Simple Datagram Server & Client:

- listener.c & talker.c

- **listener** sits on a machine waiting for an incoming packet on port 4950.

- **talker** sends a packet to that port, on the specified m achine, that contains whatever the user enters on the command line.

- **The server (listener.c) code (skeleton):**

- #include .....

  #define MYPORT 4950   /* the port
        users will be connecting to */

- #define MAXBUFLEN 100

- main()
- {
- int **sockfd**;
- struct sockaddr_in **my_addr**;
        /* my address information */
- struct sockaddr_in   **their_addr**;
   /* connector's address information */
- int **addr_len**, **numbytes**;
- char **buf**[MAXBUFLEN];

- **sockfd** = **socket**(AF_INET,
                SOCK_DGRAM, 0) ;

```c
my_addr.sin_family = AF_INET;

my_addr.sin_port = MYPORT;

my_addr.sin_addr.s_addr =
                  INADDR_ANY;
    /* auto-fill with my IP */

bind(sockfd, &my_addr,
        sizeof(struct sockaddr));

numbytes = recvfrom(sockfd, buf,
        MAXBUFLEN, 0,
        &their_addr,   &addr_len);

printf("got packet from %s\n",
                their_addr.sin_addr);
printf("packet is %d bytes long\n",
                        numbytes);
buf[numbytes] = '\0';
printf("packet contains "%s", buf);
```

- **close**(**sockfd**);
- } /* End Datagram Server */

The Client (talker.c) code (skeleton):

- #include ....

- #define ServPORT 4950
- /* the port clients will be connecting to */

- **main**(int argc, char *argv[])
- {
- int **sockfd**;
- struct sockaddr_in **their_addr**;
- /* Server's address information */
- struct hostent ***he**;
- int **numbytes**;

- **he** = **gethostbyname**(argv[1]))

- **sockfd** = **socket**(AF_INET,
                    SOCK_DGRAM, 0) ;

```c
their_addr.sin_family = AF_INET;

their_addr.sin_port = ServPORT;

their_addr.sin_addr = he->h_addr;

numbytes = sendto(sockfd, argv[2],
 strlen(argv[2]), 0, &their_addr,
        sizeof(struct sockaddr));

printf("sent %d bytes to %s\n",
 numbytes, their_addr.sin_addr);

close(sockfd);

return 0;
}/* End Datagram Client */
```