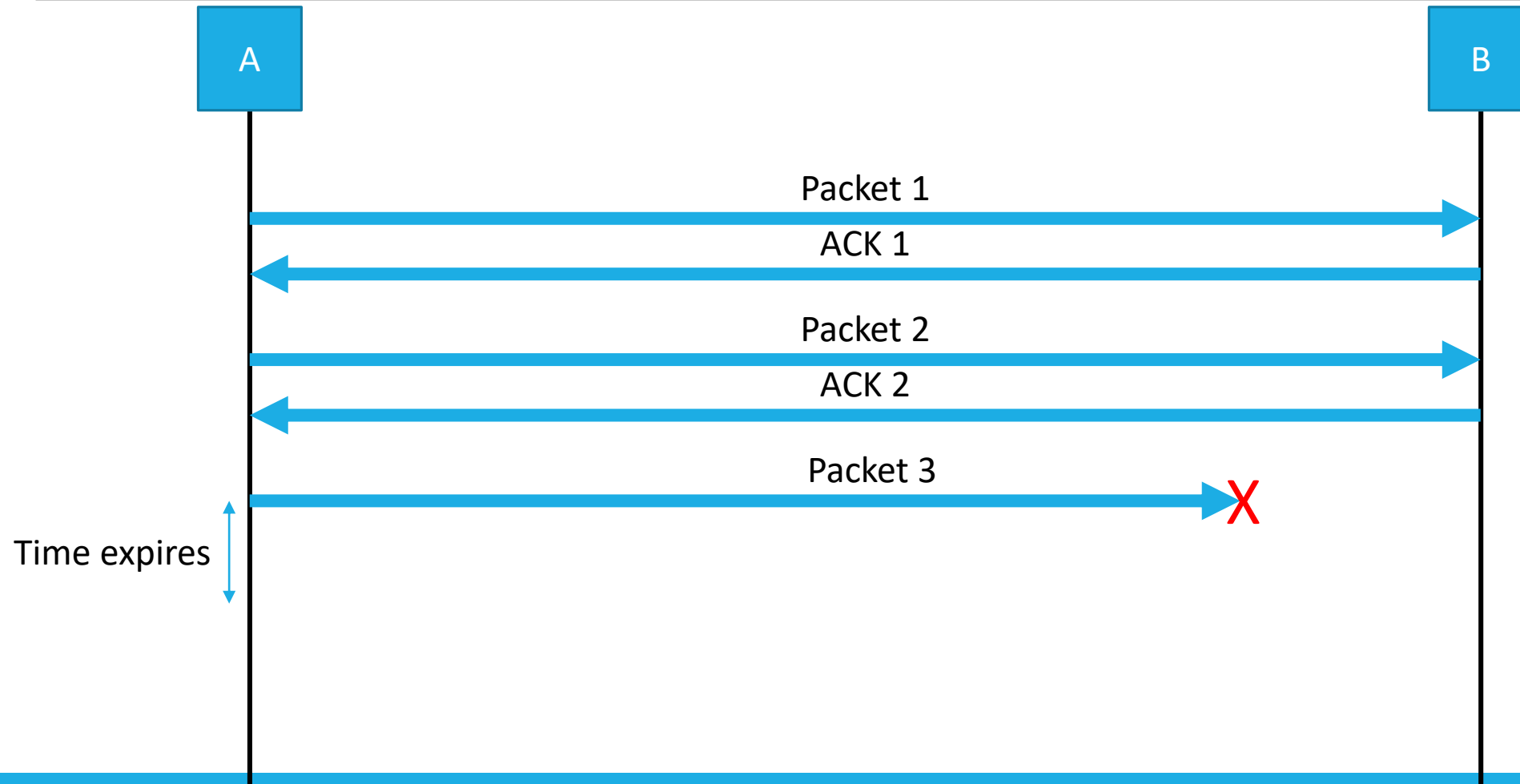
TCP Data transfer



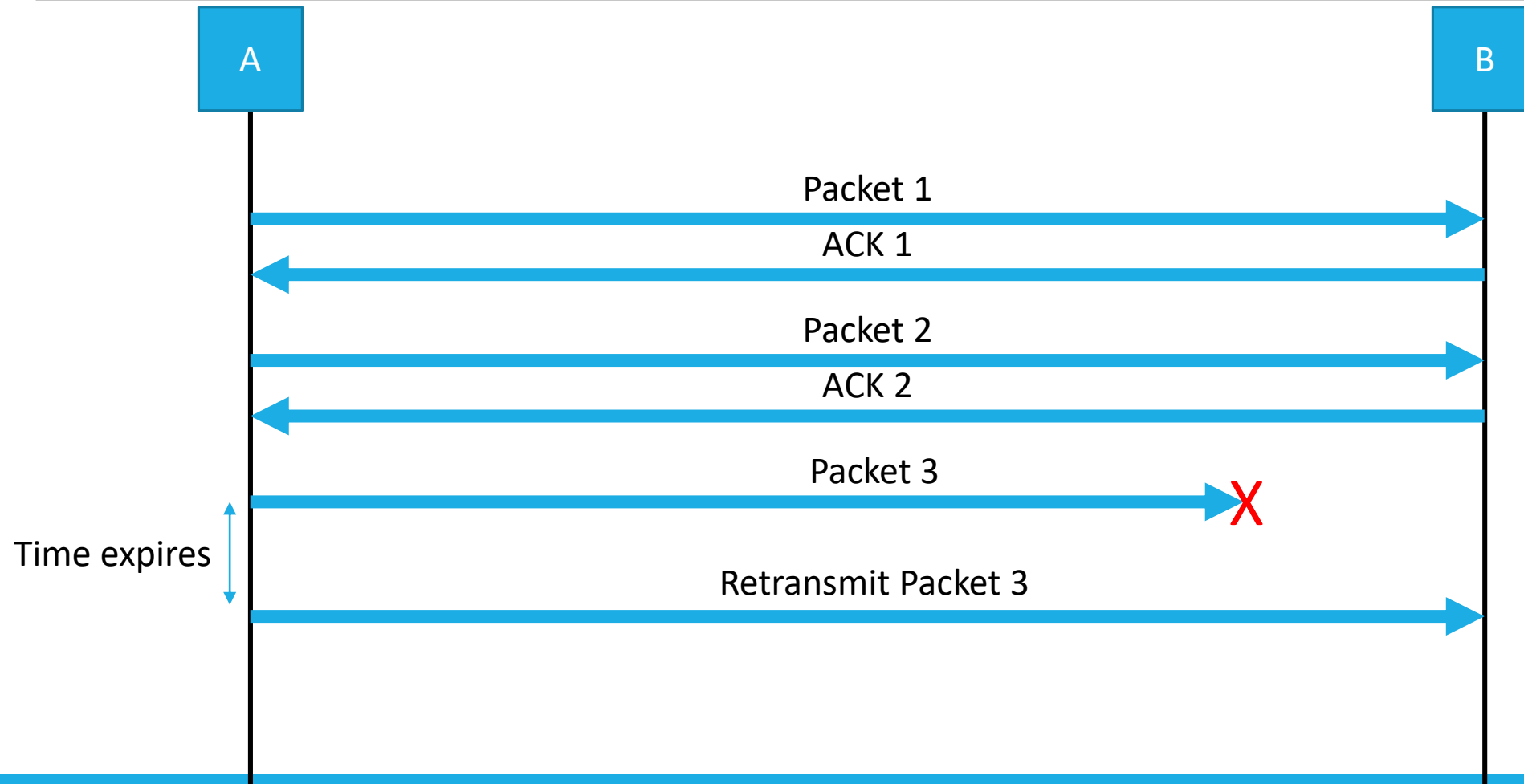
TCP Data transfer



