SE331: Introduction to Computer Networks	**Recitations 9**
Semester 1 5785	**8 Jan 2025**
Lecturer: Michael J. May	**Kinneret College**

IPv6 Communication

This week we'll implement some simple chat communication tools using IPv6 in Java. As with the regular IPv4 communication tools we've developed, we'll use the standard `Socket` and `ServerSocket` classes. The only difference is that in this case we'll be using the IPv6 family of addresses and so we must configure the classes appropriately.

1 IPv6 in Windows 11

Windows 10 supports both IPv4 and IPv6 so we can access the machine using either address. Looking at my laptop IP configuration, I have the following addresses:

```
Windows IP Configuration


Wireless LAN adapter Wireless Network Connection 3:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wireless Network Connection 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wireless Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : HOME
   Link-local IPv6 Address . . . . . : fe80::a1ac:21b6:2fc2:c677%10
   IPv4 Address. . . . . . . . . . . : 10.0.0.4
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.0.138

Tunnel adapter isatap.{DBBCB769-3C4E-4159-AFF2-6F44224BEF78}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter isatap.HOME:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : HOME

Tunnel adapter isatap.{8D7947A2-C1DE-4E18-B436-3121E06F0825}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Local Area Connection* 12:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2001:0:5ef5:79fb:1c0e:13b8:a674:b42d
   Link-local IPv6 Address . . . . . : fe80::1c0e:13b8:a674:b42d%24
   Default Gateway . . . . . . . . . : ::

Tunnel adapter isatap.{3482420A-32E5-4547-8A12-018464BAF498}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

Note that there are several IP addresses for the computer:

1. 10.0.0.4: My local NAT-based IPv4 address.

2. `fe80::a1ac:21b6:2fc2:c677%10`: My link local IP address (private IPv6). It's local to the 10 interface on the computer.
3. `2001:0:5ef5:79fb:1c0e:13b8:a674:b42d`: A (Teredo) tunneling IPv6 address used for sending IPv6 traffic over an IPv4 network. It tunnels IPv6 packets inside IPv4 ones.
4. `fe80::1c0e:13b8:a674:b42d%24`: The tunnel adapter's link local address. It's local to the 24 interface on the computer.

We can use any of the above addresses for communication, even talking on the same computer between different link local addresses.

Note: In order for two computers to communicate using IPv6, it's necessary that the router that connects them be able to speak IPv6. Last time I checked, the routers in Triguboff do not support IPv6, so it will be difficult to test the tool for real. My home router also does not support IPv6.

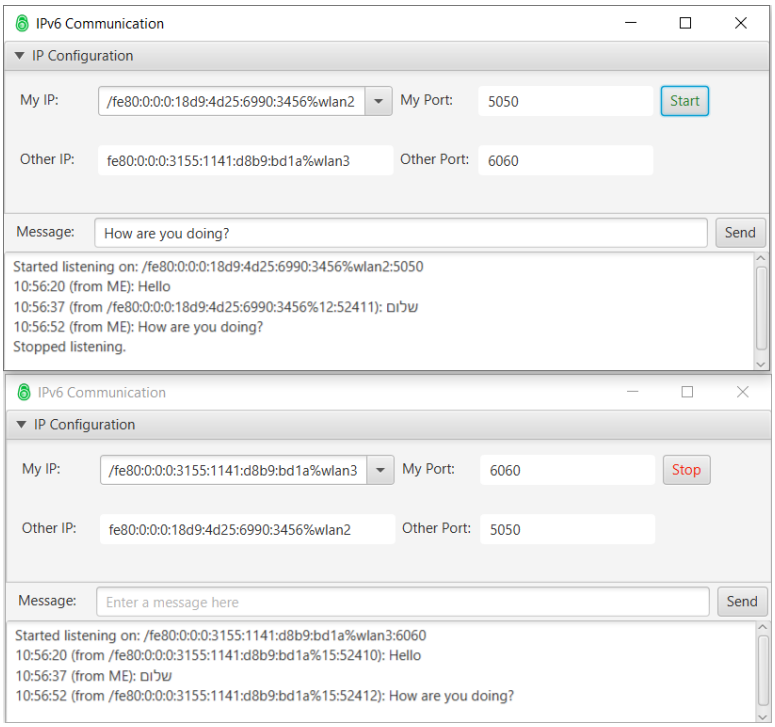
2 Chat Tool

We'll implement a simple chat tool using multithreading and IPv6 addresses in Java. The tool has the following features:

1. It can automatically fill in the IPv6 addresses owned by the computer. This is done using the DNS functionality in Java. The user selects the address to listen on using the Combobox on top.
2. It listens for incoming messages using a thread that the GUI can start and stop listening at will. Messages that arrive are sent as single lines, so there are no lingering connections left open.
3. Messages sent are sent to the recipient using the GUI thread in a single action so as not to make the GUI hang.
4. The log at the bottom of the screen shows all messages sent and received including a time stamp.
5. The tool checks that the provided ports and IP addresses are valid. If they're not, the tool shows the problematic fields in red.

3 GUI Details

A screen shot of the tool in Java using JavaFX is shown below:



To make the tool work correctly, you'll need to implement a listening thread and a mechanism for inter-thread communication. I wrote my version of the tool using the following classes:

1. Task<Void> - `https://openjfx.io/javadoc/21/javafx.graphics/javafx/concurrent/Task.html`
2. Thread
3. Task.messageProperty (Javadoc link) to send updates from the Task to the parent GUI thread.

4 What to do

Use the starter JavaFX GUI and code to complete the IPv6 communication tool. There are two possibilities for how to run and test your code:

1. If your computer network supports IPv6 communication, you can run the program and test it on your home computer.
2. If your computer network does not support IPv6, you can create a virtual NAT network in Virtual Box that does support IPv6. Then you can run the program on two Linux Virtual Machines that are connected to the virtual NAT network. The two Virtual Machines will then be able to communicate using IPv6.