1. How to compile the program

The program was developed in **JAVA** language using Eclipse 3.5 IDE. To compile and run this program, JDK 6 and Eclipse 3.5 are needed. JDK 6 can be downloaded at http://java.sun.com/javase/downloads/widget/jdk6.jsp and Eclipse 3.5 at http://cloud.eclipsesource.com/R/eclipse-java-galileo-win32.zip .

1) Install JDK 6

   Install the **java.exe** file downloaded and add to the Environment Variable **PATH** "**C:\Sun\SDK\bin**" (default)

2) Install Eclipse 3.5

   Unzip the **eclipse-java-galileo-win32.zip** file downloaded from the above address and you will find the executable file **./eclipse/ eclipse.exe**. Double click the icon of the executable file, and Eclipse IDE will start.

3) Import the project

   Click "File" à "Import", and choose "Existing Projects into Workspace" under "General" folder. Then click "Next" and select the root directory by clicking "Browse". Click "Finish" after choosing the root directory of the project.

4) Compile & Run the program

   There are two JAVA files in the project, named Pastry.java and PastryUI.java. **Pastry.java** is the logic part and **PastryUI.java** is the UI part. To run the program, you can click "**Run**" à "**Run as**" à "**Java Application**" in ***PastryUI.java***.

   An executable *.jar* file ***PastryUI.jar*** is also provided. After installing the JRE (Java Running Environment), you can run this file by typing "java –jar PastryUI.jar" at the command line.

2. How to use the program

```
|-------------------------------------------------------------------------|
|                        Welcome to Pastry System                         |
|        Assignment 1 for CSIS8301 Advanced Topics in Computer Systems    |
|      Developed by ZHANG Chenggang (U no.: 2009950026), CS Department, HKU |
|-------------------------------------------------------------------------|

Please choose the working mode: 0 for USER mode, 1 for EXPERIMENT mode.
```

Welcome Information

Choose Working mode

```
Please choose the working mode: 0 for USER mode, 1 for EXPERIMENT mode.
0

Currently there are 2 protocols implemented, please choose one to work with:
   0: Improved Pastry Protocol (Highly Recommended!)
   1: Original Pastry Protocol
```

Working Mode

Choose Protocol

```
Currently there are 2 protocols implemented, please choose one to work with:
   0: Improved Pastry Protocol (Highly Recommended!)
   1: Original Pastry Protocol
0
For USER mode, the default setting is: base = 4, an ID has 4 quaternary digits
Please input the size of the leaf set, |L| (needs to be an even number like 2,4,6...) :
```

Protocol ID

Default Setting

Input |L|

```
For USER mode, the default setting is: base = 4, an ID has 4 quaternary digits
Please input the size of the leaf set, |L| (needs to be an even number like 2,4,6...) :
4
```

Size of |L|

```
Please input the size of the neighborhood set, |M|:
4
```

Size of |M|

```
Please enter the NodeIDs you want to insert, each nodeId on one line and ending with -1!
For Example:
1102
2131
3100
-1
```

⇓

```
Please enter the NodeIDs you want to insert, each nodeId on one line and ending with -1!
For Example:
1102
2131
3100
-1
```

Example Input

```
1234
The node ID is INVALID! Please re-enter:
21312
The node ID is INVALID! Please re-enter:
fadf
The node ID is INVALID! Please re-enter:
323112
The node ID is INVALID! Please re-enter:
```

Exception Handling

```
1230
Node "1230" has been successfully inserted!
Totally   1   nodes are in the Pastry network,
Please enter the next node ID or "-1" to end:
```

Successfully Insert One Node

```
1230
Inserting failure!
Node "1230" has already been in the network!
Please enter the next node ID or "-1" to end:
```

Exception Handling for inserting existing node

```
1233
Node "1233" has been successfully inserted!
Totally   2   nodes are in the Pastry network,
Please enter the next node ID or "-1" to end:
1302
Node "1302" has been successfully inserted!
Totally   3   nodes are in the Pastry network,
Please enter the next node ID or "-1" to end:
3201
Node "3201" has been successfully inserted!
Totally   4   nodes are in the Pastry network,
Please enter the next node ID or "-1" to end:
2313
Node "2313" has been successfully inserted!
Totally   5   nodes are in the Pastry network,
```

Inserting nodes

```
Please enter the next node ID or "-1" to end:
-1
Finished inserting nodes!
```

