**Computer Science 4457A – Network II**

**2017-18 Fall Semester**

**Group 3**

**Jameel Kaba**

**Pearson Radu**

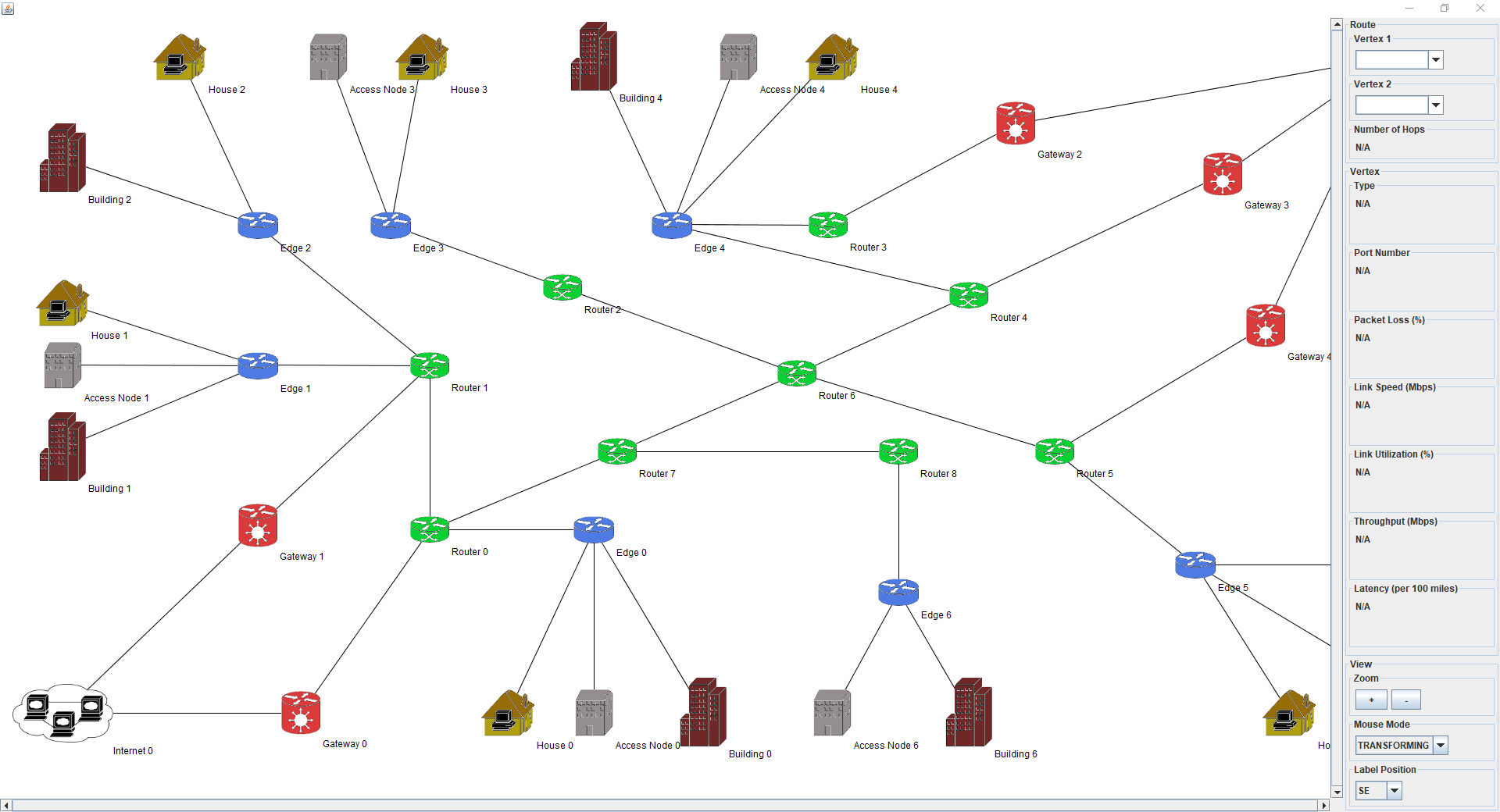
**Perdro Ortega**

**ProactiveQoS User Guide**

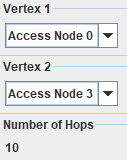
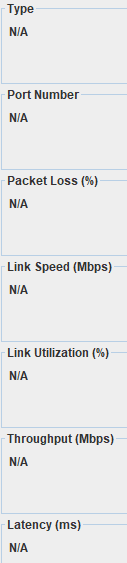
**Requirements**

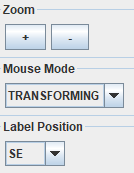
* Java 8 or higher

**Running the Application**

* Double-click the **ProactiveQoS.jar** file
* Alternatively, you can use the command **java -jar <path>/ProactiveQoS.jar**
* Once you have started the application you will be greeted by a network topology
* 
* **Note** this is a fixed topology and we did not reach the stage of the user being able to design their own topology. JUNG supported a framework called graphml, which is a commonly used graph designer that follows an xml format, but unfortunately integration for fixed positioning was non-existent and all vertices would pile on top of one another. This would make the feature useless as the user would have to manually move all vertices so it was left out.

