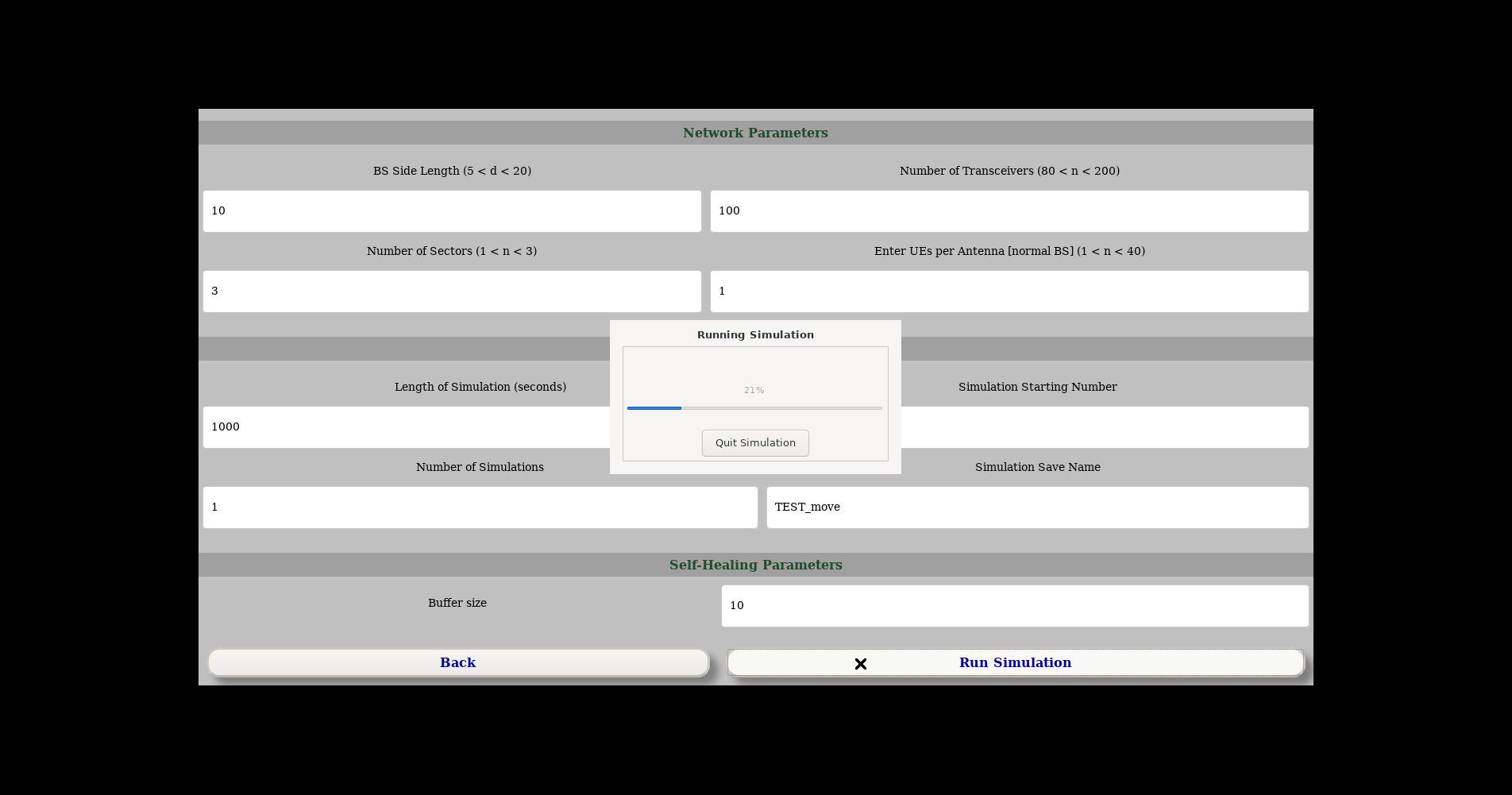
Merged changes between all branches

Features on new main:

