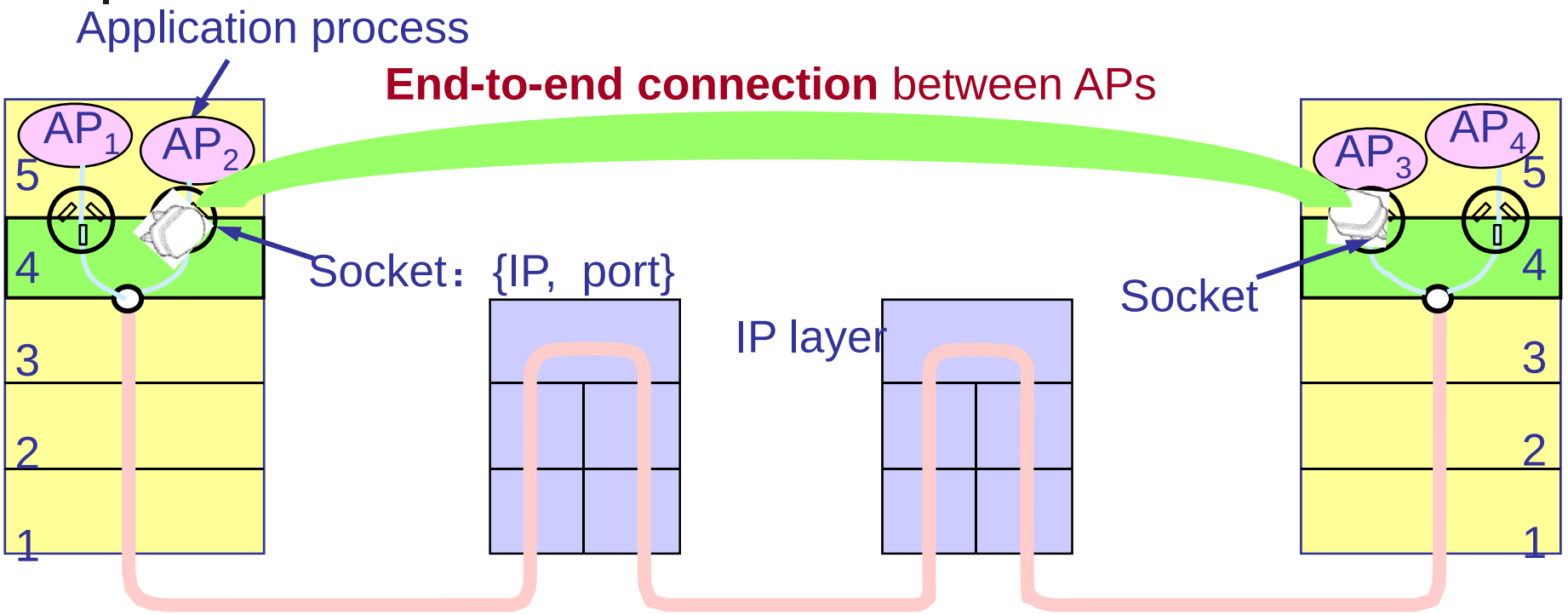
Experiment 1: Content

- **Project**: Socket programming
- **Purpose**: Implement the mutual communication between two hosts based on TCP or UDP.
- **Principle**: Communication procedure of TCP and UDP
- **Requirement**:
  a) Implement one-way data transmission. One sends data, and the other receives data.
  b) Implement Client and Server send and receive data at the same time.
  c) Try to transmit a media file and analyze the features of TCP/UDP
  d) Please submit your experiment report after the experiment class within one week.