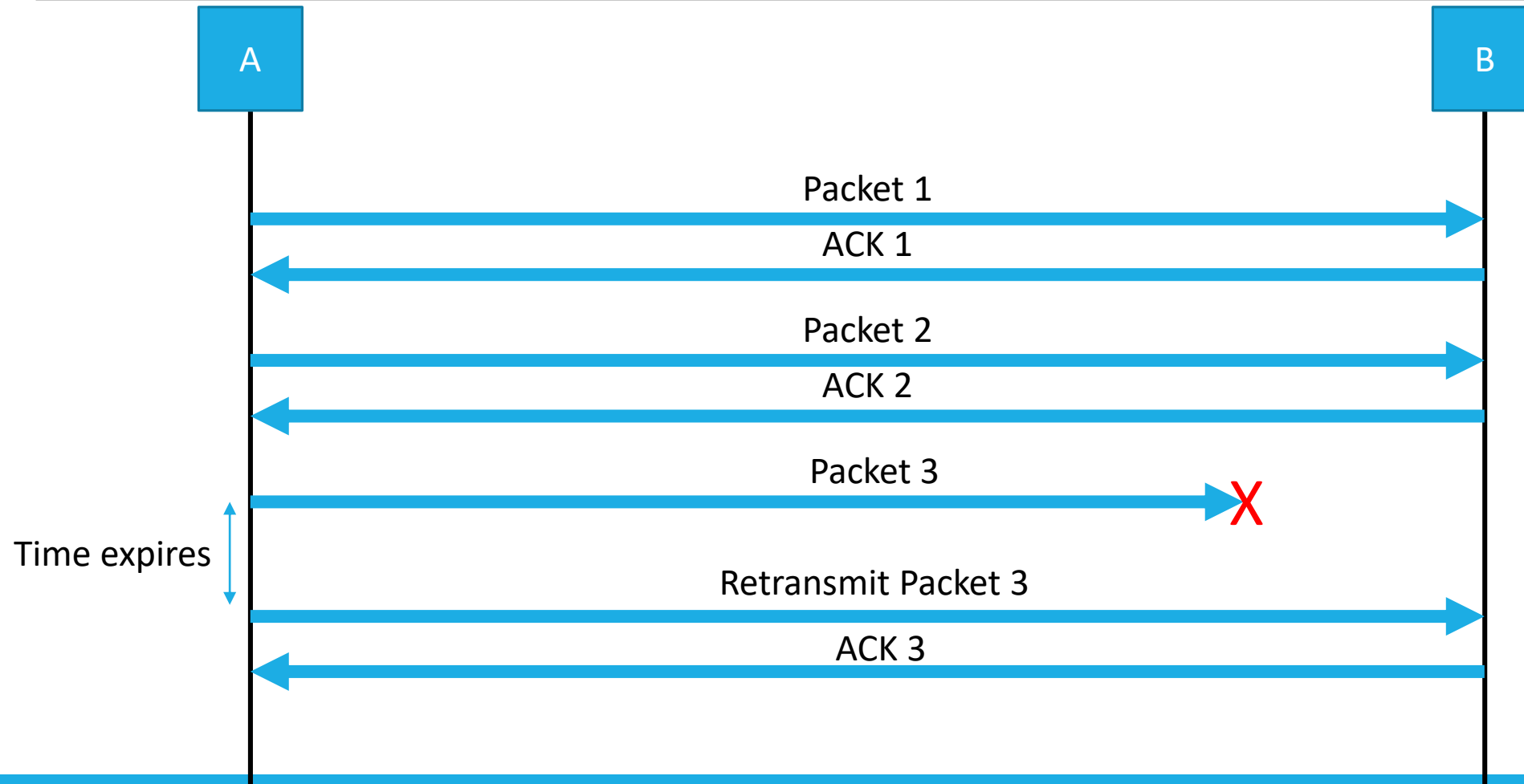
TCP Data transfer



TCP Data transfer



TCP Data transfer



IMPLEMENTACIÓN JAVA

[HTTPS://GITHUB.COM/DOMICIANO/PROGRAMACIONENRED191/TREE/MASTER/SEMANA%203/TCPCLIENTSERVER](https://github.com/DOMICIANO/PROGRAMACIONENRED191/TREE/MASTER/SEMANA%203/TCPCLIENTSERVER)