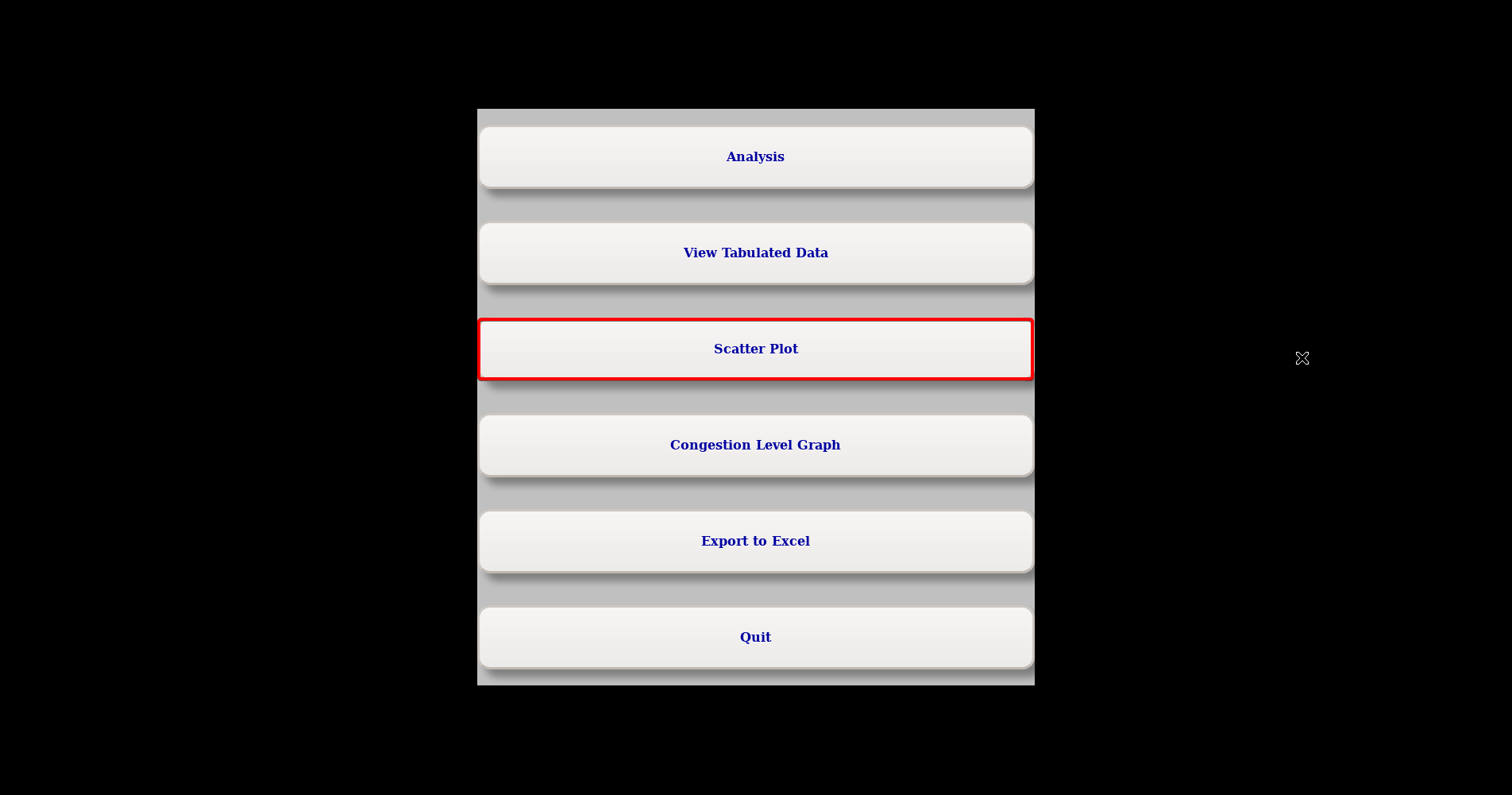
* GUI Implementation
* User Mobility Implementation

Showcase:

Loading bar



ScatterPlot



Scatterplot to help visually understand what is happening.