Finish inserting nodes

```
|----------------------------------------------------------------------|
|Please choose the operations as follows:                              |
|    l: List ALL the nodes                             e.g.  l         |
|    c: Check the Routing table and L,M set of a node  e.g.  c3333     |
|    r: Route a message starting from some node        e.g.  r2312     |
|    q: Quit                                           e.g.  q         |
|----------------------------------------------------------------------|
l
********     ALL the nodes     ********
  1230
  1233
  1302
  2313
  3201
********     ALL the nodes     ********
```

Operations Prompt

Command for listing all nodes

List All nodes

```
|----------------------------------------------------------------------|
|Please choose the operations as follows:                              |
|    l: List ALL the nodes                             e.g.  l         |
|    c: Check the Routing table and L,M set of a node  e.g.  c3333     |
|    r: Route a message starting from some node        e.g.  r2312     |
|    q: Quit                                           e.g.  q         |
|----------------------------------------------------------------------|
c1230
|*****************   Node  "1230"'s state infromation   *************
Routing table:
-----------------------------------------------
       ||   SELF   ||   2313   ||   3201   ||
-----------------------------------------------
       ||          ||   SELF   ||   1302   ||
-----------------------------------------------
       ||          ||          ||   SELF   ||
-----------------------------------------------
SELF   ||          ||          ||   1233   ||
-----------------------------------------------
Leaf set:       || 1233  1302
Neighborhood set:     1302  3201  1233  2313
|*****************   Node  "1230"'s state infromation   *************
```

Command for checking "1230"'s state information

Routing table information

Leaf set information

Neighborhood set information

```
|----------------------------------------------------------------------|
|Please choose the operations as follows:                              |
|    l: List ALL the nodes                                e.g.   l     |
|    c: Check the Routing table and L,M set of a node     e.g.   c3333 |
|    r: Route a message starting from some node           e.g.   r2312 |
|    q: Quit                                              e.g.   q     |
|----------------------------------------------------------------------|
r1230
Input key:
2312
Routing path is as follows:
1230 --> 2313 --> 3201
```

Command for routing

Targeted Key

The resulted routing path

```
|----------------------------------------------------------------------|
|                        Welcome to Pastry System                      |
|       Assignment 1 for CSIS8301 Advanced Topics in Computer Systems  |
|     Developed by ZHANG Chenggang (U no.: 2009950026), CS Department, HKU |
|----------------------------------------------------------------------|

Please choose the working mode: 0 for USER mode, 1 for EXPERIMENT mode.
1

Currently there are 2 protocols implemented, please choose one to work with:
   0: Improved Pastry Protocol (Highly Recommended!)
   1: Original Pastry Protocol
0
The EXPERIMENT mode is designed to test Pastry Protocol's performance on various aspects.
The default setting is: base = 4, an ID has 4 quaternary digits.

Please input the size of the leaf set, |L| (needs to be an even number like 2,4,6...) :
4
Please input the size of the neighborhood set, |M|:
4
```