Ejercicio en clase

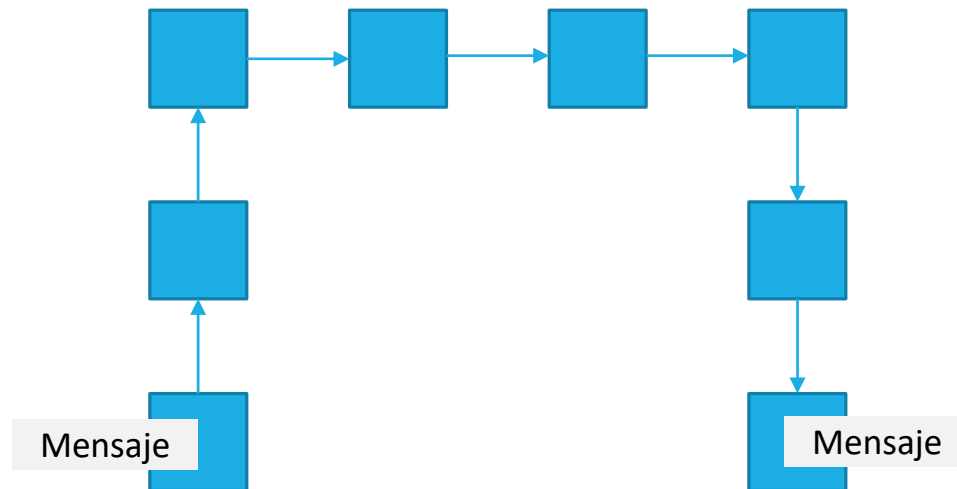
Cronometro en equipo

- Haga equipo con un compañero y comunique, vía TCP, sus dos equipos para cronometrar segundos y minutos.
- Seleccionen a alguien que comience con la cuenta.
- El computador A cronometra un segundo, le avisar al computador B. El computador B sigue la cuenta con otro segundo y cuando termina, éste le avisa al PC A para seguir con la cuenta, así sucesivamente.

