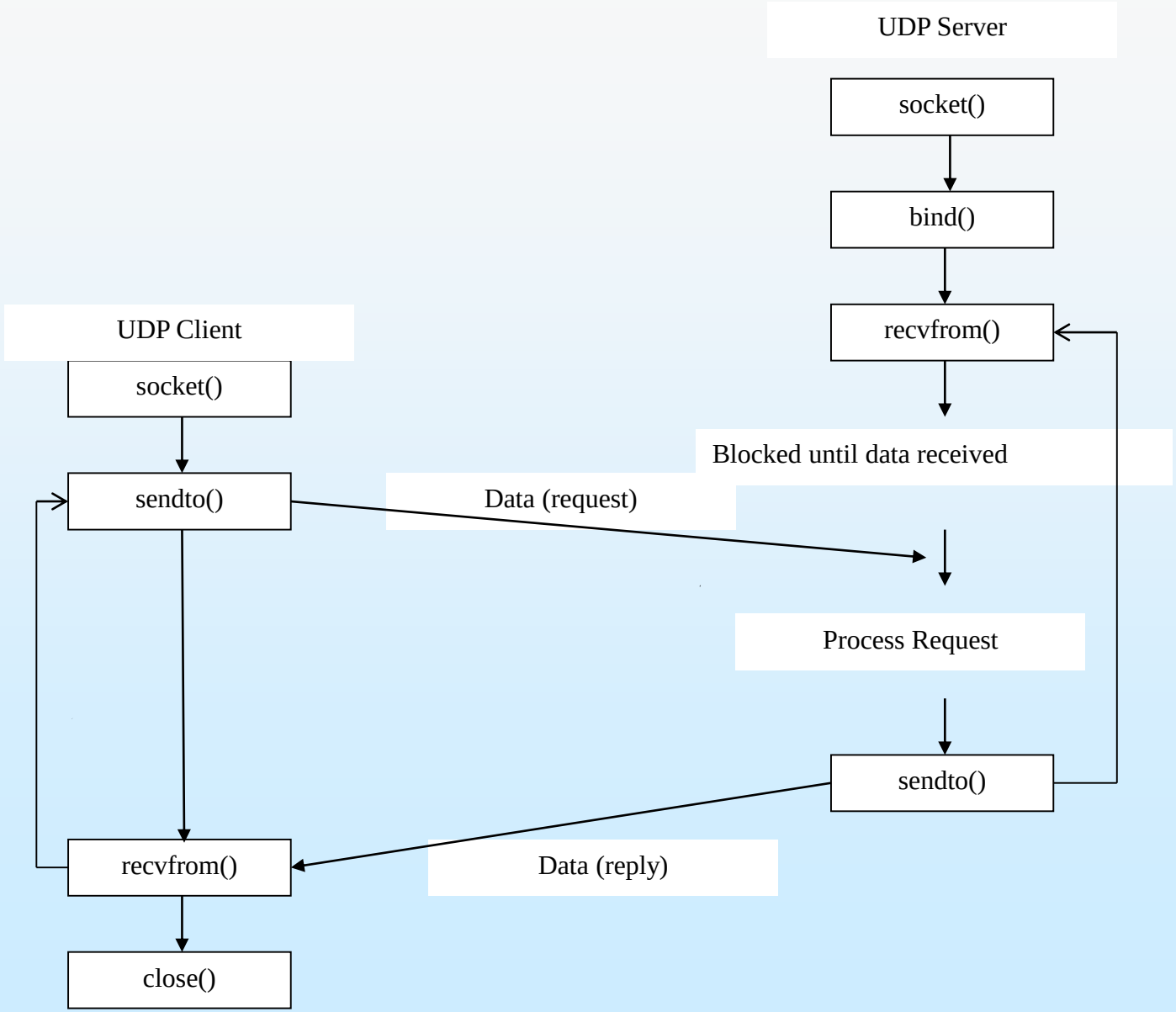