Working mode

Protocol ID

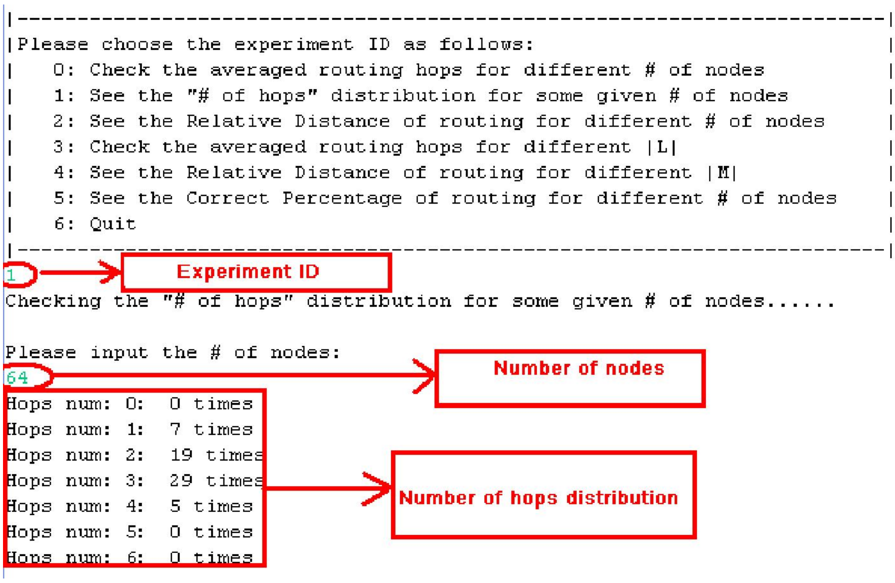
Number of |L|

Number of |M|

```
|----------------------------------------------------------------------|
|Please choose the experiment ID as follows:                           |
|    0: Check the averaged routing hops for different # of nodes       |
|    1: See the "# of hops" distribution for some given # of nodes     |
|    2: See the Relative Distance of routing for different # of nodes  |
|    3: Check the averaged routing hops for different |L|              |
|    4: See the Relative Distance of routing for different |M|         |
|    5: See the Correct Percentage of routing for different # of nodes |
|    6: Quit                                                           |
|----------------------------------------------------------------------|
0
Checking the averaged routing hops for different # of nodes......
****************************************************
numOfNodes = 2; Averaged num of routing hops: = 0
numOfNodes = 4; Averaged num of routing hops: = 0
numOfNodes = 8; Averaged num of routing hops: = 1
numOfNodes = 16; Averaged num of routing hops: = 1
numOfNodes = 32; Averaged num of routing hops: = 2
numOfNodes = 64; Averaged num of routing hops: = 2
numOfNodes = 128; Averaged num of routing hops: = 2
numOfNodes = 256; Averaged num of routing hops: = 2
****************************************************
```

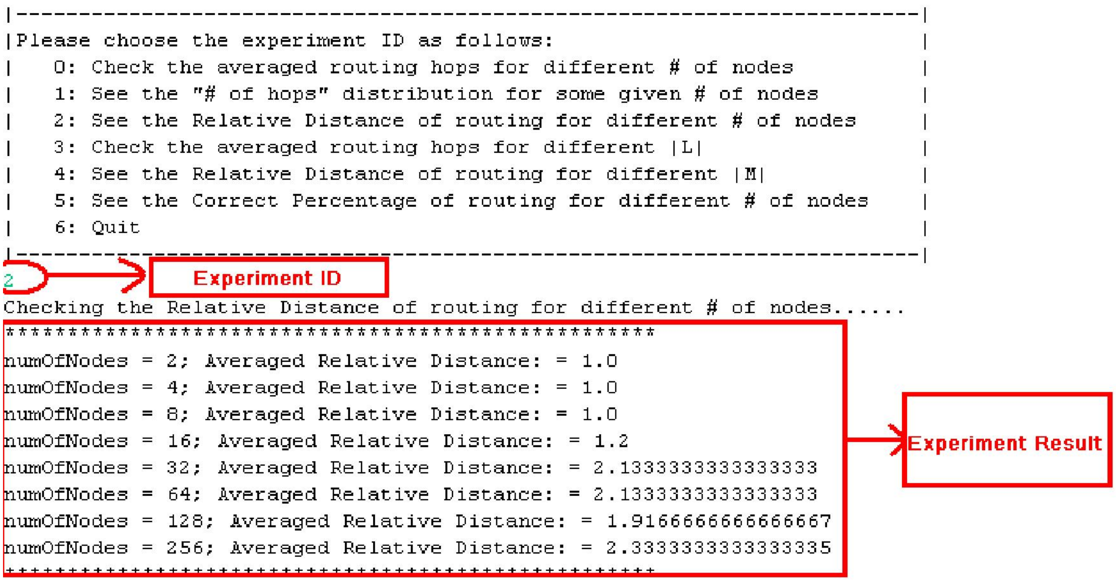
Experiment ID

Experiment Result



```
|---------------------------------------------------------------------|
|Please choose the experiment ID as follows:                          |
|    0: Check the averaged routing hops for different # of nodes      |
|    1: See the "# of hops" distribution for some given # of nodes    |
|    2: See the Relative Distance of routing for different # of nodes |
|    3: Check the averaged routing hops for different |L|             |
|    4: See the Relative Distance of routing for different |M|        |
|    5: See the Correct Percentage of routing for different # of nodes|
|    6: Quit                                                          |
|---------------------------------------------------------------------|
1
Checking the "# of hops" distribution for some given # of nodes......

Please input the # of nodes:
64
Hops num: 0:  0 times
Hops num: 1:  7 times
Hops num: 2:  19 times
Hops num: 3:  29 times
Hops num: 4:  5 times
Hops num: 5:  0 times
Hops num: 6:  0 times
```