**Navigation Pane**

* On the right-hand side is the Navigation Pane
* It consists of three main sections including **Route**, **Vertex**, and **View**
* **Route**
  + The route section contains three subsections
    - **Vertex 1**
      * Drop down menu to select a starting vertex for a path
    - **Vertex 2**
      * Drop down menu to select an ending vertex for a path
    - **Number of Hops**
      * Once both vertices have been selected it will tell the user the number of hops taken on that path
* **Vertex**
  + The vertex section contains seven subsections
    - **Type**
      * The type and name of the vertex
    - **Port Number**
      * The port number the node is using
    - **Packet Loss (%)\***
      * The packet loss percentage of the node
    - **Link Speed (Mbps)**
      * The link speed of the node in mbps
    - **Link Utilization (%)\***
      * The link utilization percentage of the node
    - **Throughout (Mbps)**
      * The throughput of the node in mbps
    - **Latency (ms)**
      * The latency in milliseconds per node
    - \* percentages are shown in decimal form, the actual percentage is what is displayed multiplied by 100

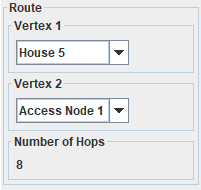
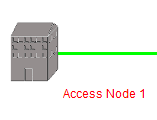
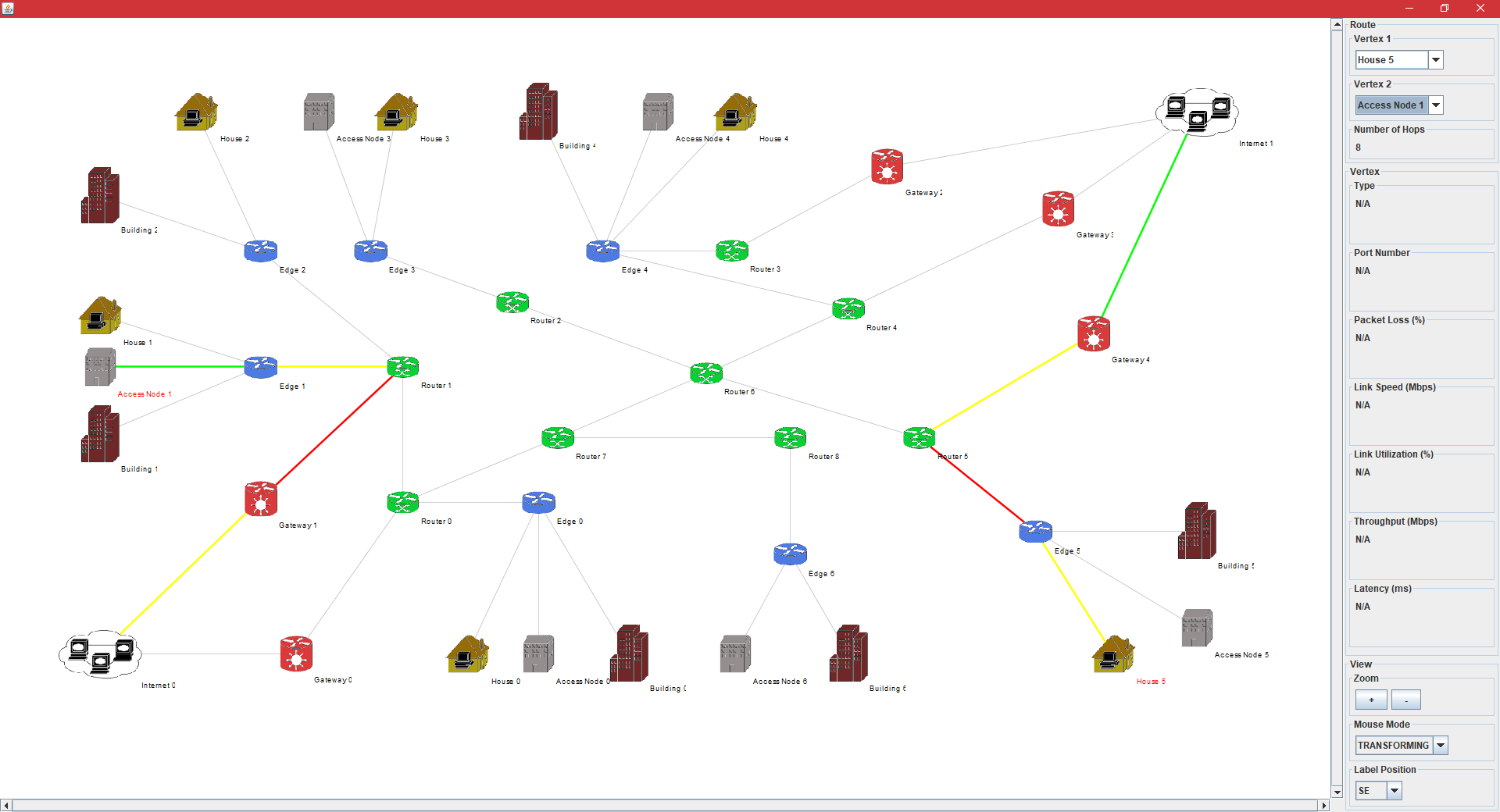