Taller grupal

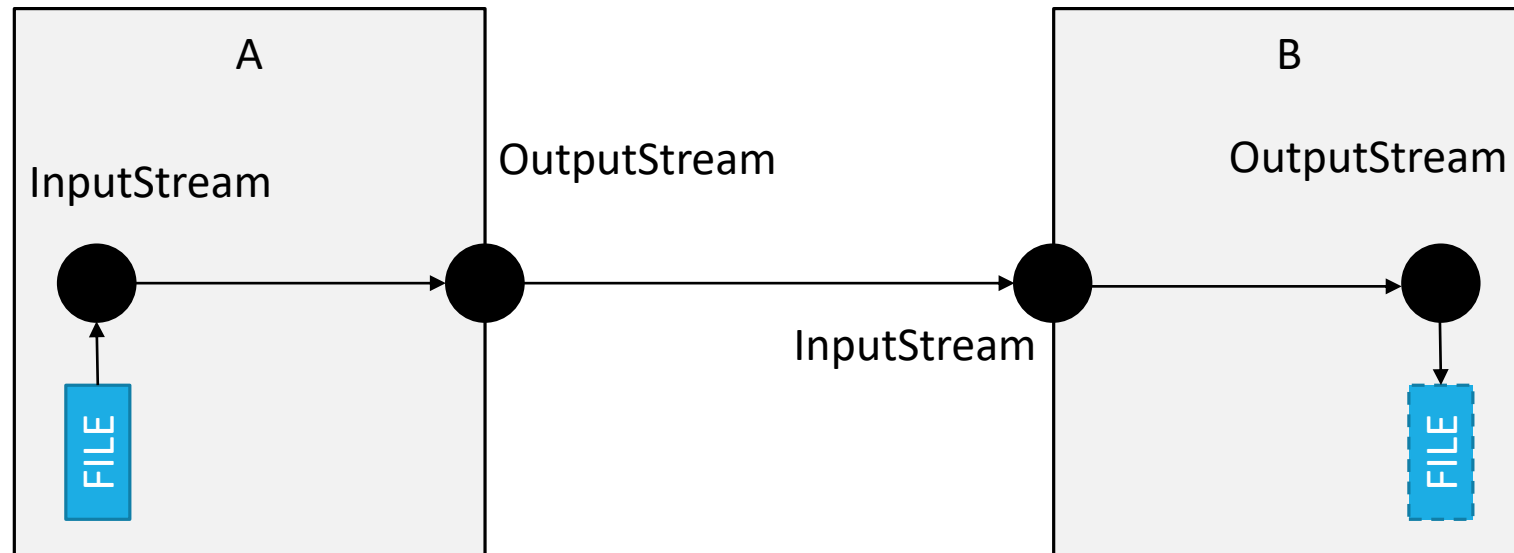
Teléfono roto

Seleccionen un mensaje y transmítanlo de forma que pase por cada uno de los computadores.



Transferencia de archivos

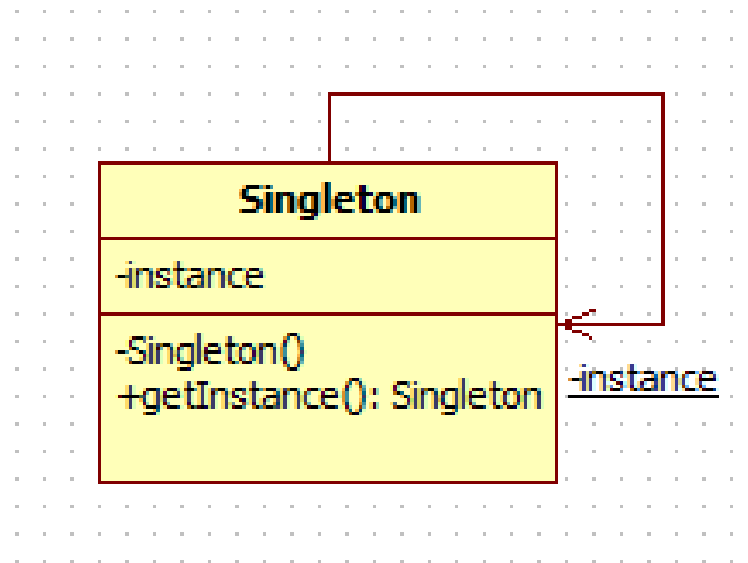
Para la transferencia de archivos necesita



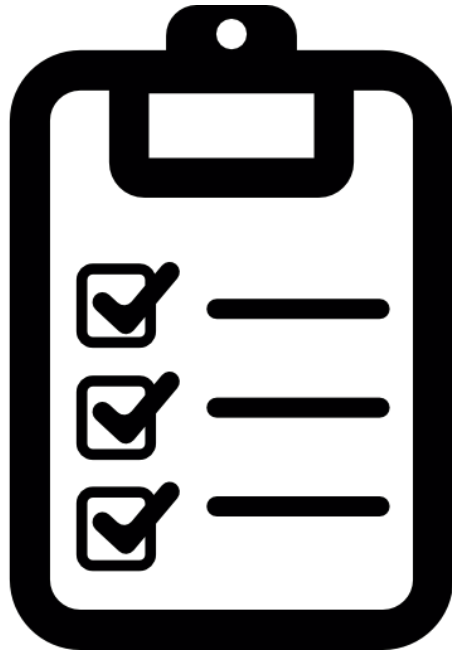
Singleton

El patrón Singleton permite crear una única instancia a partir de una clase.

En el caso de conexiones, el patrón Singleton cobra importancia, permitiendo usar una conexión activa por múltiples objetos dentro de un software



Ejercicio



Implemente la conexión Singleton del cliente.

Pruebe el envío de mensajes usando el servidor Singleton y el cliente Singleton

Haga que el servidor TCP sea capaz de mantenerse disponible a pesar que ya haya terminado una sesión con un cliente.

Conexiones asíncronas

Una conexión asíncrona implica que hay un hilo de **ENVÍO** y otro hilo de **RECEPCIÓN**.

Ambos hilos deben surgir en el momento que ocurre el **HANDSHAKE**.

La clase Singleton de conexión debe poder dirigir y obtener información hacia los dos hilos de recepción y emisión.

El hilo de emisión sólo requiere estar activo mientras envía el mensaje.

El hilo de recepción requiere estar siempre activo.

