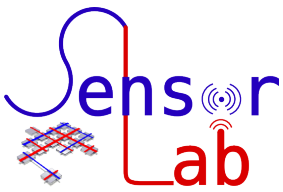
Faculty of Mathematics and Computer Science

– Georg-August University Göttingen –

Telematics Group

Lab 4 - TinyOS Networking

Sensorlab



Contents

1 Introduction

This lab is all about networking with TinyOS. Two very important protocols are provided. The goal of a dissemination protocol is to reliably deliver a piece of data to every node in the network. It allows administrators to reconfigure, query, and reprogram a network. Reliability is important because it makes the operation robust to temporary disconnections or high packet loss. Collection is the complementary operation to disseminating and it consists in "collecting" the data generated in the network into a base station. The general approach used is to build one or more collection trees, each of which is rooted at a base station. When a node has data which needs to be collected, it sends the data up the tree, and it forwards collection data that other nodes send to it. For an introduction, please read the following links:

- `http://tinyos.stanford.edu/tinyos-wiki/index.php/Network_Protocols`
- `http://tinyos.stanford.edu/tinyos-wiki/index.php/Dissemination`

Question 4.1 (3 Points)
Describe the hidden terminal problem and explain how to solve it.

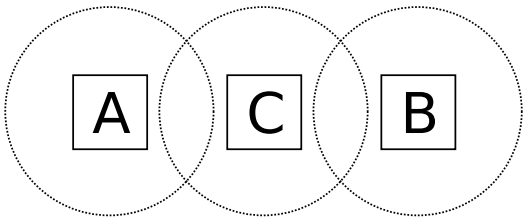


Figure 1: Hidden terminal problem

Question 4.2 (3 Points)
Describe also the exposed node problem and explain how to solve it.

Question 4.3 (3 Points)
Describe the trickle algorithm (`www.scss.tcd.ie/publications/tech-reports/reports.10/TCD-CS-2010-22.pdf`). What is its purpose?

Question 4.4 (3 Points)
Which other network protocols does TinyOS provide? Explain them briefly.

2 Application development

Now consider the following scenario: You have a number of rooms. One of them provides a reference brightness. You want to measure the brightness in that room and other rooms. If the brightness of the other rooms falls below that of the reference room, you want to turn on an LED on the mote that measured the sub-standard brightness. You also want to display the measurements together with node IDs as well as the reference brightness on a PC connected to a base station.
For calibration, have a node with a specific ID (use `TOS_NODE_ID` to determine the current node ID) disseminate its measured light level as the reference/-calibration value. Use the Drip dissemination protocol. Have all motes send out messages containing their node ID, calibration value and measured light levels, and display them at the base station.
Should the use of these protocols fail or seem inappropriate (give reasons), use regular radio communications (see lab 2) instead.

Assignment 4.1 (30 Points)
Test your program and include a description of your results as well as the source code in your lab report. Use the Java-program from lab 2 in order to provide clear and structured output of exchanged data.

Hints:

- You will find the required interfaces on the web pages mentioned. It could also be helpful to understand examples for the protocol usage to be found `/opt/tinyos-2.1.2/apps/`.
- Since usually multiple persons are working on this assignment at the same time, it would be a good idea to customize the ID (the example code uses `0x1234` and `0x5678`) of the disseminated values so there are no conflicts. Also customize the ID used for `AMSenderC` and `AMReceiverC` as in lab 2 to avoid interference.

3 Practical introduction to Deluge

As you have seen two labs ago, the Deluge protocol can be used to reflash a network of sensor motes with new programs. In the course of this lab, we will take a more practical look at Deluge and use it to disseminate programs. During this lab, we will be roughly following:
`http://tinyos.stanford.edu/tinyos-wiki/index.php/Deluge_T2`.

Due to the nature of this lab, please be sure to coordinate with the other students, so your work does not interfere with each others.

4 Setup

We can run a mostly automatic test of deluge to confirm that everything is in working order. Attach a mote to the USB programmer connected to the PC and execute the following commands:

```
cp -a /opt/tinyos-2.1.2/apps/tests/deluge/Blink $HOME/DelugeBlink
cd $HOME/DelugeBlink
./burn mib520,/dev/ttyUSB0 serial@/dev/ttyUSB1:57600 iris
```

Now follow the instructions on the screen and make sure the mote behaves as expected. Did it work?

