

## CSIS0234B Computer and Data Communication (Class B)

### Tutorial 10

### Multicast file transfer

In tutorial 3, you've written a program to download a file from a UDP server. In this tutorial, you're going to do a similar task, but using multicasting. The **client** starts first, listening to a well-known port 12345 after joining a multicast group. The server will then multicast a single UDP message that provides the whole file.

For reference, sample programs of tutorial 3 (`UDPServer.cc` and `UPDClient.cc`) are placed in the computer, under the network user. You can modify them instead of rewriting the whole program. Here are the detailed steps.

1. Modify the server so that it sends the file `orig.txt` to the multicast address specified on the command line as a single UDP datagram, towards port 12345. The port number used by the server itself is not important (so it needs not bind to an address). It should not wait for the client to send it a packet before sending. Use the group address `239.1.0.x` (where `x` = the computer number of your computer) to test your program—run `etherreal` in order to capture the packet on that port to learn to which network interface the packet is sent.
2. By adding a route as described in the tutorial note, see whether you can force the packet to the `lo` interface. At the end, remove the routes you add, and make sure that the packet is sent to `eth0`.
3. Modify the client program so that it binds to the port 12345, and listen to the multicast address specified on the command line. It should not try to send anything to the server at the beginning. Instead, just wait until a datagram is received. Then the content of the datagram should be saved in the file `received.txt`. Again, use `239.1.0.x` to test your client, without running the server. Check `/proc/net/igmp` in order to make sure you have joined the right group at the correct interface (`eth0`).
4. Now run the server program when the client is already running. Make sure the client can correctly receive the file by looking for the file `received.txt` and comparing it to the `orig.txt` file.
5. Try to send the file to multiple computers at the same time, and make sure that all of them receives the same file.
6. Try to listen to two interfaces at the same time. Hint: you can use the `if_nameindex` function to get a list of interface, defined in `<net/if.h>`, to find the indices of all interfaces configured in the host. (In our lab, there are 3: `lo`, `eth0` and `eth1`.) After modifying the program, read `/proc/net/igmp` to make sure that the port is binded to all the interfaces you specified.