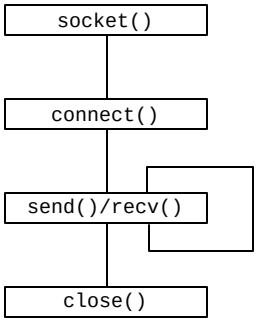
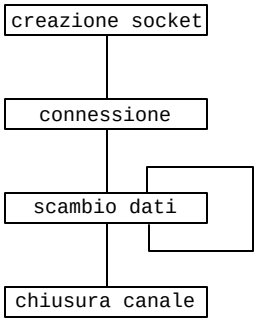
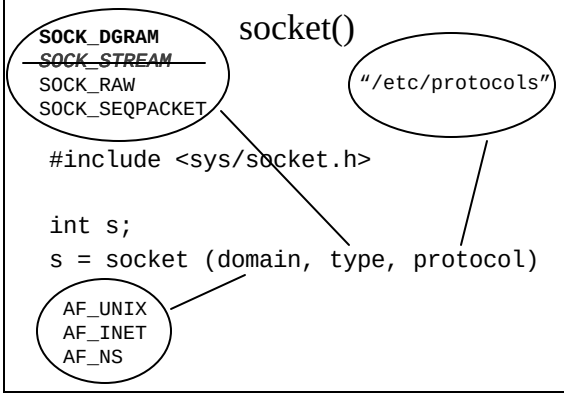
Socket UDP

La differnza tra TCP e UDP è che il secondo non implementa strutture per il controllo di flusso

Ogni singolo blocco di dati viene spedito come un pacchetto indipendente, se viene perso il sistema non si preoccupa di ritrasmetterlo







socket()

```
int s;
s = socket(AF_INET, SOCK_DGRAM, 0);
If ( s < 0 ) {
     perror("socket() ");
     exit(1);
}
```

send()

```
#include <sys/types.h>
#include <sys/socket.h>

int size;
size = send(socket, buffer, len, flags)
```

0
MSG_EOF

socket()

connect()

```
#include <sys/socket.h>

int error;
error = connect(socket, addr, len);
```

struct sockaddr_in
struct sockaddr_un

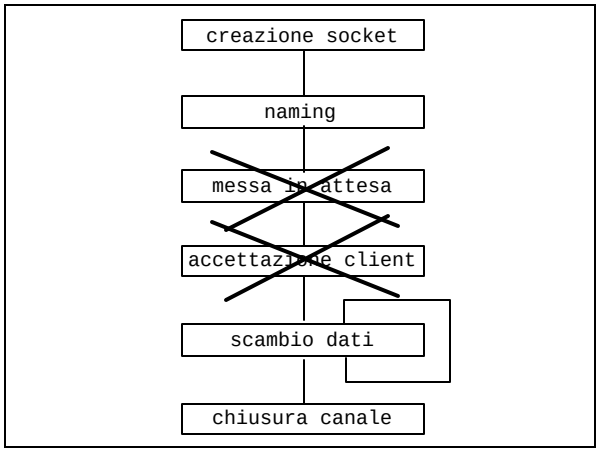
sizeof(addr)

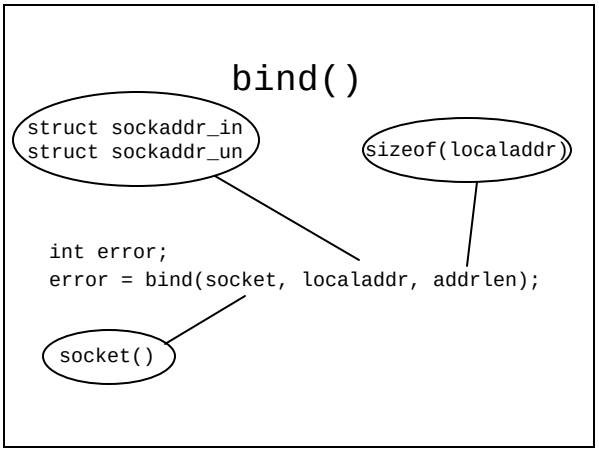
socket()

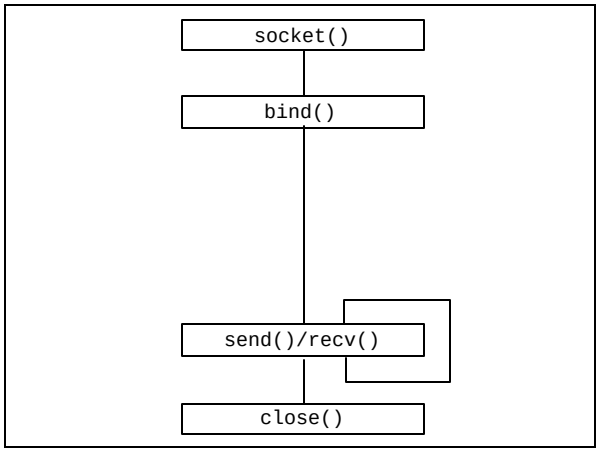
recv()