* **View**
  + The view section contains three subsections
    - **Zoom**
      * The user can choose to scroll with the mouse wheel over the topology or use the “+” and “-” buttons to adjust the zoom of the graph
    - **Mouse Mode**
      * Drop down menu to select interaction type with the topology
      * **Transforming** (default)
        + Allows the user to drag the graph to different locations
      * **Picking**
        + Allows the user to select a specific node to view information in the **Vertex** section or move nodes to different locations
    - **Label Position**
      * The user can adjust the compass direction of a label for easier visualization

**Node Types**

* **Access Node 🡪 C:\Users\Pearson Radu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\AccessNode.png**
* **Building 🡪 C:\Users\Pearson Radu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Building.png**
* **Edge 🡪 C:\Users\Pearson Radu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Edge.png**
* **Gateway 🡪 C:\Users\Pearson Radu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Gateway.png**
* **House 🡪 C:\Users\Pearson Radu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\House.png**
* **Internet 🡪 **
* **Router 🡪 C:\Users\Pearson Radu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Router.png**

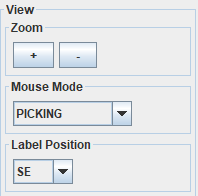
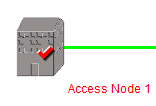
**Graph Results**

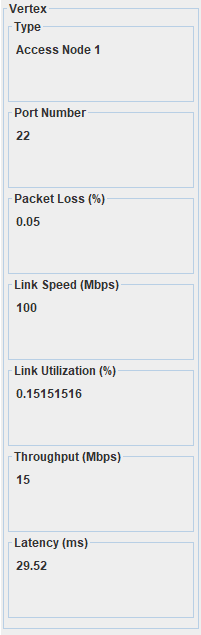
* As mentioned before the user must select a starting and ending vertex within the **Route** section in the **Navigation Pane**
  + Example, we want to measure the network between **House 5** and **Access Node 1**
  + 
  + To assure the user they have selected the correct vertices the label on the chosen vertices will change to red
    -  
* Once a route has been chosen the graph will update automatically
* 
* The shortest route to the nearest Internet source will be provided to the user
* **Note** we have not created a path the takes all network parameters in account. JUNG provided a lot of trouble with manipulating the highlighted path and being restricted with time we were not able to successfully modify our shortest path algorithm.

**Graph Edge Status**

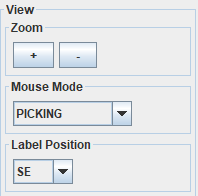
* As you can see in the above example there are four different edge colors between the nodes
  + GREY
    - Grey means that the edge is not active given the inputted route
  + GREEN
    - Green means the edge is active and the status of the path is good
  + YELLOW
    - Yellow means the edge is active and the status of the path is cautious
  + RED
    - Red means the edge is active and the status of the path is unstable

**Node Status**

* To view the status of an individual node the user can go to the **View** section in the **Navigation Pane** and change the **Mouse Mode** to **Picking**
  + 
* The user can then click on a single node within the path
  + 
  + Note when the user does this a check mark will appear on the node
* Once a node has been chosen the **Vertex** section in the **Navigation Pane** will update automatically



**User Selected Path**

* The user can select their own path within the topology but unfortunately, we did not have enough time to implement the functionality of calculating the path
* To do so the user can go to the **View** section in the **Navigation Pane** and change the **Mouse Mode** to **Picking**
  + 
* The user can then hold down the “SHIFT” key and continue to click on node at a time with the mouse
* For example
* 