Team 2 Documentation

# Implementation

## Initialization:

**Step 1:** Populate the distance table for each entity, giving each node that is not a neighbor the value 999.

**Example:** Distance Table Initialized in Entity 0 (to I via neighbor J)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| J I -> | 0 | 1 | 2 | 3 |
| 0 | 0 | 1 | 3 | 7 |
| 1 | 999 | 0 | 999 | 999 |
| 2 | 999 | 999 | 0 | 999 |
| 3 | 999 | 999 | 999 | 0 |

**Step 2:** Print the distance table with the provided method printDT()

**Step 3:** Store the minimum costs from the given entity into a new array

**Step 4:** For each neighbor of the entity, send a packet to it with the minimum cost array

## Update:

**Step 1:** Identify who the source of the packet being sent to update is.

**Step 2:** Update the current entities distance table for the packets source with the new minimum costs sent to it in the packet.

**Step 3:** For every entity, test whether the distance to the source of the packet plus the distance from the source to the entity is less than the current distance to that entity in the distance table. If so update the distance table to reflect the shorter distance.

**Step 4:** If the distance table was updated in step 3, then send the current entities distance table in a packet to all neighbors so they can update their distance tables.

Flow Chart Example for Node 0:

|  |
| --- |
| Step1: Initializer Entity0.java  Contains **distanceTable** 2D Array.  **minCost**: {0,1,3,7} for nodes 0,1,2,and 3 respectively.  **Neighbors:** Nodes 1,2,3.  **Create Packets for Each Neighbor:** Create packet objects for Nodes 1,2, and 3. Each packet contains mincost cost array. Send packet to each neighbor |

NetworkSimulator.toLayer2(packet)

|  |
| --- |
| Entity0.java  **update(Packet p)** is called. {   * Received a packet from another Entity. * For this example on Entity0, Entity 2 sends packet p. P contains source=2, destination=0, and mincost array from source=2. * Update the entities distance tables from neighbors receiving packet.     **IF** (distance table of entity 0 > distance table from source of p + distance from source to entity ) {   * Set entity0 distance table = distance table from source of p + distance from source to 0, * Send updated distance table from Entity0 by Calling NetworkSimulator.toLayer2(packet p) again to all of Entity 0 neighbors nodes. Other Entities will follow similar steps as Entity0, until every entity has finished updating the least cost path for each entities distance tables.     **}else** {  Finished update on Entity.  } |

# Dev Environment

* IDE: Eclipse
* To compile and run in Eclipse, go to the file Project.java and click run.

# Team Member Contribution

Brian Canela: 33%

Francisco Gudino: 33%

Mohammed Kuko: 33%