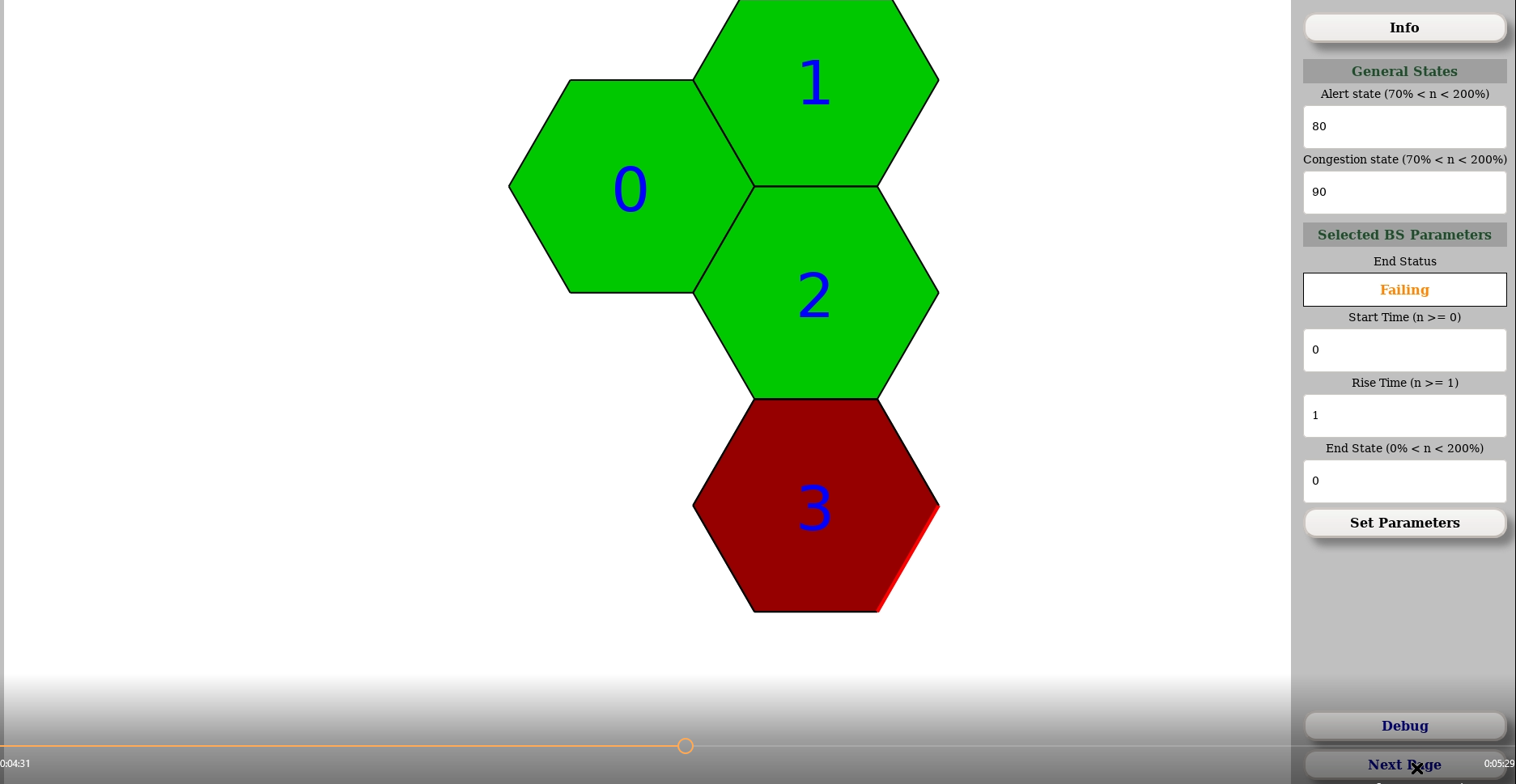
Current uses:

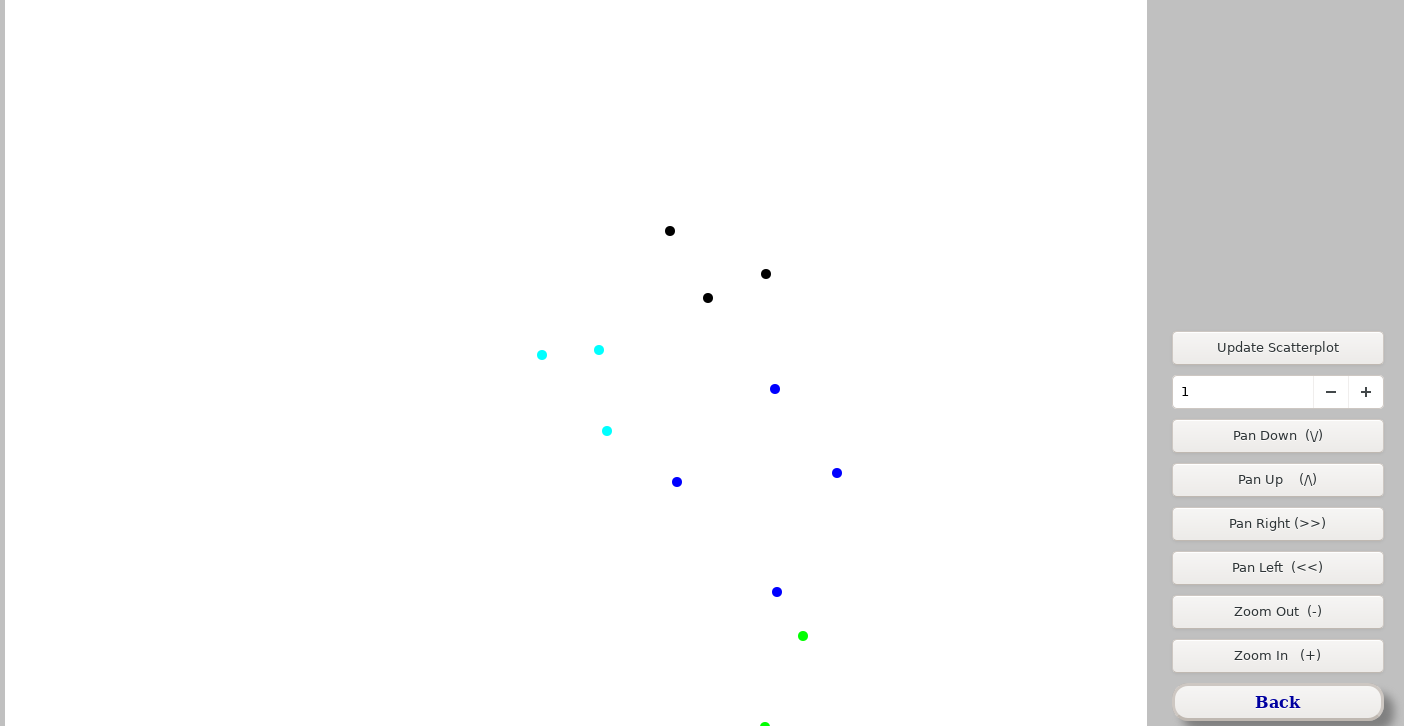
* Keep track of user’s location and which base station they are connected to

Future ideas:

* Toggling of lines drawn between user and their connected base station (instead of relying on color only)
* Hover over user to see more information (SNR, Individual KPIs, Bandwidth, etc.)
* Draw hexagons based on the base station regions to visually show range
* Add grid + legend to reference real life distance units (meters probably)

Example: Initial state when there is 1 user assigned per antenna and each base station has 3 antennas (sectors)