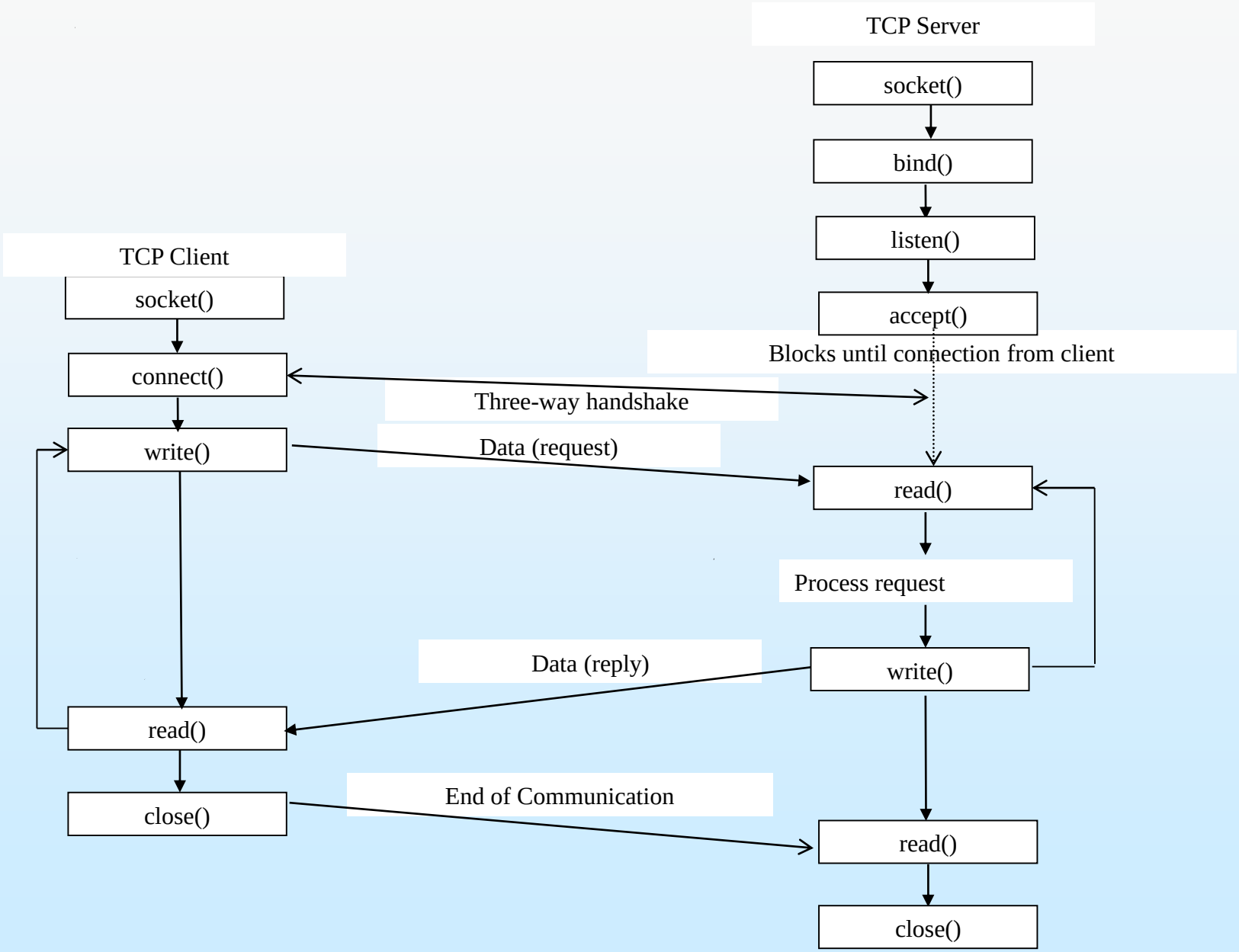
Socket (UDP)


