CST8130: Data Structures

Assignment #5- Routing Table

Test Plan

Mark Kaganovsky

040-789-903

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

The hashtable in the Router class.

|  |  |  |
| --- | --- | --- |
| **Condition Description** | **Sample** | **Result** |
| Does the method processPackets() add new Routing packets in the correct position? | Sample file:  p 192 168 1 2 24 192 168 1 4 24 e0  p 192 168 4 6 24 192 168 3 2 24 e1  p 192 168 3 8 24 11 0 4 2 8 s0 | Assuming a size of 256 for the routing table array.  Going line by line:  Added at index 2  Added at index 6  Added at index 3 |
| Does the method processPackets() route data packets out the correct port? | Adding to the above file:  d 192 168 1 2 24 192 168 1 4 24 123456778123  d 192 168 4 1 24 192 168 3 2 24 aaaaaa  d 192 168 3 3 24 192 168 3 2 24 bbbbb  d 192 168 5 5 16 192 168 3 2 24 abcde | Going line by line  Routed out port e0  Routed out port e1  Routed out port s0  Dropped |
| Does the method processPackets() drop routing packets which cause a hash collision? | Sample file:  p 192 168 1 2 24 192 168 1 4 24 e0  p 192 168 1 3 24 192 168 3 2 24 e1 | The first packet gets added to the routing table at index 1  The second packet causes a hash collision with the first and is therefore dropped. |

Hashing an IPAddress

The hashing done in this program is very trivial. When **Packet.hashCode()** is called, it calls the **IPAddress.hashCode()** method on its destination ip and returns that value. All **IPAddress.hashCode()** does is return the element at index 2 of the ip address array.