

The improvement of mobile code dissemination program

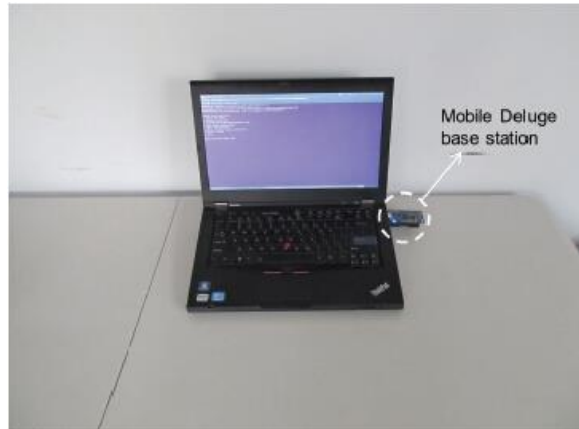
Final Report

Author: Zhihao Cao

Abstract

Wireless sensor network has gained increasing attention and been used more and more in environmental monitoring applications. For long term deployment, the network maintenance becomes a key issue. Beside replacing batteries and fixing broken nodes, network maintenance also involves reprogramming nodes for updating and improving applications. However, manually reprogramming sensor nodes is very cumbersome because we need to retrieve the location of each target node. To address this problem, a program called MobileDeluge is introduced in this paper.

The program is created by a group of IUPUI Computer Science PHD students. This program is a hand-held code dissemination tool for outdoor WSN deployments over low power links. It enables wireless reprogramming of WSN nodes in harsh but accessible environments within a one-hop neighborhood with respect to the hand-held Deluge base station. MobileDeluge creates a control service to coordinate the mobile Deluge base station and the target sensor nodes within the neighborhood of the mobile Deluge base station for code dissemination. The key idea is to establish an instant connection between the mobile Deluge base station and its target sensor nodes within the neighborhood, where the target nodes are to be updated with the same new code image.

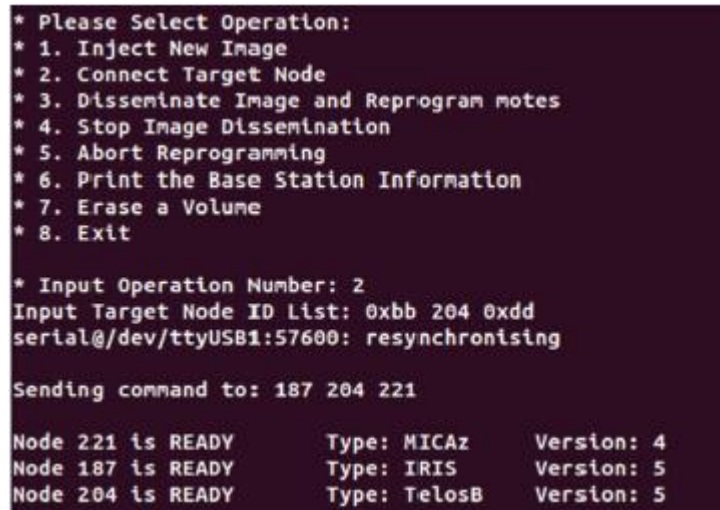


Due to the program interacts user with a console window, it only contains basic functions about node connecting and image disseminating. For example, when user uses this program in the field, they have to input the ID of each node by hand to create a connection with a node. It is a cumbersome job to do that for each node because it may not succeed to create connection to some nodes. If it does not, user has to input another ID to continue creating connection. By inputting each node ID to create connection in each time is really not an efficient way in field work. This program needs to manage about 60 nodes in a field so improving the efficiency of this program is an important job to make it practical.

In this paper, I will explore the problems of this program, and explain the way that I designed to improve the program.

Introduction

MobileDeluge is a useful tool for users to maintain wireless sensors in field work. It prevents users to retrieve and maintain the node one by one so that it saves a lot of time for users. However, this program is not easy to use since the way that it offered for users to interact with is so primitive.



```
* Please Select Operation:
* 1. Inject New Image
* 2. Connect Target Node
* 3. Disseminate Image and Reprogram notes
* 4. Stop Image Dissemination
* 5. Abort Reprogramming
* 6. Print the Base Station Information
* 7. Erase a Volume
* 8. Exit

* Input Operation Number: 2
Input Target Node ID List: 0xbb 204 0xdd
serial@/dev/ttyUSB1:57600: resynchronising

Sending command to: 187 204 221

Node 221 is READY      Type: MICAz      Version: 4
Node 187 is READY      Type: IRIS       Version: 5
Node 204 is READY      Type: TelosB     Version: 5
```

