

## Chapter 12 – Broadcast and Multicast

### Three types of IP address

Unicast	– A single interface on a single host
Broadcast	– All hosts on a single network (or networks)
Multicast	– Some group of hosts

### Packet filtering

LAN	Address mismatch and not (broadcast or multicast) Multicast ethernet mask: 01:00:00:00:00
Link Driver	Network layer Protocol unknown
IP	Transport layer protocol unknown
UDP	Port not usable

### Principle objective in broadcasting and multicasting

Force the discard to take place as low as possible if no one is interested

### IP Broadcast

Limited broadcast – 255.255.255.255

Used by a host that doesn't (yet) know its IP address and subnet mask

Datagrams sent this way are *never* forwarded by any router

Should multi-homed hosts send on all interfaces or not?

For BSD no.

Net directed broadcast – netid.255.255.255

Router must be able to forward net directed broadcasts *and*  
to disable such forwarding

Subnet directed broadcast – netid.subnetid.255

All subnets directed broadcast

Obsolete

Broadcasting examples:

Ping (OS/2) 130.127.48.255 –

Works as expected: all hosts respond

Ping (OS/2) 130.127.255.255 –

Response only from 130.127.44.1 (router on the back side of ours.)

## Disadvantages of net and subnet directed broadcasts

Creates excessive traffic  
Enables "Smurf" denial of service attacks when source address spoofed on broadcast ping.  
Recommended solutions for router managers:  
    Don't forward broadcasts *period*.  
    Don't forward packets carrying source addresses you don't own.

## Broadcasting with UDP

You may have to set the SO\_BROADCAST socket option before attempting a broadcast  
On some systems *root* privilege is required.

## Summary

Broadcast should be used with care.  
Your broadcast mileage may vary!  
Subnet directed Ping provides a good way for a hacker to easily retrieve all your IP addresses  
Multicast is probably a better plan.

## Multicasting on the Internet

Good starting-point references

    Van Jacobsen Tutorial – ACM SigComm '94  
    Comm. of the ACM – Aug 1994 – Special issue on Internet Technology

Primary target application

    Multimedia conferencing on the internet  
    Available tools:  
        sd, ivs, vat, nv (MBONE, CACM Aug 1994)

Historical perspective and distortions (Jacobson's view)

    Myths:

        Real-time traffic like audio and video requires a connection oriented virtual circuit network (preferably unreliable like ATM-SONET)  
        Datagram networks won't work because:  
            Lack of state information precludes resource reservation necessary to meet real-time demands  
            IP transit times are best effort and vary wildly

Proof of mythical nature of above statements:

    Existence counterexample:  
    The MBONE 10000+ users on 1500 networks in 30 countries

My view: the real issue is not *real-time traffic*, per se, but the *scope of the conference*  
2 – person conference – Unreliable connection oriented service works just fine.. consider your telephone  
n – person conference

Problems with the connection oriented approach

- To set it up all end users all have to know how to connect to each other
- N – participants requires  $N^2$  connections... bad scalability
- Reliability problems.. conference may fail if any of the connections do
- Difficult to joint in progress
- Have you ever *tried* to make a conference call?

Keys to IP's suitability

Its the one address of the conference itself that must be considered in routing  
No need to worry about the  $N^2$  connections

“Minor” problem

Privacy of conversations

Can be addressed by encryption

## IP multicast architecture

Process joins a multicast group by specifying (group\_address, local\_interface)..  
The host might have multiple interfaces group membership is interface specific.  
Any datagram sent using the multicast address goes to all members of the group

Class D addresses used

1 1 1 0 ----- 28 bit multicast group ID -----

Group addresses live in the range

224.0.0.0 – 239.255.255.255

Reserved addresses

224.0.0.0

224.0.0.1 – All hosts participating in IP multicast (Comer)

224.0.0.2 – All routers participating in IP multicast

To receive a multicast message you must belong to a multicast *host group*.

Membership is dynamic... hosts may enter and exit

Membership may cross networks

Membership is managed by IGMP

Some host group numbers are permanently assigned by IANA (see assigned # RFC)

## Mapping of Multicast to Ethernet addresses

Recall ethernet multicast mask 01:00:00:00:00:00

IANA reserved ethernet addresses in range  
01:00:5e:00:00:00  
01:00:5e:ff:ff:ff

IANA allocated half of these for multicast  
01:00:5e:00:00:00  
01:00:5e:7f:ff:ff

Mapping is:  
ENet address = 01:00:5e:low order 23 bits of IP

The low order bits of an IP address specify the multicast group thus,  
since there are  $2^{28}$  multicast addresses and  $2^{23}$  E-net mcst addresses,  
32 distinct multicast addresses map to a single E-net address.

## Hardware considerations:

There are  $2^{23}$  different multicast hardware addresses

How does an adapter determine filter out unwanted ones:  
Via a table lookup until table overflows  
Multicast promiscuous mode thereafter.  
(Other possibilities exist).

INTERNET MULTICAST ADDRESSES

Host Extensions for IP Multicasting [RFC1112] specifies the extensions required of a host implementation of the Internet Protocol (IP) to support multicasting. Current addresses are listed below.