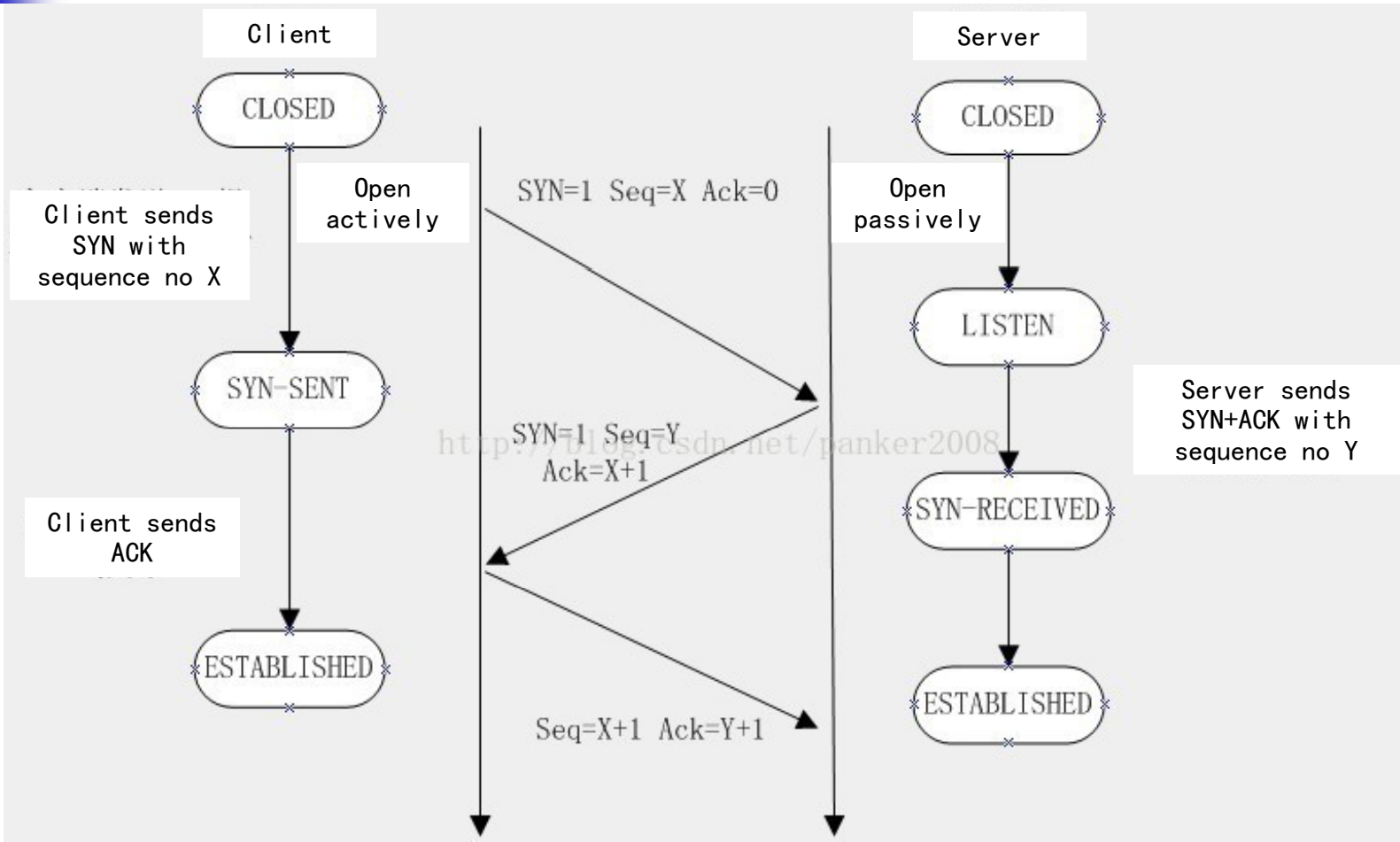
  Submission email address: 253769492@qq.com.
  The report file should be named as stu no. + name + exp no.