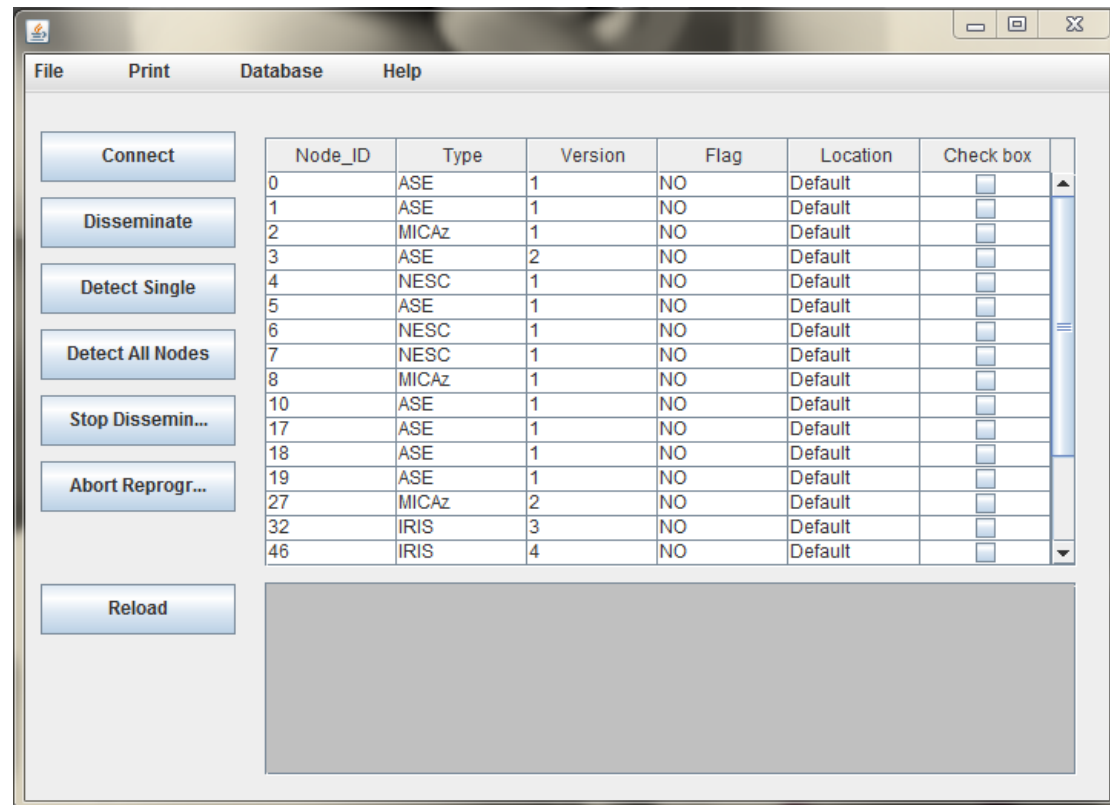
The image shows the interface offered by MobileDeluge. There are mainly three parts shown in this image. The first part is the menu options; the second part is the input line; the third part is the status indication. When the program runs, the menu and input line will show. The menu offers seven options and each of them will achieve a function that designed in the program. When users type an option in the input line, the running information will show in the bottom of the window. User can observe the node identifications, status, types and versions which are necessary information for operator.

Due to the limited information offered from the interface and obsolete way to control the program, it is inconvenient for user to do maintenance in field work. I have explored some problems that decrease the usability of this program:

1. It is very inconvenient when the program is applied to dozens of sensor nodes.
Because this program needs to establish a connection before user do any operations on a target sensor node, and the only way to establish a connection is to type manually a target sensor node ID number on a console window.
2. To know which sensor node needs reprogram, user has to get connected with that node at first.
3. If some sensor nodes are not reachable, user could not get any information of them.

To improve the program, an efficient GUI and database should be design and add to the program.

Project Design



The image shows the GUI design for the program.

The original menu is converted into a list of buttons that settles on the left hand side of the main window. A database table and an information panel are settled on the right. In the menu bar, the first menu option is to import code image. The second menu option is to print the table. The third menu option offers database edit mode. The Fourth menu option offers instruction helps.

When MobileDeluge is opened, the main window will show up. The program will load the database data into the table.

In the main window, user could select a button to do operation.

Connect button: user needs to check at least one node before do this operation. If the selected nodes are reachable, the information panel will show the information of these nodes and the program is connected to them.