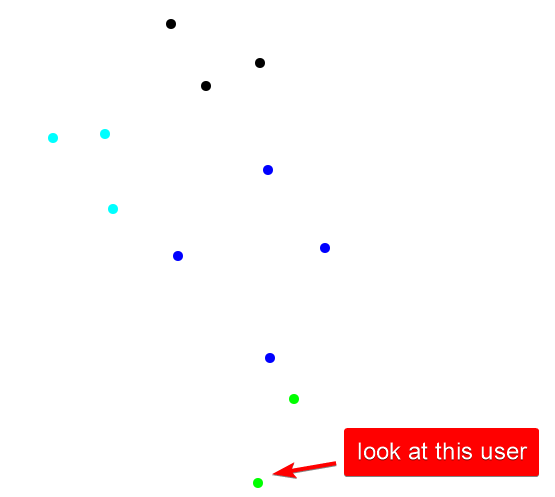


Color = Which base station they are connected to

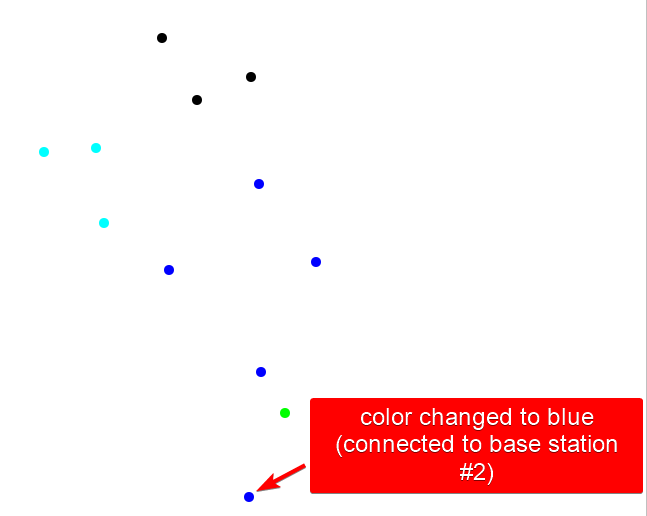
Dot location = Where the user’s REAL location is

|  |  |
| --- | --- |
| Base Station # | Color |
| 0 | Cyan |
| 1 | Black |
| 2 | Blue |
| 3 | Green |

Time = 10 (before self-healing since buffer = 10)



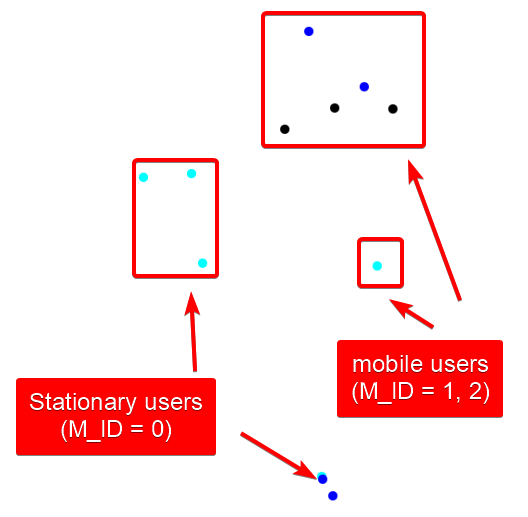
Time = 11 (Self-healing occurs since buffer = 10)



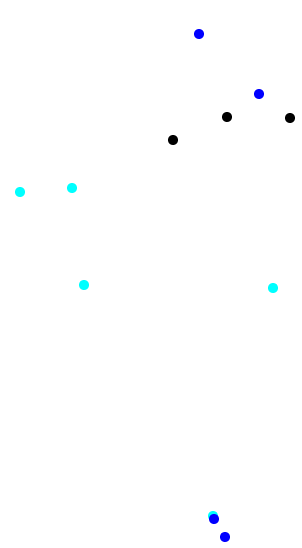
Note that all users on base station 3 (green) are now connected to different base stations because base station 3 has failed. Therefore, the offloading has already occurred, and they are now connected to a healthy base station.

|  |  |
| --- | --- |
| Time = 599 (Before user mobility has occurred) | Time = 600 (User mobility has shuffled location of users based on their mobility ID (stationary, walking, driving) |
|  |  |

Showcase of differences:



Time 600+: no changes to color of user because the self-healing algorithm only searches for users connected to “failing base stations”.



If we implemented a way for base stations to optimize their user connections, then the users would eventually be equally distributed to their closest base station to maximize connection stability and speed.