UDP Server
socket()
bind()
recvfrom()
Blocked until data received
Process Request
sendto()
Data (reply)
UDP Client
socket()
sendto()
Data (request)
recvfrom()
close()

TCP Socket


TCP Server
socket()
bind()
listen()
accept()
Blocks until connection from client
TCP Client
socket()
connect()
Three-way handshake
write()
Data (request)
read()
Process request
Data (reply)
write()
read()
close()
End of Communication
read()
close()

UDP Echo Client

```
main()
{
    int sockfd,returnvalue,len=100,n;        unsigned int addrlen;   unsigned short serv_port=25000;
    char serv_ip[]="192.168.3.1";            char text[]="abcde";                char rtext[100];
    struct sockaddr_in servaddr;             addrlen=sizeof(servaddr);
    bzero(&servaddr,sizeof(servaddr));       servaddr.sin_family=AF_INET;
    servaddr.sin_port=htons(serv_port);      inet_aton(serv_ip,&servaddr.sin_addr);
    sockfd=socket(PF_INET,SOCK_DGRAM,0);
returnvalue=sendto(sockfd,text,strlen(text),0,(struct sockaddr *) & servaddr,sizeof(servaddr));
    if(returnvalue!=-1)
            printf("\n\t\t\t*** message sent successfully ***\n");
    else
            printf("\n\t\t\t*** message sent failure ***\n");
n=recvfrom(sockfd,rtext,len,0,(struct sockaddr *) &servaddr,&addrlen);
if(n!=-1)
    {
    rtext[n]='\0';        printf("\n\t\t\t*** %s ***\n",rtext);
    }
    close(sockfd);
}
```

UDP Echo Server

```
main()
{
short int n=0,m=0;  int len=100;          int addrlen;      unsigned short serv_port=25000;
short sockfd;         char serv_ip[]="192.168.3.1";char mesg[1000];    char *client;
struct sockaddr_in servaddr, clientaddr;              client=(char *)malloc(sizeof(servaddr));
sockfd=socket(AF_INET, SOCK_DGRAM, 0);
if(sockfd<0)          {          printf("failed");      exit(1);    }
bzero(&servaddr, sizeof(servaddr));                   servaddr.sin_family=AF_INET;
inet_aton(serv_ip,&servaddr.sin_addr);                servaddr.sin_port=htons(serv_port);
bind(sockfd, (struct sockaddr *) &servaddr, sizeof(servaddr));
n=recvfrom(sockfd,mesg,len,0,(struct sockaddr *) & clientaddr,&addrlen);
if(n!=-1)             printf("\n\t\t\t*** receive success ***\n");
else                  printf("\n\t\t\t*** receive failure ***\n");
m=sendto(sockfd,mesg,n,0,(struct sockaddr *) & clientaddr,sizeof(clientaddr));
if (m!=-1)                    printf("\n\t\t\t*** send success ***\n");
else                          printf("\n\t\t\t*** send failure ***\n");
close(sockfd);
}
```

TCP Echo Client

```
main()
{
      int sockfd,returnvalue,len=100,n;          unsigned int addrlen;
      unsigned short serv_port=25000;            char serv_ip[]="192.168.3.1";
      char text[]="abcde";              char rtext[100];
      struct sockaddr_in servaddr;  addrlen=sizeof(servaddr);
      bzero(&servaddr,sizeof(servaddr));
      servaddr.sin_family=AF_INET;
      servaddr.sin_port=htons(serv_port);
      inet_aton(serv_ip,&servaddr.sin_addr);
      sockfd=socket(PF_INET,SOCK_STREAM,0);
      connect(sockfd,(struct sockaddr *) & servaddr,sizeof(servaddr));
returnvalue=write(sockfd,text,strlen(text));
      if(returnvalue!=-1)                        printf("\n* sent successfully *\n");
      else                                       printf("\n* sent failure *\n");
n=read(sockfd,rtext,len);
      rtext[n]='\0';        printf("\n* %s *\n",rtext);
      close(sockfd);
}
```

TCP Echo Server

```
main(){
short int n=0,m=0;      int len=100;            int addrlen;            unsigned short serv_port=25000;
short sockfd,connfd;    char serv_ip[]="192.168.3.1";    char mesg[1000];    char *client_add_dotted;
short client_port;      struct sockaddr_in servaddr, clientaddr;
client_add_dotted=(char *)malloc(sizeof(servaddr));
sockfd=socket(AF_INET, SOCK_STREAM, 0);
if(sockfd<0)            {           printf("failed");       exit(1);    }
bzero(&servaddr, sizeof(servaddr));         servaddr.sin_family=AF_INET;
inet_aton(serv_ip,&servaddr.sin_addr);      servaddr.sin_port=htons(serv_port);
bind(sockfd, (struct sockaddr *) &servaddr, sizeof(servaddr));
listen(sockfd,1);
connfd=accept(sockfd,(struct sockaddr *) & clientaddr,&addrlen);
n=read(connfd,mesg,100);
if(n!=-1)               printf("\n\t\t\t***receive success***\n");
else        printf("\n\t\t\t***receive failure***\n");
m=write(connfd,mesg,n);
if (m!=-1)              printf("\n\t\t\t***send success***\n");
else                    printf("\n\t\t\t***send failure***\n");
close(sockfd);}
```