



Mobile Computing

Exercise 1

Assigned: October 31, 2005

Due: November 7, 2005

1 Ad-Hoc Wireless LAN Connectivity

The purpose of this exercise is to switch your Wireless LAN cards (typically installed on a notebook¹) to the *ad-hoc* mode and to check their operation by sending a message between two devices. (Before switching to the *ad-hoc* mode, read on and download all required files to your computer.)

Use the configuration tool shipped with your Wireless LAN card to switch to the *ad-hoc* mode². If you are using Windows on a Neptun IBM notebook the configuration tool is called *IBM Access Connections*. In case of the Cisco Aironet card, the tool is called *Aironet Client Utility*. Use the tool's wizard to set up a new connection. Set the Service Set ID (*SSID* for IBM Access Connections and Cisco Aironet; *Network Name* for Lucent WaveLAN and Apple AirPort cards) to *mobicomp*. Since there is no DHCP available you must set your IP address manually. To prevent the problem that several people in the room accidentally use the same address, you should only use the IP address you receive on a label during the first exercise lesson. The subnetmask should be set to 255.255.255.0

Note that using the *ad-hoc* profile, you won't be able to connect to the base station infrastructure unless you switch back to the according profile.

As a second step and if necessary, install the Java runtime environment on your mobile device (<http://java.sun.com/products/>).

Now download the two files `Sender.jar` and `Receiver.jar` from the lecture homepage (<http://dgc.ethz.ch> → lectures) to your mobile devices. Start the `Receiver` on one, the `Sender` on another device, type in some text and send it off. If the receiver device confirms receipt of a message, your environment should be set up correctly for the rest of the practical exercises.

Finally an important hint: If you have a firewall running, turn it off. (Don't forget to turn it on again after the exercises.)

2 SANS: Simple Ad-Hoc Network Simulator

For certain parts of the course exercises we are going to use a program simulating an *ad-hoc* network. In this exercise you will install and test this simulator. Download the `Simulator.jar` file from the lecture homepage and try to get the above `Sender` and `Receiver` programs running within the simulator. Also read the short manual on Page 2 for more information on the simulator.

¹We assume most of you use notebooks for the exercises of this lecture. In case you want to use something "exotic", such as a Compag IPAQ device, we can give you some, though not full support.

²If you haven't been using a Wireless LAN card yet, take a look at <http://www.wireless.ethz.ch>, where you will find instructions on how to connect to the ETH Wireless LAN via one of the Wireless LAN base stations.

3 Application Scenarios

Write down some notes on what you think might be possible scenarios for mobile computing. Feel free to suggest anything that occurs to be more or less useful to you. Can you even think of a so-called *killer application* for mobile computing?

4 Short Simulator Manual

Starting the Simulator The simulator is started from the console with the command `java -jar Simulator.jar`. It can also be started by double-clicking the jar file if this function is supported by the operating system. Of course the programs to be run in a node must be already compiled.

Editing the graph On the left side of the main window of the simulator, the graph can be edited. With the right mouse button clicked in an empty location, a new node can be added, while a right click on an existing node opens a window to edit its properties. The left mouse button is used to change the position of a node and to add or remove links. A node can be moved around by dragging it in the middle. When clicked on its border, a new link starting at this node is created. To remove a link, it must be grabbed at one of its arrows and released in an empty area of the graph.

The graph can also be edited by the use of the buttons on the right side of the window. These buttons offer even more possibilities to change the graph, e.g. adjusting the properties like the delay or the error probability of a link. A graph can also be saved or loaded using the corresponding buttons.

Configuring a node After clicking the right mouse button in an empty position of the main window a new node is created at this position and a configuration dialog pops up. In the first field “Class name (incl.package)” the main class of the application to be run on this node must be specified. Note that the full name of the main class needs to be specified including its package (e.g. `ch.ethz.dcg.sample.MainClass`). Also note that if a jar file is to be run on the node, the main class within the jar file needs to be specified, once again including its package.

The field “Classpath” must be used to specify where the classes of the application to be run on the node can be found. If the application is within a jar file, the path to the jar file including the file name must be entered. (e.g. `c:\mobicomp\Test.jar`)

Starting the Simulation Process When the graph is drawn and the class name including the parameters for `main()` of each node is specified, the simulation process can be started by pressing the “Start Clients” button. Even when the programs are started, the graph can still be edited and new clients can be added. A subsequent click on the “Start Clients” button activates the recently added programs.

Stopping a Client To remove a client from the simulation, the “Remove Client” button should be used. The simulator then removes all links of the corresponding node and closes all sockets of the client. The window of the client remains visible. If this window is closed, it is very likely that the client executes `System.exit()`. This would result in the termination of the whole simulator.