

## Feature:

-----

1. Server is IP Protocol independent, it can handle both IPv4 and IPv6 addresses.
2. Client may open a socket with 'ai\_family' = AF\_INET or AF\_INET6.

## Technical:

-----

1. Client may open a socket with 'ai\_family' = AF\_INET/AF\_INET6.
2. Client uses getaddrinfo to get the address family information depending on the ipv4 or ipv6 address given as input.
3. Server opens a IPv6 socket with address family 'in6addr\_any'.
4. Server will listen to any IP address and stores the client information in 'sockaddr\_storage'.

## Reference:

-----

<http://man7.org/linux/man-pages/man2/socket.2.html>  
<http://man7.org/linux/man-pages/man3/getaddrinfo.3.html>  
<http://man7.org/linux/man-pages/man3/getnameinfo.3.html>  
<http://pubs.opengroup.org/onlinepubs/009696699/basedefs/sys/socket.h.html>  
[http://www.ibm.com/support/knowledgecenter/ssw\\_i5\\_54/rzab6/xacceptboth.htm](http://www.ibm.com/support/knowledgecenter/ssw_i5_54/rzab6/xacceptboth.htm)

## Screenshots:

-----

Server accepts both Ipv4 and Ipv6 clients.

```
nichail@n:~/CS16MTECH11009_Submission_Assignment2/Part_C_Protocol_Independent$ gcc server.c -o server
nichail@n:~/CS16MTECH11009_Submission_Assignment2/Part_C_Protocol_Independent$ ./server 2222
Handling Client: [::1], port: [38999]
Handling Client: [::ffff:127.0.0.1], port: [60821]
```

---

```
nichail@n:~/CS16MTECH11009_Submission_Assignment2/Part_C_Protocol_Independent$ gcc client.c -o client
nichail@n:~/CS16MTECH11009_Submission_Assignment2/Part_C_Protocol_Independent$ ./client ::1 2222
Type string: Hello Server, I am IPv6 Client
Received: Hello Server, I am IPv6 Client
Type string:
```

---

```
nichail@n:~/CS16MTECH11009_Submission_Assignment2/Part_C_Protocol_Independent$ ./client 127.0.0.1 2222
Type string: Hello Server, I am IPv4 Client
Received: Hello Server, I am IPv4 Client
Type string: █
```