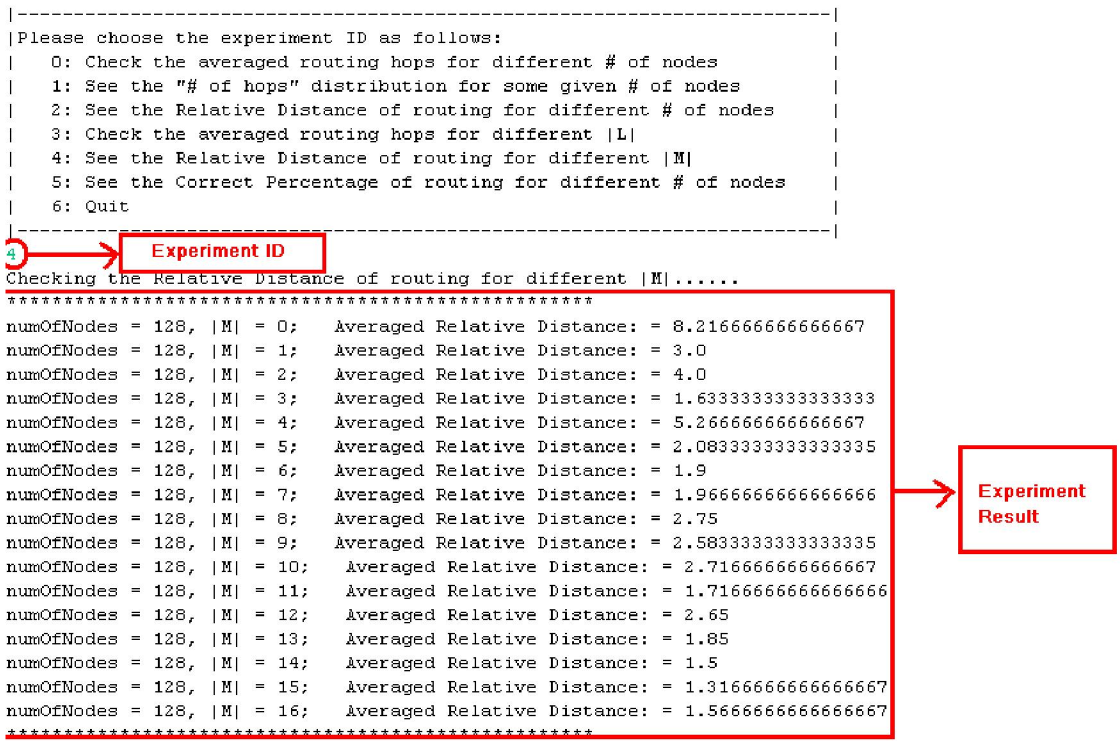



```
|---------------------------------------------------------------------|
|Please choose the experiment ID as follows:                          |
|    0: Check the averaged routing hops for different # of nodes      |
|    1: See the "# of hops" distribution for some given # of nodes    |
|    2: See the Relative Distance of routing for different # of nodes |
|    3: Check the averaged routing hops for different |L|             |
|    4: See the Relative Distance of routing for different |M|        |
|    5: See the Correct Percentage of routing for different # of nodes|
|    6: Quit                                                          |
|---------------------------------------------------------------------|
2
Checking the Relative Distance of routing for different # of nodes......
****************************************************
numOfNodes = 2; Averaged Relative Distance: = 1.0
numOfNodes = 4; Averaged Relative Distance: = 1.0
numOfNodes = 8; Averaged Relative Distance: = 1.0
numOfNodes = 16; Averaged Relative Distance: = 1.2
numOfNodes = 32; Averaged Relative Distance: = 2.1333333333333333
numOfNodes = 64; Averaged Relative Distance: = 2.1333333333333333
numOfNodes = 128; Averaged Relative Distance: = 1.9166666666666667
numOfNodes = 256; Averaged Relative Distance: = 2.3333333333333335
****************************************************
```

```
|-----------------------------------------------------------------------|
|Please choose the experiment ID as follows:                            |
|    0: Check the averaged routing hops for different # of nodes        |
|    1: See the "# of hops" distribution for some given # of nodes      |
|    2: See the Relative Distance of routing for different # of nodes   |
|    3: Check the averaged routing hops for different |L|               |
|    4: See the Relative Distance of routing for different |M|          |
|    5: See the Correct Percentage of routing for different # of nodes  |
|    6: Quit                                                            |
|-----------------------------------------------------------------------|
3
Checking the averaged routing hops for different |L| ......
***************************************************
Number of nodes = 256, |L| = 2;  Averaged num of routing hops: = 3
Number of nodes = 256, |L| = 4;  Averaged num of routing hops: = 2
Number of nodes = 256, |L| = 8;  Averaged num of routing hops: = 2
Number of nodes = 256, |L| = 16;  Averaged num of routing hops: = 2
***************************************************
```

