

the domain **AF\_INET** is used. The next field type has the value **SOCK\_DGRAM**. It supports datagrams (connectionless, unreliable messages of a fixed maximum length). The protocol field specifies the protocol used. We always use 0. If the socket function call is successful, a socket descriptor is returned. Otherwise -1 is returned. The header files necessary for this function call are `sys/types.h` and `sys/socket.h`.

## 2. Filling the fields of the server address structure.

The socket address structure is of type `struct sockaddr_in`.

```
struct sockaddr_in {  
  
    u_short sin_family;  
    u_short sin_port;  
    struct in_addr sin_addr;  
    char sin_zero[8]; /*unused, always zero*/  
};  
struct in_addr {  
  
    u_long s_addr;  
  
};
```

The fields of the socket address structure are

**sin\_family** which in our case is **AF\_INET**

**sin\_port** which is the port number where socket binds

**sin\_addr** is used to store the IP address of the server machine and is of type `struct in_addr`

The header file that is to be used is **netinet/in.h**

The value for `servaddr.sin_addr` is assigned using the following function

```
inet_pton(AF_INET, "IP_Address", &servaddr.sin_addr);
```

The binary value of the dotted decimal IP address is stored in the field when the function returns.

## 3. Binding of a port to the socket in the case of server