What is SOCKET？

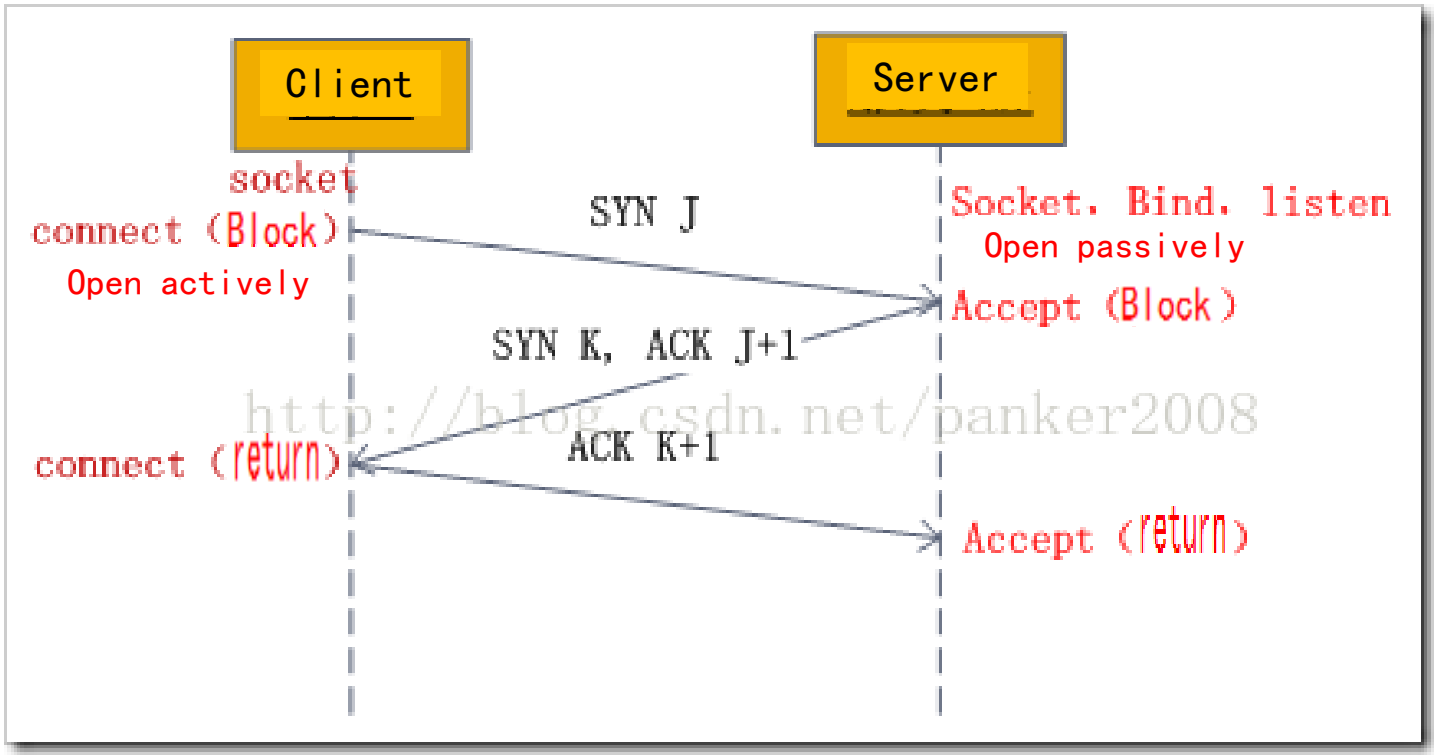


A host can run many sorts of software in the Internet, and provide several services. Each service opens a Socket , which is bound to a port. So different ports are corresponding to different services.

