Multicast and Realtime Service[1]

HARDNESS : 6/10

University of Tehran
School of Electrical and Computer Engineering

دانشگاه تهران

دانشکده‌ی مهندسی برق و کامپیوتر

Computer Network Lab

آزمایشگاه شبکه‌های کامپیوتری

Professor:
Dr. Ahmad Khonsari
دکتر احمد خونساری
a_khonsari@ut.ac.ir

Amir Haji Ali Khamseh'i
امیر حاجی‌علی‌خمسه‌ء
khamse@ut.ac.ir

Reza Sharifnia
رضا شریف نیا
Reza.sharifnia@ut.ac.ir

Muhammad Borhani
محمد برهانی
borhani.m@ut.ac.ir

AmirAhmad Khordadi
امیراحمد خردادی
a.a.khordadi@ut.ac.ir

Sina Kashipazha
سینا کاشی‌پزها
sina_kashipazha@ut.ac.ir

Hadi Safari
هادی صفری
hadi.safari@ut.ac.ir

December 6, 2021

۱۵ آذر ۱۴۰۰

[1] S. Panwar, S. Mao, J.-dong Ryoo, and Y. Li, "Multicast and realtime service," in TCP/IP Essentials: A Lab-Based Approach, Cambridge: Cambridge University Press, 2004, pp. 134–158.

Objectives

- Multicast addressing.
- Multicast group management.
- Multicast routing: configuring a multicast router.

- Realtime video streaming using the **vlc**.
- Protocols supporting realtime streaming: RTP/RTCP and RTSP.
- Analyzing captured RTP/RTCP packets using **wireshark**.

Part I

Simple Multicast Exercises

For all the exercises in this section, the network topology is given in Figure 1.3, where all the hosts are connected to a single network segment using their default IP addresses, i.e. from 128.238.66.100 to 128.238.66.107.

Table 1: The IP addresses of the hosts (Table 1.2)

Host	IP Address	Subnet Mask
h0	128.238.66.100	255.255.255.0
h1	128.238.66.101	255.255.255.0
h2	128.238.66.102	255.255.255.0
h3	128.238.66.103	255.255.255.0
h4	128.238.66.104	255.255.255.0
h5	128.238.66.105	255.255.255.0
h6	128.238.66.106	255.255.255.0
h7	128.238.66.107	255.255.255.0

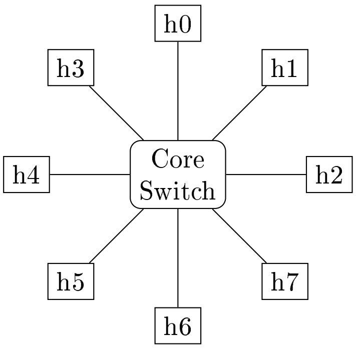

Figure 1: A single segment network (Figure 1.3)

1 Linux Multicast Routing Table

Execute the following command to display the routing table of one host (for example *h0*).

h_0's Console

```
netstat -rn
```

Note: If there is no entry for the 224.0.0.0 subnet, you need to provide a default route for multicast traffic, by: [1]

h_i's Console , i ∈ {0,1,...,7}

```
route add -net 224.0.0.0 netmask 240.0.0.0 dev eth0
```

Then, run the first command, *i.e.* `netstat -rn`. Save the new routing table.

h_0's Console

```
netstat -rn
```

[1] This command can be appended to the `/etc/rc.local` file, so that it will be executed automatically when the system bootstraps. Each time when the network interface is brought down and up again by the `ifconfig` command, you may need to run the `route` command to re-insert the multicast routing entry.

Report

1. Submit the routing table you saved.

2 Multicast Membership

Execute the following command to show the multicast group memberships for all the interfaces in your host (for example *h0*).

h_0's Console

```
netstat -g
```

Report

1. How many multicast groups did the interface belong to? What were the groups? Explain the meaning of the group IDs.

3 Multicast `ping`

Execute following command on *h1*:

h_1's Console

```
ping 224.0.0.1
```

Examine the `ping` output to see which hosts reply.

Ping a broadcast address using:

h_1's Console

```
ping -b 128.238.66.255
```

Examine the `ping` output to see which hosts reply.

Note: You can see and change broadcast ping replay state by (Not need):

```
cat /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts # see the kernel parameter
echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts # set 0 or 1 with kernel parameter
sysctl net.ipv4.icmp_echo_ignore_broadcasts        # see with sysctl command
sysctl -w net.ipv4.icmp_echo_ignore_broadcasts=0   # set with sysctl command
echo "net.ipv4.icmp_echo_ignore_broadcasts = 0" >> /etc/sysctl.conf # set in boot time
```

Report

1. Which hosts replied when the multicast address was pinged? Which hosts replied when the broadcast address was pinged? Verify you answers with hosts config `/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts`.