Disseminate button: user needs to have the connected status with target nodes. When this operation is applied, the program will send code image to target nodes, and the information panel will show the status of this operation.

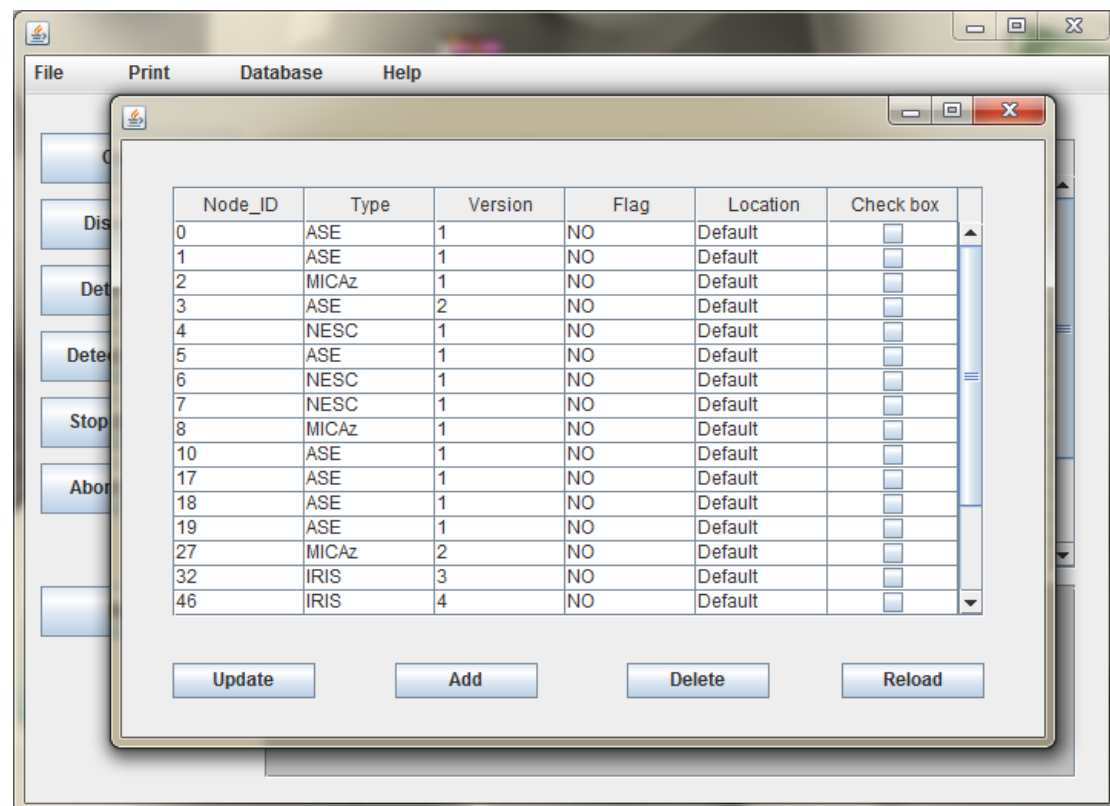
Detect Single button: user could check if the selected nodes are reachable. The results will show on the information panel.

Detect All Nodes button: the program will send a package to all nodes in the database. The results will show on the information panel.

Stop Disseminate button: the program will request the target nodes to stop receiving code image mode.

Abort Reprogram button: the program will request the target nodes to stop reprogram mode.

Reload button: the program will update table based on the database.



In the database edit mode, there are four buttons on this window. The table is editable in this window.

Update: Save all changes into database.

Add: Insert a new node into database.

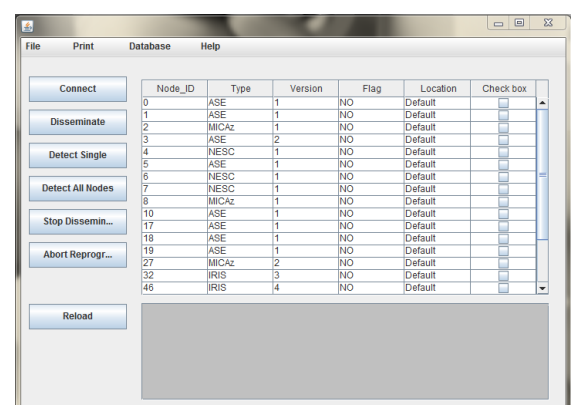
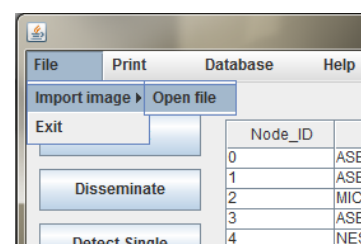
Delete: Remove selected nodes from database.