Establish TCP connection by three-way handshake

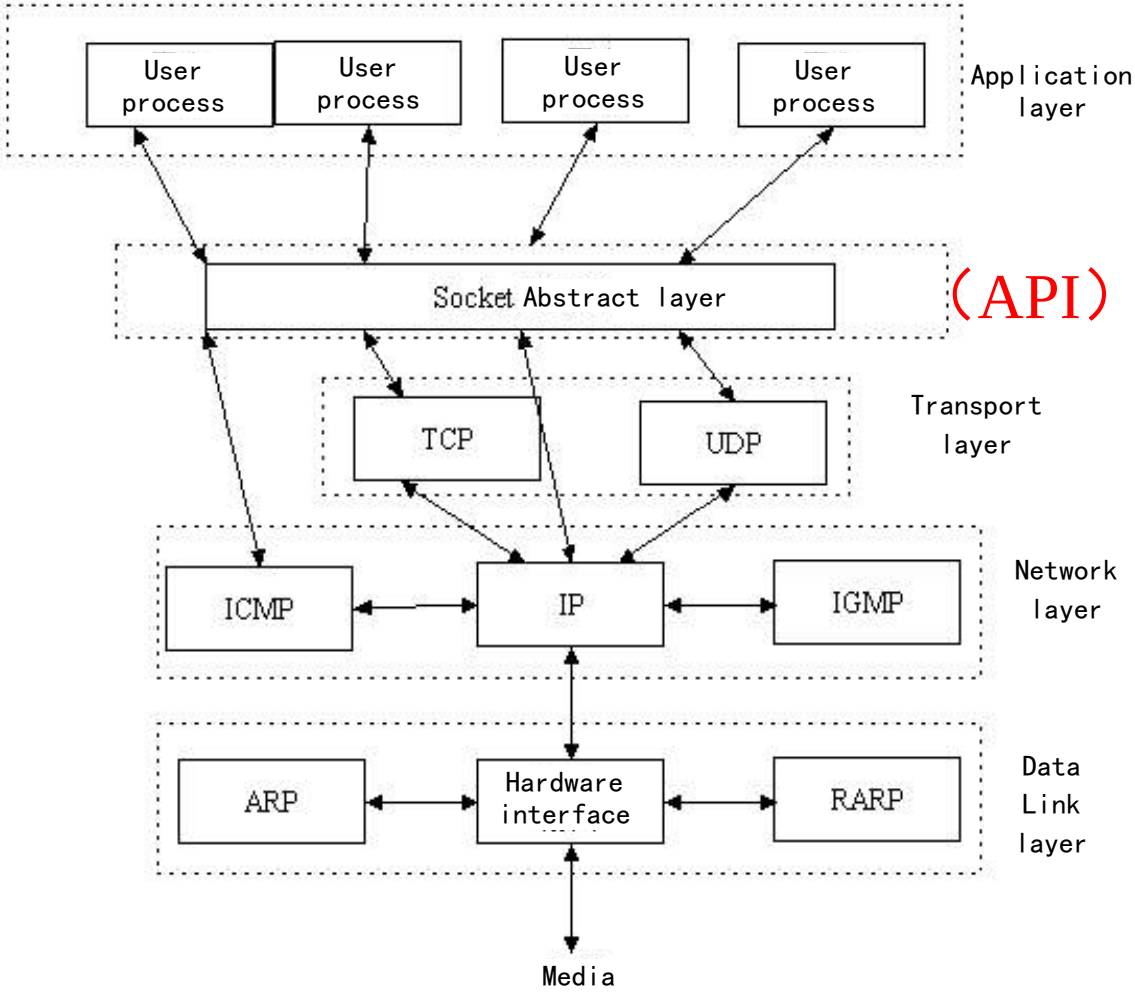


Socket implements three-way handshake

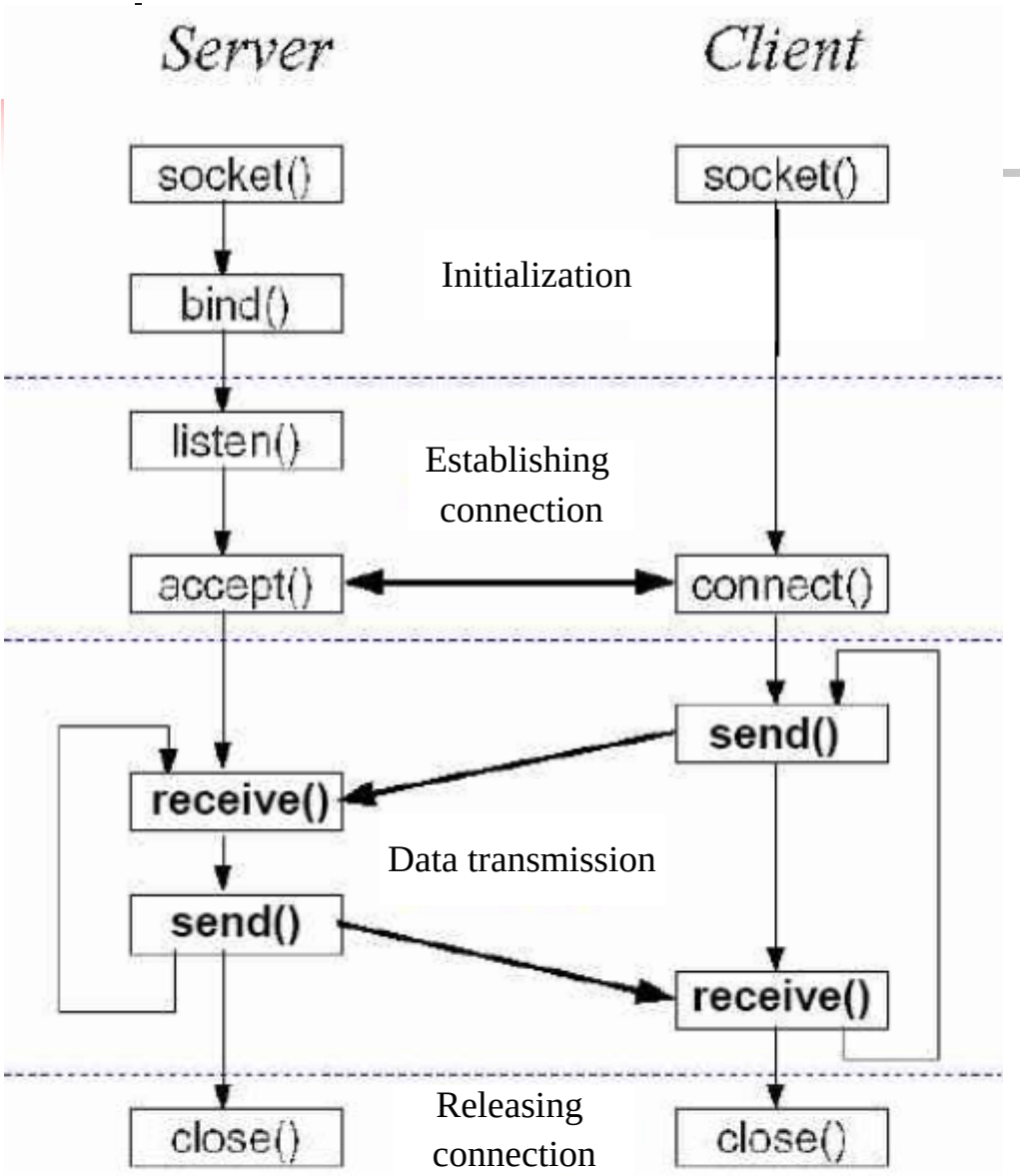


What is SOCKET?

- Socket is abstract layer between the application layer and the transport layer. It can **abstract** the complex operations of **TCP/IP** to some simple **interfaces**, and the application processes can communicate with each other by **calling** the interfaces.