224.0.0.0	Base Address (Reserved)	[RFC1112, JBP]
224.0.0.1	All Systems on this Subnet	[RFC1112, JBP]
224.0.0.2	All Routers on this Subnet	[JBP]
224.0.0.3	Unassigned	[JBP]
224.0.0.4	DVMRP Routers	[RFC1075, JBP]
224.0.0.5	OSPFIGP OSPFIGP All Routers	[RFC1583, JXM1]
224.0.0.6	OSPFIGP OSPFIGP Designated Routers	[RFC1583, JXM1]
224.0.0.7	ST Routers	[RFC1190, KS14]
224.0.0.8	ST Hosts	[RFC1190, KS14]
224.0.0.9	RIP2 Routers	[GSM11]
224.0.0.10	IGRP Routers	[Dino Farinacci]
224.0.0.11	Mobile-Agents	[Bill Simpson]
224.0.0.12-224.0.0.255	Unassigned	[JBP]
224.0.1.0	VMTP Managers Group	[RFC1045, DRC3]
224.0.1.1	NTP Network Time Protocol	[RFC1119, DLM1]
224.0.1.2	SGI-Dogfight	[AXC]
224.0.1.3	Rwhod	[SXD]
224.0.1.4	VNP	[DRC3]
224.0.1.5	Artificial Horizons - Aviator	[BXF]
224.0.1.6	NSS - Name Service Server	[BXS2]
224.0.1.7	AUDIONEWS - Audio News Multicast	[MXF2]
224.0.1.8	SUN NIS+ Information Service	[CXM3]
224.0.1.9	MTP Multicast Transport Protocol	[SXA]
224.0.1.10	IETF-1-LOW-AUDIO	[SC3]
224.0.1.11	IETF-1-AUDIO	[SC3]
224.0.1.12	IETF-1-VIDEO	[SC3]
224.0.1.13	IETF-2-LOW-AUDIO	[SC3]
224.0.1.14	IETF-2-AUDIO	[SC3]
224.0.1.15	IETF-2-VIDEO	[SC3]
224.0.1.16	MUSIC-SERVICE	[Guido van Rossum]
224.0.1.17	SEANET-TELEMETRY	[Andrew Maffei]
224.0.1.18	SEANET-IMAGE	[Andrew Maffei]
224.0.1.19	MLOADD	[Braden]
224.0.1.20	any private experiment	[JBP]
224.0.1.21	DVMRP on MOSPF	[John Moy]
224.0.1.22	SVRLOC	<veizades@ftp.com>
224.0.1.23	XINGTV	<hgxing@aol.com>
224.0.1.24	microsoft-ds	<arnoldm@microsoft.com>
224.0.1.25	nbc-pro	<bloomer@birch.crd.ge.com>
224.0.1.26	nbc-pfn	<bloomer@birch.crd.ge.com>
224.0.1.27-224.0.1.255	Unassigned	[JBP]

### Example multicast program:

```
struct ip_mreq mreq;
unsigned char mgroup[4] = {0xe0, 0x00, 0x00, 0xe0};
int on = 1;
        :
        :
    sock = socket(PF_INET, SOCK_DGRAM, 0);

/* Get host network address from command line parm */

hp = gethostbyname(argv[1]);
if (hp == 0)
{
    printf("Host %s not found\n", argv[1]);
    exit(1);
}
```

A multicast address -- like any other IP address -- is a (host, port) pair. Thus, if multiple processes on the same system want to all listen in, they must explicitly share the port.

```
status = setsockopt(sock, SOL_SOCKET, SO_REUSEADDR,
                    (char *)&on, sizeof(on));
if (status < 0)
    printf("Set reuse failed with %d \n", status);

bcopy((char *)hp->h_addr, &mreq.imr_interface, 4);
bcopy((char *)mgroup, &mreq.imr_multiaddr, 4);
```

The setsockopt function call *is required* for the socket to be multicast usable and generates an igmp add membership notification. For solaris systems the socket level should be set to IPPROTO\_IP (both are #defined as 0)

```
status = setsockopt(sock, SOL_IP, IP_ADD_MEMBERSHIP,
                    (char *)&mreq, sizeof(mreq));
```

The remainder of the example is just like a non multicast program.

```
/* Fill in protocol family and port # from command line then */
/* bind the socket to the specified address. */

name.sin_family = AF_INET;
name.sin_port = htons(atoi(argv[2]));
bcopy((char *)mgroup,
      (char *)&name.sin_addr, 4);

status = bind(sock, (struct sockaddr *)&name, sizeof(name));
printf("Bind status = %d \n", status);
if (status < 0)
    exit(1);

printf("Dest net address = ");
for (i = 0; i < hp->h_length; i++)
{
    c = *((char *)&name.sin_addr + i);
    printf(" %2x", c);
}

status = sendto(sock, "This is a multicast\n", 20, 0,
               &name, sizeof(name));
printf("%d bytes sent \n", status);
exit(1);

/* Wait for someone to send me a datagram */

namelen = sizeof(struct sockaddr_in);
status = recvfrom(sock, buf, sizeof(buf), 0,
                 (struct sockaddr *)&sname, &namelen);
```

TCP Dump output:

```
14:38:56.214849 jmw3 > 224.0.0.224: igmp nreport 224.0.0.224 [ttl 1]
14:38:56.221381 jmw3.55555 > 224.0.0.224.55555: udp 20 [ttl 1]
14:38:56.367139 jmw3 > 224.0.0.224: igmp leave 224.0.0.224 [ttl 1]
```