2. In each case, was there a reply from *h0* and *h1*?

4 Multicast vs Unicast

On *h1* execute `tcpdump -n -nn -e` and `tcpdump ether multicast -n -nn -e` (or run `wireshark`) to capture an Ethernet unicast frame, an Ethernet multicast frame, and an Ethernet broadcast frame.

h_1's Console

```
    tcpdump -n -nn -e            # or run wireshark
```

h_1's Auxiliary Console

```
    tcpdump ether multicast -n -nn -e       # or run wireshark
```

To generate an Ethernet unicast frame, run below command:

h_0's Console

```
    socket -i -u -n1 128.238.66.101 echo
```

Execute the following command in *h0* to generate an Ethernet multicast frame:

h_0's Console

```
    socket -i -u -n1 230.11.111.10 2000
```

Generate another Ethernet multicast frame, but with a different group address of 232.139.111.15, e.g.:

h_0's Console

```
socket -i -u -n1 232.139.111.15 2000
```

To generate an Ethernet broadcast frame, you may `ping` a remote host from *h0* that has no entry in the ARP table of your host, e.g. *h5*. [2]

h_0's Console

```
    ping 128.238.66.105
```

Recall that the ARP request is broadcast.

Save the frames captured for the lab report.

Report

1. Compare the source and destination MAC addresses of the frames you captured.
2. Use one of the multicast frames captured to explain how a multicast group address is mapped to a multicast MAC address. For the two multicast frames captured, do they have the same destination MAC address? Why?

5 Simple UDP Multicast Client and Server

Note: Go to *code* directory for *netlab* user at */home/netlab/code* with *cd* command. You can compile source code with `gcc $fileName -o $outputName`.

Start the multicast client `netspy` on all the hosts, by executing:[3]

h_i's Console , i $\in$ {0,1,...,7}

```
    /home/netlab/code/netspy 224.111.111.111 1500
```

Then, start the multicast sender `netspyd` on *h0*, by executing:

[2] You can see ARP table by `arp -a` delete a cached ARP entry by `arp -d ip`.

[3] If you get *Netspy : cannot join multicast group '224.111.111.111'*, add route for the groups as described in section 1.

h$_0$'s Auxiliary Console

```
/home/netlab/code/netspyd 224.111.111.111 1500 1
```

Execute `tcpdump ip multicast` or `wireshark` on every host to capture multicast IP datagrams. For example, in *h1*'s Console run below command:

h$_1$'s Console

```
tcpdump ip multicast         # or run wireshark
```

Login to *h0* from a remote machine, e.g. *h6*, using *telnet*

h$_6$'s Auxiliary Console

```
telnet 128.238.66.100
```

Use `netlab` as username and password.

Save the captured multicast datagram sent by `netspyd` and exit the `telnet` session.

Report

1. From the `tcpdump` output, how many messages are sent by `netspyd` when a new user logged in to *h0*? From the `netspy` outputs on all the hosts, how many copies of the message are received in total?
2. Did *h0*, where the multicast sender, `netspyd` , was running, receive the multicast datagram? Why? If yes, through which interface did *h0* receive this datagram?

6 ping Replay

Keep the `netspy` and the `tcpdump` (on *h6*) programs running. Execute the following command from *h6*:

h$_6$'s Auxiliary Console

```
ping 224.111.111.111
```

Examine the `tcpdump` and `ping` outputs to see which hosts replied.

Terminate the `netspy` client programs on several hosts, e.g. *h0*, *h1* and *h5*. Execute the `ping` command again.

Also, examine the `tcpdump` and the `ping` outputs (on *h6*) to see which hosts replied.

Part II

IGMP Exercises

In the following exercises, use four hosts and one router. The network topology is given in Figure 7.13, and the corresponding host IP addresses and router IP addresses are given in Table 7.2 and Table 7.3, respectively.

Table 2: Hosts IP addresses for Figure 7.13 (Table 7.2)

Name	IP Address
h1	128.238.63.101/24
h2	128.238.63.102/24
h3	128.238.64.103/24
h4	128.238.64.104/24

Table 3: Router IP addresses for Figure 7.13 (Table 7.3)

Host	eth0	eth1
router	128.238.63.3/24	128.238.64.3/24

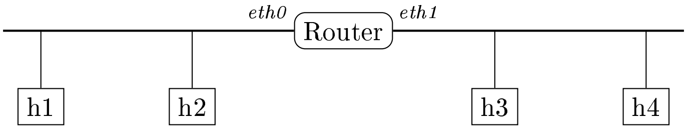

Figure 2: The network topology for IGMP Exercises (Figure 7.13)

You can use prepared topology as described in Part I.