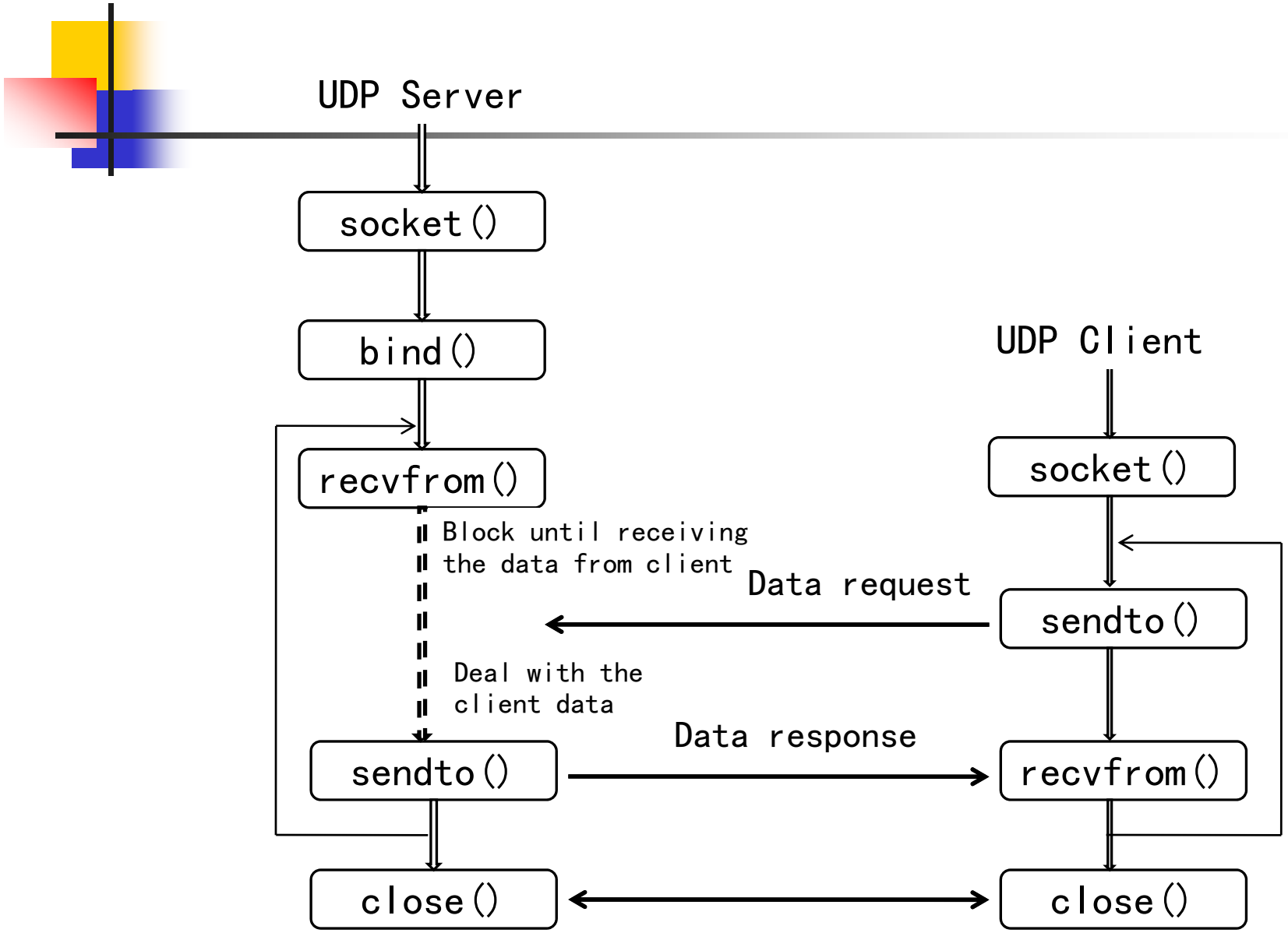
Client/Server program based on TCP

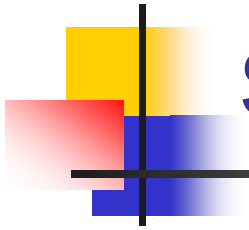


Socket application program

- **Socket()** creates a socket.
- **Bind()** gives an identifier (ip and port) to the socket.
- **Listen()** distributes data space, in order to store connection requests of users.
- **Accept()** waits to receive connection requests of users.
- **Connect()** requires to establish connection.
- **Send()** and **receive()** finish the data transmission in full-duplex.
- **Close()** releases connection..

Client/Server program based on UDP





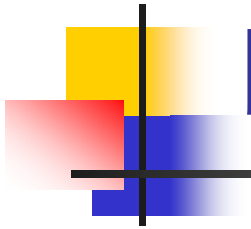
socket()

```
#include <winsock.h>
SOCKET  socket( int af, int type, int protocol);
```

Create socket.

type： describe the type of the socket to be created.

- SOCK_STREAM use TCP
- SOCK_DGRAM use UDP
- int fd = socket(AF_INET, SOCK_STREAM, 0)

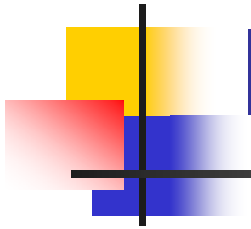


bind()

int bind(SOCKET s, struct sockaddr name, int namelen);

Bind a socket to the ip address and port of a hosts.。

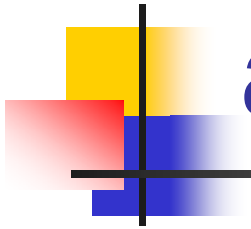
```
struct sockaddr_in {
        short int sin_family; /* Address family */
        unsigned short int sin_port; /* Port number */
        struct in_addr sin_addr; /* Internet address */
        unsigned char sin_zero[8]; /* Same size as struct sockaddr
*/
        };
```



listen()

```
#include <winsock.h>
 int listen( SOCKET s, int backlog);
```

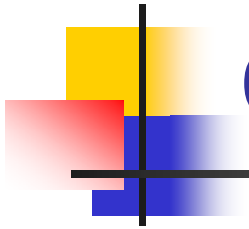
Listen（）is used in the connection-oriented server to listen to the connection of a client.



accept()

```
#include <winsock.h>
SOCKET accept(SOCKET s, struct sockaddr * addr, int * addrlen);
```

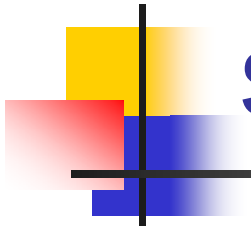
accept() is responsible to fetch a connection from the array of established connection and bind it to s.



connect()

```
#include <winsock.h>
int connect( SOCKET s, struct sockaddr * name, int namelen);
```

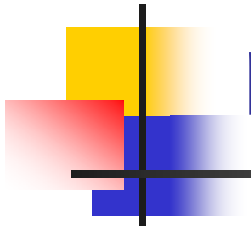
This function is used to create the connection to the specified port of a server.



send()

int send(SOCKET s, char *buf, int len, int flags);

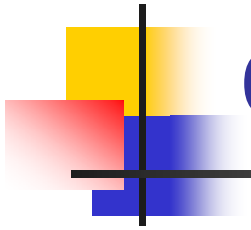
Processes No matter in the client or in the server send the data to the other end of TCP by using send( ).



recv()

int recv(SOCKET s, char FAR *buf, int len, int flags);

Processes No matter in the client or in the server receive the data from the other end of TCP by using recv( ).



closesocket()

BOOL closesocket(SOCKET s);

closesocket() closes the socket s.