Reload: Refresh the table.

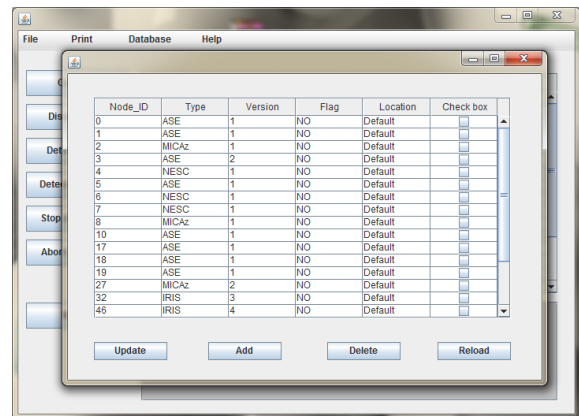
Implementation and Data Summary

At first, user needs to import a code image file into program. A variable will get the path of the code image and send to back end.

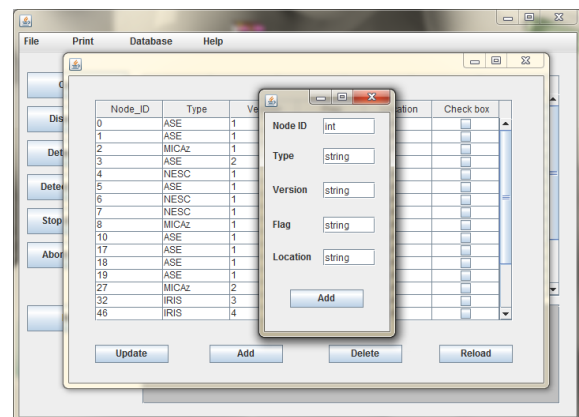
In the table, when a check box is checked, the node id value will be added to a list. When operation is selected, the program will send the node id list to the back end. In the back end, the program will separate the node id and send packages to each node id. And then wait for reply packages.



When select the database edit mode, a database window will be opened. The program draw a new table from database and the table is editable. User can edit every cell except Node ID cell and when Update is applied, all changes will be saved. The check boxes will save according node IDs into a list. Delete operation will remove all nodes in the list from database.



When Add button is clicked, an Add node window will be opened. In this version, there are five boxes that need to fill. The text in the box indicates the data type required by the database. When add operation is applied, these five data will be sent to a database update function. And then, database will be updated.



As this report mentioned above, first, it is inconvenient when applying to numbers of nodes. In the improved version, user does not need to create connections to numbers of nodes manually. It just needs user to tick the type of nodes by looking at the table and apply operations.

Second, in the original version, user does not know which node needs update or maintenance; in the improved version, user just needs a glance to the table and select nodes that need operations.

Third, even when a node is not reachable, user could still know the information of the node by looking up the table.

In the database edit mode, it offers convenient functions to maintain the database. In the real world, manually maintaining nodes happens usually because some nodes may be out of battery, stuck in bug, broken and so on reasons. If a node is updated or maintained not by this MobileDeluge program, the database's data will be not right. Therefore, the database edit mode could correct the database due to this situation and could keep the node date and database data synchronized.

Results and Discussion

The improved MobileDeluge is tested by a creator of this program. The experiment is to simulate doing operations for nodes and updating data in database. There are five real nodes are brought to this experiment. Firstly, he opens the program, imports the code image by selecting the file in the menu bar. Then, he selects "Detect All" to see if

there five nodes are reachable. Then, the information panel shows the result that five nodes are detected and the details of these nodes follow each node ID. Since this is first time to detect these five nodes, database has no their data yet. Then he goes to table edit mode, and insert the information of these five nodes into database. In this time, the database is synchronized with nodes. Then, he tries to update them. He checks the checkbox of these five nodes and clicks "Connect" button. Then information panel returns the feedback showing successful for each node. Then he clicks "Disseminate" button to transmit code image to these five nodes. When the information panel returns the successful feedback from each node, the code image starts to be reprogramed inside each node automatically. And the meantime, the program's database will update the version column automatically. The update operations end in this step. If nodes are stuck when receiving code image or reprogramming, "Stop Disseminate" and "Abort Reprogramming" buttons can save them from these situations.

The experiment shows the truth that this program become much easier and efficient to use.

Reference

- MobileDeluge: Mobile Code Dissemination for Wireless Sensor Networks, http://www.researchgate.net/publication/268034283_MobileDeluge_Mobile_Code_Dissemination_for_Wireless_Sensor_Networks