# DOGE SAS

## Definitions

1. Path/Route - A collection of nodes and edges through which a single packet reaches its destination.
2. Single path routing– A single path exists in a routing table between a source and destination node.
3. Multipath routing – Multiple paths may exist in a routing table between a source and destination node.
4. Routing processor - Edison

# Web Interface/Routing Processor Requirements

1. By default, the Web UI shall display all nodes and neighbor edges present in the network.
2. The Web UI shall present the user with GUI element(s) to visibly hide nodes and neighbor edges in the network graph.
   1. These element should have no impact on the network topology. They are just there to aid in network visualization.
3. When nodes and edges are visible, the Web UI shall present the user with GUI element(s) to mask nodes and neighbor edges in the network graph.
4. When nodes and edges are visible, the Web UI shall present the user with GUI element(s) to delete nodes and neighbor edges in the network graph.
5. When nodes and edges are visible, the Web UI shall present the user with GUI element(s) to add nodes and neighbor edges to the network graph.
6. When masked nodes and masked edges are visible, the Web UI shall present the user with GUI element(s) to unmask nodes and neighbor edges in the network graph.
7. When nodes and edges are visible, the Web UI shall display the LQE value on each neighbor edge in the network.
8. When nodes are visible, the Web UI shall provide a mechanism for the user to review the HW configuration of each node.
9. When nodes are visible, the Web UI shall provide a mechanism for a user to select one or two nodes.
10. When two nodes in the network graph are selected, all routes between the two nodes shall be presented to the user.
11. When routes between two nodes are presented to a user, the user shall be able to select any of these routes.
12. When the Web UI is initialized, the routing processor shall initiate network topology discovery through any algorithm that can provide the following functions. See Appendix for a sample algorithm.
    1. Find all nodes in the network.
    2. Update the routing table for every node in the network.
13. After the routing processor completes a network element removal or addition request, the routing processor shall send the updated network state (nodes, neighbor edges, routes) to the Web UI.
14. The routing processor shall check every network element removal request to prevent creation of orphan nodes.
15. When a user has selected a route between two nodes and requested its deletion, the routing processor shall:
    1. Check the routing impact of deleting this route by calculating the number of affected of “sub” routes.
       1. If there are no “sub” routes using this route, the routing processor shall delete the route.
       2. If there are 1 or more “sub” routes using this route, the routing processor shall attempt to recalculate replacement routes for every source, destination pair in the affected “sub” routes.
          1. TODO: Add orphan check. If there are no routes that can replace the affected routes, the routing processor shall halt the route deletion process, request that the Web UI prompt the user for confirmation, and then continue with route deletion once the operation is confirmed by the user.
          2. If there are routes that can replace the affected routes, the routing processor shall add the new routes by updating the relevant node routing tables and then proceed with the requested route deletion.
    2. Delete all “sub” routes contained within the requested route.
16. When a user has selected two nodes and requested that a route be added to connect them, the routing processor shall:
    1. Calculate a route between the two nodes.
    2. Check the feasibility of adding this route by analyzing its component nodes.
       1. If there is no space available in the node routing tables along the route, the routing processor shall cancel the route addition process and request that the Web UI report an error to the user.
       2. If there is space available in the node routing tables along the route, the routing processor shall add the route by updating the relevant node routing tables.
17. When a user masks a neighbor edge in the Web UI, the routing processor shall:
    1. Check the routing impact of masking this edge by calculating the number of affected of routes.
       1. If there are no routes using this neighbor edge, then the routing processor shall proceed with edge masking.
       2. If there are 1 or more routes using this neighbor edge, the routing processor shall attempt to recalculate replacement routes which do not include the masked neighbor edge.
          1. TODO: Add orphan check. If there are no routes that can replace the affected routes, the routing processor shall halt the edge masking process, request that the Web UI prompt the user for confirmation, and then continue with edge masking once the operation is confirmed by the user.
          2. If there are routes that can replace the affected routes, the routing processor shall add the new routes by updating the relevant node routing tables and then proceed with edge masking.
    2. Invalidate the neighbor edge by adjusting the single-hop LQE in the relevant node neighbor tables.
    3. Invalidate all routes using this neighbor edge by adjusting the multi-hop LQE in the relevant node routing tables.
18. When a user deletes a neighbor edge in the Web UI, the routing processor shall:
    1. Check the routing impact of deleting this edge by calculating the number of affected of routes.
       1. If there are no routes using this neighbor edge, then the routing processor shall proceed with edge deletion.
       2. If there are 1 or more routes using this neighbor edge, the routing processor shall attempt to recalculate replacement routes which do not include the deleted neighbor edge.
          1. TODO: Add orphan check. If there are no routes that can replace the affected routes, the routing processor shall halt the edge deletion process, request that the Web UI prompt the user for confirmation, and then continue with edge deletion once the operation is confirmed by the user.
          2. If there are routes that can replace the affected routes, the routing processor shall add the new routes by updating the relevant node routing tables and then proceed with edge deletion.
    2. Remove the nodes connected by the edge from node neighbor tables.
    3. Remove any routes present in all routing tables that uses this edge for multi-hop communication.
19. When a user masks a node in the Web UI, the routing processor shall:
    1. Check the routing impact of masking this node by calculating the number of affected of routes.
       1. If there are no routes using this node, then the routing processor shall proceed with node masking.
       2. If there are 1 or more routes using this node, the routing processor shall recalculate replacement routes which do not include the masked node.
          1. If there are no routes that can replace the affected routes and the node masking will not create orphan nodes, the routing processor shall halt the node masking process, prompt the user for confirmation, and then continue with node masking once the operation is confirmed by the user.
          2. If there are no routes that can replace the affected routes and the node masking will create orphan nodes, the routing processor shall cancel the node masking operation, and request that the Web UI report an error to the user.
          3. If there are routes that can replace the affected routes, the routing processor shall add the new routes by updating the relevant node routing tables and then proceed with node masking.
       3. Invalidate the node in all neighbor tables by adjusting the single-hop LQE.
       4. Invalidate all routes in the network that use this node by adjusting the multi-hop LQE.
20. When a user deletes a node in the Web UI, the routing processor shall:
    1. Check the routing impact of deleting this node by calculating the number of affected of routes.
       1. If there are no routes using this node, then the routing processor shall proceed with node deletion.
       2. If there are 1 or more routes using this node, the routing processor shall recalculate replacement routes which do not include the deleted node.
          1. If there are no routes that can replace the affected routes and the node deletion will not create orphan nodes, the routing processor shall halt the node deletion process, prompt the user for confirmation, and then continue with node deletion once the operation is confirmed by the user.
          2. If there are no routes that can replace the affected routes and the node deletion will create orphan nodes, the routing processor shall cancel the node deletion operation, and request that the Web UI report an error to the user.
          3. If there are routes that can replace the affected routes, the routing processor shall add the new routes by updating the relevant node routing tables and then proceed with node deletion.
       3. Invalidate the node in all neighbor tables by adjusting the single-hop LQE.
       4. Delete any routes in the network that use this node.
21. When a user unmasks a previously masked node to the network, the routing processor shall:
    1. Reset the invalid single-hop LQE for the masked node’s neighboring nodes.
    2. Reset the invalid multi-hop LQE for all routes in the network containing this node.
    3. Send a message to the previously masked node in order to update the single-hop and multi-hop LQEs.
22. When a user unmasks a previously masked edge to the network, the routing processor shall:
    1. Reset the invalid single-hop LQE in the relevant node neighbor tables.
    2. Reset the invalid multi-hop LQE for all routes using this neighbor edge.
    3. Find all routes using this neighbor edge, and initiate one or more messages along these routes until their multi-hop LQEs are set to normal values.
23. ~~When a user adds a previously deleted node to the network, the routing processor shall:~~
    1. ~~Request a user confirmation from the WebUI to initiate full route recalculations.~~
       1. ~~If the user accepts, the routing processor shall recalculate all routes between every two nodes in the network and perform the following steps.~~
          1. ~~The routing processor shall add non-existent routes to all routing tables.~~
          2. ~~If there are new routes available with a better multi-hop LQE than existing routes, then the routing processor shall replace the worst routes with the newly computed ones.~~
       2. ~~Reset the invalid LQE links for neighboring nodes, and send a message to the new node in order for LQE to be reset to a normal value.~~

# Appendix

## Routing Processor Algorithm for populating the network graph

**TBD. Edison has to add routes incrementally as it builds the network graph, and then it must replace routes on the fly with more optimal routes.**