CS444/544
Programming Project 3 - vi
Due: Thurs, March 3

In this next part of the project, we're going to take a step closer to building a real router for the Internet by implementing the basics of IPv6 addresses:

- Obtain an IP address
- Answer Neighbor Discovery (ARP) requests
- Answer ICMP echo requests

Obtaining a Local IPv6 Address

There are several ways to obtain an IPv6 address. For this project, we're going to start with the simplest form using a *link local* address. You should create a link local address for your computer as discussed in section 5.19 of the textbook.

Obtaining a Global IPv6 Address

In IPv6, global addresses are provided by routers using ICMPv6 Router Advertisement messages (see section 22.19 in your textbook). You don't have a router yet (because you won't build it until the next assignment). Until then, you can use a simple Router Advertisement Daemon (vrad) that we will provide. It will periodically (every 30 seconds by default) provide the needed information. This will provide you with a second, global IPv6 address for each of your interfaces. When this information becomes available, you should print it out and begin responding to that address as well.

Neighbor Discovery

Once you have an IPv6 address, you need to be prepared to give that information to your neighbors on the local LAN so that they can contact you. We will do that using Neighbor Discovery, which is very similar to the ARP mechanism used in IPv4. You can read about neighbor discovery in section 22.21 of the textbook, but you'll need to refer to RFC2461 for the details. It's a very simple protocol. Since you're the server in this dialog, reliability isn't your responsibility with this protocol. Use the algorithm:

```
if (ethernet multicast) {
  if (Neighbor Discovery request) {
    if (IPv6 address target is ``me'') {
      answer_the_question();  }}}
```

ICMP Echo Request / Echo Reply

The next step is to provide a server for ICMP Echo Requests (as in chapter 8, but using the protocol described in RFC2463). The algorithm should be:

```
if (ethernet dest address matches interface ethernet address) {
  if (IPv6) {
     if (IPv6 destination address is one of mine) {
        if (ICMP echo request) {
            answer_the_question();  }}}
```

You will test this functionality using a *vping* program that we will provide. It's basically the same as the normal *ping* program, but works with our virtual LANs.

Your vi program must support the following options:

-d

To print debugging information (more detailed output)

-l

List all interfaces (names and hardware addresses) and exit

-g NUM

Use group NUM. By default, you should use the group number that you were assigned.

-h Print a quick summary of the command line arguments and exit. This information should also be printed if you don't understand one of the command line arguments.

You should implement these three pieces of functionality in the order specified and verify their correct operation before continuing. Your program should not process any requests until it has determined the correct IP address for each of its interfaces. Your program doesn't need any detailed output, the correct behavior of the program can be observed from the output of vping and vdump. However, your program MUST print the IPv6 address of each of its interfaces as it determines them. With debugging enabled, it should also give information about the packets that it receives and the replies that it gives. This assignment will use the same virtual network setup as the vbridge project.