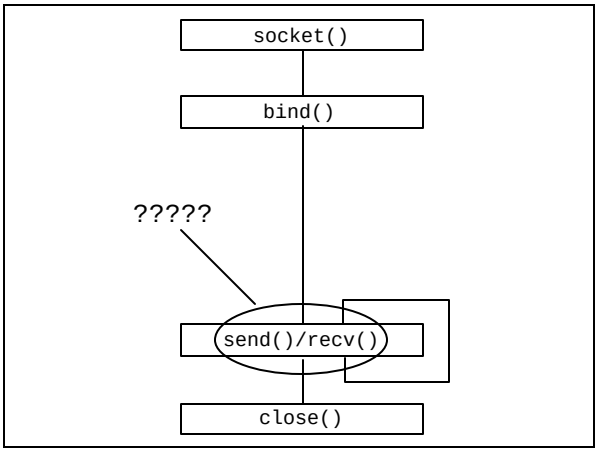
```
#include <sys/types.h>
#include <sys/socket.h>

int size;
size = recv(socket, buffer, len, flags)
```

0
MSG_PEEK
MSG_WAITALL


creazione socket
naming
messa in attesa
accettazione client
scambio dati
chiusura canale


bind()
struct sockaddr_in
struct sockaddr_un
sizeof(localaddr)
int error;
error = bind(socket, localaddr, addrlen);
socket()


socket()
bind()
send()/recv()
close()

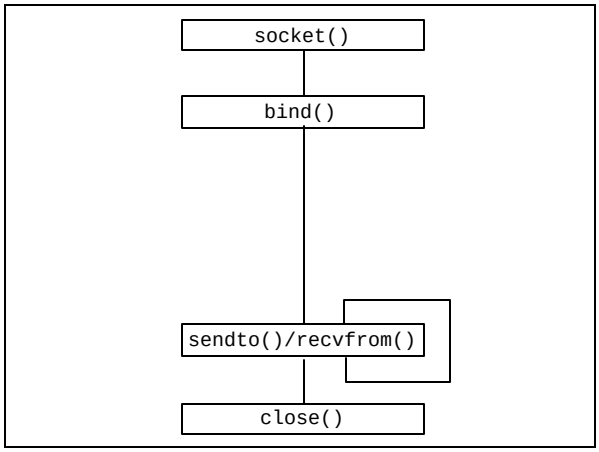

socket()
bind()
?????
send()/recv()
close()

Connectionless !

Non avendo fatto la `connect` la socket locale NON possiede informazioni sul peer remoto !

Questa cosa può essere risolta a livello applicazione, dall'altra parte anche il client deve fare una `bind` e darci la possibilità di mandare indietro i dati (comunicandoci la porta)

NOTA: lo schema appena visto NON va bene!



Client/Server completamente connectionless

Perchè fare la `connect` (o `accept`) se puoi non abbiamo controllo di flusso ?

È possibile usare la socket in maniera connectionless se facciamo uso di altre primitive (specializzate) per inviare/ricevere messaggi a/da indirizzi arbitrari

`sendto()`

```
#include <sys/types.h>
#include <sys/socket.h>

int size;
size = sendto(socket, buffer, len, flags,
          to_addr, addrlen);
```

`struct sockaddr_in`
`struct sockaddr_un`

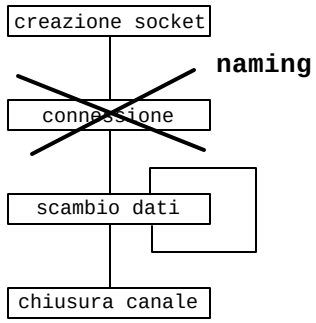
`sizeof(to_addr)`

recvfrom()

```
#include <sys/types.h>
#include <sys/socket.h>

int size;
size = recvfrom(socket, buffer, len, flags,
            from_addr, addrlen);
```

struct sockaddr_in
struct sockaddr_un

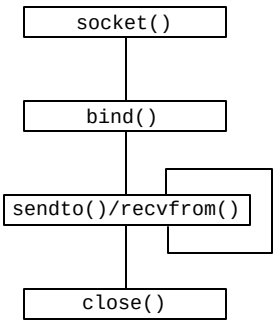
sizeof(from_addr)



Client/Server completamente connectionless

Anche da parte del client possiamo fare a meno della `connect`

Però questo ci costringe ad usare una `bind`, altrimenti il sistema non allocherà una porta alla socket locale ed il server non riuscirà comunque a rispondere



Client o Server ?

Lo schema ottenuto risulta identico allo schema che abbiamo visto parlando del server

Siamo di fronte ad una struttura completamente simmetrica