Experiment ID

Experiment Result



```
|-----------------------------------------------------------------------|
|Please choose the experiment ID as follows:                            |
|    0: Check the averaged routing hops for different # of nodes        |
|    1: See the "# of hops" distribution for some given # of nodes      |
|    2: See the Relative Distance of routing for different # of nodes   |
|    3: Check the averaged routing hops for different |L|               |
|    4: See the Relative Distance of routing for different |M|          |
|    5: See the Correct Percentage of routing for different # of nodes  |
|    6: Quit                                                            |
|-----------------------------------------------------------------------|
4
Checking the Relative Distance of routing for different |M|......
*****************************************************
numOfNodes = 128, |M| = 0;   Averaged Relative Distance: = 8.216666666666667
numOfNodes = 128, |M| = 1;   Averaged Relative Distance: = 3.0
numOfNodes = 128, |M| = 2;   Averaged Relative Distance: = 4.0
numOfNodes = 128, |M| = 3;   Averaged Relative Distance: = 1.6333333333333333
numOfNodes = 128, |M| = 4;   Averaged Relative Distance: = 5.266666666666667
numOfNodes = 128, |M| = 5;   Averaged Relative Distance: = 2.0833333333333335
numOfNodes = 128, |M| = 6;   Averaged Relative Distance: = 1.9
numOfNodes = 128, |M| = 7;   Averaged Relative Distance: = 1.9666666666666666
numOfNodes = 128, |M| = 8;   Averaged Relative Distance: = 2.75
numOfNodes = 128, |M| = 9;   Averaged Relative Distance: = 2.5833333333333335
numOfNodes = 128, |M| = 10;   Averaged Relative Distance: = 2.716666666666667
numOfNodes = 128, |M| = 11;   Averaged Relative Distance: = 1.7166666666666666
numOfNodes = 128, |M| = 12;   Averaged Relative Distance: = 2.65
numOfNodes = 128, |M| = 13;   Averaged Relative Distance: = 1.85
numOfNodes = 128, |M| = 14;   Averaged Relative Distance: = 1.5
numOfNodes = 128, |M| = 15;   Averaged Relative Distance: = 1.3166666666666667
numOfNodes = 128, |M| = 16;   Averaged Relative Distance: = 1.5666666666666667
*****************************************************
```


3. Special notes

Currently two versions of Pastry protocols have been implemented: the original Pastry protocol and the improved Pastry protocol. **The improved Pastry protocol** is relatively better tested than the **original Pastry protocol**, so there may be some bugs in the original Pastry protocol implementation.

The **Original Pastry Protocol** works as follows when constructing a new node X's Routing table, Leaf set and Neighborhood set (will be referred to as R table, L set and M set respectively): Define the new node as X, and the physically nearest node in Pastry Network is A. Then A routes the "join" message with X's ID. Suppose the routing path is A à B à …… à Z, and Z is the numerically nearest node to the new node X. To construct the R table, L set and M set, X copies the routing path's i^{th} node's i^{th} row to its i^{th} row, copies Z's L set to its L set, and uses A's M set as its own M set.

The **Improved Pastry Protocol** works as follows when constructing a new node X's Routing table, Leaf set and Neighborhood set (will be referred to as R table, L set and M set respectively): Define the new node as X, and the physically nearest node in Pastry Network is A. Then A routes the "join" message with X's ID. Suppose the routing path is A à B à …… à Z, and Z is the numerically nearest node to the new node X. To construct the R table, L set and M set, instead of just copying the routing path's i^{th} node's i^{th} row to its i^{th} row, copying Z's L set to its L set, and using A's M set as its own M set, X considers all the nodes along the routing path and uses them as currently known live nodes in the Pastry Network to construct its own R table, L set and M set. (More details can be found in the following methods:

`constructRTable()`, `constructLSet()` and `constructMSet()` in the ***Pastry.java*** in the source code directory)

Also, two working modes are provided. One is **USER mode**, which is designed for the user to use and test Pastry. The other is **EXPERIMENT mode,** which is used for testing the performance of Pastry Protocol.

If you have any questions about the program itself and how to compile and run it, do not hesitate to contact with me. Thank you!

MSN: chenggangschool@hotmail.com
E-mail: cgzhang@cs.hku.hk