7 Configuring Router

Connect the hosts and the route in your group as shown in Figure 7.13.

Login to the router (in GNS3, open router's console) and run `ip multicast-routing` to enable multicast routing in the *Global Configuration* mode. Then, enable the PIM protocol on each interface, by running `ip pim dense-mode` in the *Interface Configuration* mode. Now the router is enabled to do multicast routing using PIM.

R's Console

```
R# conf term
R(config)# ip multicast-routing
R(config)# int f0/0
R(config-if)# ip pim dense-mode
R(config-if)# exit
R(config)# int f0/1
R(config-if)# ip pim dense-mode
R(config-if)# end
```

Login to the router, execute the following commands in the *Privileged EXEC* mode:

R's Console

```
R# show ip igmp interface
R# show ip igmp group
```

Examine the multicast group memberships currently recorded in the router and the configurations of the router interfaces.

8 Multicast Message

Enable linux multicast routing in all the hosts (see section 1).

Note: Run below command to enable linux multicast routing(it is not need to run this command in *Figure 7.13*).

h_i's Console , i ∈ {1,2,3,4}

```
route add -net 224.0.0.0 netmask 240.0.0.0 dev eth0
```

Start `netspy` client on all the hosts, by using:

h_i's Console , $i \in \{1,2,3,4\}$

```
/home/netlab/code/netspy 224.111.111.111 1500
```

Start netspy server on *h1*, by using:

h_1's Auxiliary Console

```
/home/netlab/code/netspyd 224.111.111.111 1500 16
```

Login to the router. Run the following commands in the *Privileged EXEC* mode again to examine the current membership records:

R's Console

```
R# show ip igmp interface
R# show ip igmp group
```

Try if you can ping a host on the other side of the router. (e.g. ping *h4* from *h2*)?

h_2's Auxiliary Console

```
ping 128.238.64.104
```

Login to *h1* from the *h2* in your group, using telnet , then logout. (Use `netlab` as username and password). See if the multicast messages sent by netspyd reach the other side of the router.

h_2's Auxiliary Console

```
telnet 128.238.63.101
```

Report

1. Can you ping a host on the other side of the router? Will the router forward a multicast IP datagram to the other side? Justify your answers.

9 IGMP Types

Execute tcpdump or wireshark [4] in one console to capture IGMP messages.

h_1's Console

```
tcpdump ip multicast -v      # or run wireshark
```

When you see **six or more IGMP queries** in the tcpdump output, terminate tcpdump program.

Analyze the IGMP messages you captured. Print and save two different IGMP messages.

Repeat the above experiment. Terminate netspy on *h2* and *h4*. Terminate the tcpdump programs and analyze the IGMP leave message you captured.

Report

1. What is the value of the Time-to-Live (TTL) field for the IGMP messages? Why do we not set the TTL to a larger number?

2. What is the default frequency at which the router sends IGMP queries?

[4] If you have trouble in getting packets using wireshark , try tcpdump .

10 Router Join to Multicast-Group

Login to the router. See if you can make a router interface (e.g. *ethernet0*) join a multicast group of 224.0.0.2, using:

R’s Console

```
R# conf term
R(config)# ip multicast-routing
R(config)# interface f0/0
R(config-if)# ip igmp join-group 224.0.0.2
R(config-if)# exit
R(config)# end
```

Report

1. Explain why the above command fails.

Appendices

Appendix A Configuring a Multicast Router

The `no` form of this command cancels the group membership.

Appendix A.A Configuring IGMP

```
R1(config)# ip igmp join-group group-address
R1(config)# no ip igmp join-group group-address
R1(config)# ip igmp query-interval new-value-in-seconds
R1(config)# no ip igmp query-interval
show ip igmp groups   ! Displays the multicast groups in the attached networks.
show ip igmp interface ! Displays multicast related information on a router interface.
debug ip igmp         ! Displays IGMP packets received and transmitted.
```

Appendix A.B Configuring Multicast Routing

```
R1(config)# ip multicast-routing
R1(config)# no ip multicast-routing
R1(config)# ip pim [dense-mode | sparse-mode | dense-sparse-mode].
show ip mroute        ! Displays the multicast routing table.
show ip mroute summary ! Displays a one-line summary for each entry in the multicast routing
    table.
show ip mroute count  ! Displays multicast statistics.
show ip dvmrp route   ! Displays the DVMRP routing table.
show ip pim neighbor  ! Lists PIM neighbors discovered by the router.
show ip pim interface ! Displays router interface configurations.
```

Appendix A.C Cisco IOS Multicast Diagnostic Tools

```
mtrace
mrinfo
mstat
ping
```

Appendix B Wikipedia

You can reed more about multicast address in wikipedia.