Internet Multicasting

Somewhere between unicast and broadcast.
Goes to a group of machines.
Machines can register/de-register for the group.

Sender: this is the source of the multicast; it sort of looks like a client in the other models in that it initiates messages.

The send can be done from any socket; binding to a port number is not necessary.

Often the sender only sends, never receives.

Receiver: this is the receiver of the multicast; it sort of looks like a server in other models.

The receiver must bind to a particular port.

Often the receiver never sends messages.

Ports apply to multicast messages.

A multicast to port 5001 will not be heard by a receiver bound to port 5002.

Ethernet Hardware support:
1/2 the hardware addresses are multicast,
(low-bit of high-byte == 1).
A card recognizes, MAC address, broadcast address.
It can be programmed to recognize
one or more multicast addresses.

Internet support:
Internet addresses `224.0.0.0`–`239.255.255.255`
are reserved for multicast groups.
These address all start with the 4 bits 1110.
The low 28 bits are the multicast group number.

Some multicast addresses are reserved,
a few of these are:

`224.0.0.2` — all routers on this subnet.
`224.0.0.5` — OSPF routers.
`224.0.0.9` — RIP2 routers.
`224.0.0.84` — Jini Annoucement.

Multicast Groups

A host can join/leave a multicast group.

A host can be a member of an arbitrary number of groups.

Multicast is one-to-many so:
ICMP is not allowed with multicast groups: eg.
packets time-out, but this only means for some hosts.

Multicast and Routers

Packets that are multicast are "seen" by routers.
If the router has multicast enabled,
it forwards the packet
Implication: hosts do not need routing for multicast If the router has multicast enabled,

Range: if the hosts limits the time-to-live, the range of the multicast packet is limited.

Semantic: hosts join groups on specific networks (cables)
Effect: if you have multiple ethernet cards and
the multicast comes in on an unexpected cable
you won't see it.
Use `INADDR_ANY` to join on all cables.

Sending a Multicast (simple)

1) Get a socket, must be UDP since it isn't one to one.

2) Set up a internet address structure `in_addr` with the multicast address.

3) `sendto` the multicast address

```
int s;
struct sockaddr_in multiAddr;
char* msg = "Hi There";
s = socket(AF_INET, SOCK_DGRAM, 0));
multiAddr.sin_family = AF_INET;
multiAddr.sin_port = htons(5432);
multiAddr.sin_addr.s_addr = inet_addr("224.0.1.1");
sendto(s,msg,strlen(msg),0,
  (struct sockaddr *)&multiAddr,
  sizeof(multiAddr));
```

We send from whatever arbitrary port number that was setup when the socket was created.

Receiving a Multicast

1) Set up your socket. Bind to the well-known port.
2) Join a multicast group.
3) Do a standard `recvfrom`.

```
int s;
struct sockaddr_in bindAddr, him;
struct ip_mreq mreq;
int alen = sizeof(struct sockaddr_in);
int Zero = 0;
char msg[20];
s = socket(AF_INET, SOCK_DGRAM, 0);
bindAddr.sin_family = AF_INET;
bindAddr.sin_port = htons(5432);
bindAddr.sin_addr.s_addr = INADDR_ANY;
bind(s,(struct sockaddr *)&bindAddr,
  sizeof(bindAddr));
mreq.imr_multiaddr.s_addr = inet_addr("224.0.1.1");
mreq.imr_interface.s_addr = INADDR_ANY;
setsockopt(s,IPPROTO_IP,IP_ADD_MEMBERSHIP,
  &mreq,sizeof(mreq));
recvfrom(s, msg, sizeof(msg), 0,
  (struct sockaddr *)&him, &alen);
```