CST8130: Data Structures

Assignment #2- Router Simulator V2

Test Plan

Mark Kaganovsky

040-789-903

Section #302 (8:00am to 10:00am Monday)

Packet Processing in the Router class.

|  |  |  |
| --- | --- | --- |
| **Condition Description** | **Sample** | **Result** |
| New packet Needs to be inserted into the routing table. | p 1 2 3 4 24 0 0 0 0 24 e1  p 3 1 1 1 24 0 0 0 0 24 e3  p 2 1 1 12 24 0 0 0 0 24 e2  p 0 1 1 12 24 0 0 0 0 24 e0  p 0 3 1 12 24 0 0 0 0 24 e0  p 0 4 1 12 24 0 0 0 0 24 e0 | Packet table is printed out like this at the end:  *+---------------------------------+*  *| Routing Table |*  *+---------------------+-----------+*  *| Network IP | Port |*  *+---------------------+-----------+*  *| 0.1.1.0/24 | e0 |*  *| 0.3.1.0/24 | e0 |*  *| 0.4.1.0/24 | e0 |*  *| 1.2.3.0/24 | e1 |*  *| 2.1.1.0/24 | e2 |*  *| 3.1.1.0/24 | e3 |*  *+---------------------+-----------+* |
| New packet needs to be routed out of a port. | p 1 2 3 4 24 0 0 0 0 24 e1  p 3 1 1 1 24 0 0 0 0 24 e3  p 2 1 1 12 24 0 0 0 0 24 e2  p 0 1 1 12 24 0 0 0 0 24 e0  p 0 3 1 12 24 0 0 0 0 24 e0  p 0 4 1 12 24 0 0 0 0 24 e0  d 3 1 1 1 24 0 0 0 0 24 hello | When the final packets are encountered, the following text is displayed right before the routing table is displayed:  *…*  *Data to 3.1.1.1/24 on network 3.1.1.0/24 is being sent out port e3*  *Data to 1.2.3.4/24 on network 1.2.3.0/24 is being sent out port e1*  *…* |

IPAddress.compareTo(IPAddress ip) method.

|  |  |  |
| --- | --- | --- |
| **Condition Description** | **Sample** | **Result** |
| Same subnet.  The ip address passed as an argument to this function is greater than the IPAddress object which the compareTo() method was called on. | Code:  IPAddress ipSmall = new IPAddress();  IPAddress ipLarge = new IPAddress();  ipLarge.initialize(new int[]{192, 168, 0, 1}, 24);  ipSmall.initialize(new int[]{8, 7, 56, 125}, 24);  System.out.println(ipLarge.getNetwork().compareTo(ipSmall.getNetwork())); | “1” should be printed to the console. |
| Same subnet.  The ip address passed as an argument to this function is less than the IPAddress object which the compareTo() method was called on. | Code:  IPAddress ipSmall = new IPAddress();  IPAddress ipLarge = new IPAddress();  ipLarge.initialize(new int[]{192, 168, 0, 1}, 24);  ipSmall.initialize(new int[]{8, 7, 56, 125}, 24);  System.out.println(ipSmall.getNetwork().compareTo(ipLarge.getNetwork())); | “-1” should be printed to the console |
| Same subnet.  The ip address passed as an argument to this function is equal to the IPAddress object which the compareTo() method was called on. | Code:  IPAddress ipSmall = new IPAddress();  IPAddress ipLarge = new IPAddress();  ipLarge.initialize(new int[]{192, 168, 0, 1}, 24);  ipSmall.initialize(new int[]{192, 168, 0, 1}, 24);  System.out.println(ipSmall.getNetwork().compareTo(ipLarge.getNetwork())); | “0” should be printed to the console |
| Different subnet.  The ip address passed as an argument to this function has a subnet smaller than the IPAddress object which the compareTo() method was called on. | Code:  IPAddress ipSmall = new IPAddress();  IPAddress ipLarge = new IPAddress();  ipLarge.initialize(new int[]{192, 168, 0, 1}, 24);  ipSmall.initialize(new int[]{192, 168, 0, 1}, 8);  System.out.println(ipSmall.getNetwork().compareTo(ipLarge.getNetwork())); | “-1” should be printed to the console |
| Different subnet.  The ip address passed as an argument to this function has a subnet greater than the IPAddress object which the compareTo() method was called on. | Code:  IPAddress ipSmall = new IPAddress();  IPAddress ipLarge = new IPAddress();  ipLarge.initialize(new int[]{192, 168, 0, 1}, 16);  ipSmall.initialize(new int[]{192, 168, 0, 1}, 24);  System.out.println(ipSmall.getNetwork().compareTo(ipLarge.getNetwork())); | “1” should be printed to the console |

RoutingTableEntry.compareTo(IPAddress ip) method.

There is not much to test in this method since all it does is return the value from calling IPAddress.compareTo():

*return this.destination.compareTo(ip);*