UDP Broadcasts

Client: broadcasts a request on a subnet/cable.

Servers: responds to any broadcast request.

Note: a broadcast is to all machines, not to all ports.
A broadcast to the port 5432 can only be received by server programs running on port 5432.
Alternate wording: single port, multiple IP addresses.

Note: the server's response will provide the client with the IP address of the server.

Uses:
–Find a server
–Find all servers

Used by:
–Network Information Services (yp) (network accounts)
–Microsoft browser protocol (list shares)

Broadcasts–Server Side

A server is running on a "port".
It can receive any broadcast to that port.
By default, the receipt of broadcasts on a port is disabled. Any server wanting to respond to broadcasts enable receipt of broadcasts on a per port basis.

`setsockopt`
–Sets the options for a socket.

5 parameters

`int s` – the socket descriptor
`int level` – what level of the IP protocol stack this option modifies
`int opname` – which option is being set
`const void *optval` – the option is set to this value. The structure of the thing pointed to depends on the `opname`
`socklen_t optlen` – `sizeof(optval)`

1 return value
negative – Failed
other – Success

setsockopt

```
int s = passiveUDP(service);
int One = 1;
setsockopt(s,SOL_SOCKET,SO_BROADCAST,
    &One,sizeof(One));
```

`s` a socket.

`SOL_SOCKET`–the option being set applies to this socket.

`SO_BROADCAST`–the option being set is the ability to receive broadcasts. This options takes a "boolean" parameter, true enables the receipt of broadcasts, false disables it

`&One`–The option takes a boolean (`int`), to match the `void *` this is the address of a `int`.

`sizeof(One)`–as with many untyped parameters, you must say how large the parameter is.

For the server a one line addition enables the receipt of broadcasts.

The return value of `setsockopt` should be checked and an error exit done on < `0`.

.

Broadcasts–Client Side

Two things must be done to the client side.
1) the sending of broadcasts must be enabled
2) the broadcast address must be used in the sendto.

Setup steps:
1) get a socket
2) enable sending of broadcasts
3) send the broadcast, (receive replies)

Running steps:
Send a broadcast, i.e., send to the broadcast address.
Receive all replies. This probably means a loop with a select, and the select may have a timeout.

Simple Broadcast Client

```
int main(){
  time_t  now;
  int s;
  int alen = sizeof(struct sockaddr_in);
  int One = 1;
  struct sockaddr_in fsin, fsin_him;
  /* get socket */
  s = socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP);
  /* enable broadcast */
  setsockopt(s,SOL_SOCKET,SO_BROADCAST,&One,
    sizeof(One));
  fsin.sin_family = AF_INET;
  fsin.sin_port = htons(5432);
  fsin.sin_addr.s_addr = INADDR_BROADCAST;
  /* send broadcast */
  sendto(s," ",1,0,
    (struct sockaddr *)&fsin, sizeof(fsin));
  /* get first reply */
  recvfrom(s, &now, sizeof(now), 0,
    (struct sockaddr *)&fsin_him, &alen);
  return 0;
};
```

Robust Client

Need to check the return values of all system calls and handle (error exit) any problems.

Handle multiple replies or no replies (select with bailout)