5 Preparing Motes

First, a mote has to be prepared as the Deluge base station. This is different from the regular base station and can not be used to listen for received, general packets. Connect only the mote you wish to use as the base station to the PC, using the USB programming board. Navigate to the Deluge base station directory and install the application as usual. Make sure you set the node ID to 0.

```
cp -a /opt/tinyos-2.1.2/apps/tests/deluge/Basestation $HOME/DelugeBasestation
cd $HOME/DelugeBasestation
make iris install,0 mib520,/dev/ttyUSB0
```

Keep the base station connected and connect a second programming board via USB. Alternatively, you can remove the base station, program the nodes and reconnect it. In that case, you will have to change the device names in the next part of this section.
Now you are ready to prepare the motes you want to use for use with Deluge. For this, it is necessary to install a basic, deluge capable application on them. Usually, the GoldenImage application is used for this. Make sure to assign a unique node ID to each mote you prepare (remember that 0xFFFF is reserved as the broadcast ID).

```
cp -a /opt/tinyos-2.1.2/apps/tests/deluge/GoldenImage $HOME/GoldenImage
cd $HOME/GoldenImage
make iris install,1 mib520,/dev/ttyUSB2
make iris install,2 mib520,/dev/ttyUSB2
...
```

After preparing the client motes, attach them to power sources. The preparations are complete now.

6 Using Deluge

Before an application can be spread through the network using Deluge, some simple modifications have to be made to it.
Please use an application (or a copy thereof) you previously built. (You can also use a copy of Blink.)
Important note: It might be necessary to press the reset-button once or twice every time you want to inject a new program into the base station. Keep that in mind while working through the next steps!

6.1 Preparing the Application

The first step, is to add the Deluge component to your applications configuration:

```
    components DelugeC;
```

Additionally, add the following line near the top of your `Makefile`:

```
BOOTLOADER=tosboot
```

Deluge also requires a flash memory specification. Copy the following to a file called `volumes-at45db.xml`.

```
<volume_table>
    <volume name="GOLDENIMAGE" size="65536" base="0" />
    <volume name="DELUGE1" size="65536"/>
    <volume name="DELUGE2" size="65536"/>
    <volume name="DELUGE3" size="65536"/>
</volume_table>
```

To learn more about the use of flash memory from TinyOS, you can read: `http://tinyos.stanford.edu/tinyos-wiki/index.php/Storage` The details are not really important to use deluge though.

6.2 Disseminating the Application

Once the application is prepared, build it, but do not install it (`make iris`). Now it can be sent to one of the three Deluge image slots available in the base station.

```
tos-deluge serial@/dev/ttyUSB1:57600 -i 1 build/iris/tos_image.xml
```

The number (1 in this case) specifies which of the image slots are used for storing the program.
Once the program is transferred (in the case of failure due to memory write errors, try rebooting the base station using the reset button on the programming board), it is ready to be disseminated.

```
tos-deluge serial@/dev/ttyUSB1:57600 -d 1
```

The above command gives the base station the command to disseminate the program image among the client motes. This will take a little bit of time, so please wait.
Once the dissemination should be finished, you can have the base station disseminate a reflash command:

```
tos-deluge serial@/dev/ttyUSB1:57600 -dr 1
```

The client motes should now reboot to the new program.

6.3 Disseminating a Modified Application

Now you should make a (visible, e.g. involving LEDs) modification to your application. Use the steps from the last section to reflash the motes to use the new version. Do not forget to execute `make iris`.

6.4 Using a Second Image Slot

Make another modification to your application. This time inject, disseminate and reflash it using image slot 2 instead of 1.

6.5 Using Multiple Image Slots

Now the motes should all have two programs stored in separate image slots. Try switching between them, using the `-dr` command. There should be **no** need to inject programs to the base station (`-i`) or disseminating them (`-d`).

```
tos-deluge serial@/dev/ttyUSB1:57600 -dr 1
tos-deluge serial@/dev/ttyUSB1:57600 -dr 2
tos-deluge serial@/dev/ttyUSB1:57600 -dr 1
tos-deluge serial@/dev/ttyUSB1:57600 -dr 2
```

6.6 Assignment 4.2 (8 Points)

Describe your observations during the previous steps **in detail**, e.g. explain the use of different slots.