o

Self-Healing 5G Network

Design Document

Team 36
Prof.Selim
Matthew Johnson, Amber Chin, Parnumart Hanthiradej,Adam Tiedman, Yuin-Choon Ng
Team Email: sdmay21-36@iastate.edu
Team Website: https://sdmay21-36.sd.ece.iastate.edu

Revised: 11/15/20|Version 3

Executive Summary

Development Standards & Practices Used

- Software Development Standards
  - Python Coding Experience to design and implement algorithms
  - Linux Proficiency to operate testbed servers
- Networking Knowledge
  - Knowledge of wireless protocols
    - 5g
    - WiFi
    - Bluetooth
    - UDP/TCP
  - Knowledge of wireless nodes
    - WLAN
    - Radio Nodes
    - Support Servers
  - Knowledge of network testbeds
    - GENI(Wired)
    - ORBIT(Wireless)

Summary of Requirements

- The network needs to be able to:
  - Detect an outage
  - Diagnose the outage
  - Compensate for the damage while attempting a fix
  - Perform these above steps within a time efficient manner(milliseconds)
- The network should be:
  - Compatible with common network nodes
  - Modifiable to suit specific network needs
  - Usable for most 5g applications

Applicable Courses from Iowa State University Curriculum

- CprE 489
- CprE 308
- CprE 431
- CprE 430

New Skills/Knowledge acquired that was not taught in courses

- 5G network protocols
- Self-healing algorithm design
- Ruby Coding language
- Writing Linux shell scripts
- Routing ethernet networks
- Interacting with ORBIT Testbed (OMF commands)

Table of Contents

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

5G Network: The 5th generation of mobile networks. This new standard of mobile network began its launch in 2018.

GENI Platform: Testbed that uses many virtual machines to simulate a network environment

Orbit Platform: A radio grid with multiple nodes used for communication testing and development

Self-Healing: A process of diagnosing and resolving without the need for human interaction

1 Introduction

1.1 Acknowledgement

We would like to acknowledge our current advisor is Professor Selim and we would like to thank him for helping guide us through the early stages of our project. He has provided us with large amounts of background knowledge as well as helped guide our research and overall project.

1.2 Problem and Project Statement

Current cellular networks will portion out a large amount of their budget to spend on repairing and resolving network problems and outages. The current method involves relying on human expertise to identify, diagnose, and resolve any issues with the network. This process has proven to not only be very costly, but also presents significant challenges in regards to the speed at which resolutions can be made. With 5G networks rolling out, the complexity and overall cell density of these networks will prove to be too much to handle with the current processes. To be able to keep next generation networks running, a new solution for resolving network issues must be created.

The proposed solution is to create a self healing network that utilizes automation to repair damaged networks. This system can utilize intelligent algorithms to detect a full or partial outage, diagnose the cause, and compensate/fix the problem. Using an algorithm to solve these issues will allow for a network to have minimal damage during an outage while keeping the cost to maintain the network much lower than in previous generations.

1.3 Operational Environment

To fully test the self healing network, a real world simulation is needed. For the ORBIT platform, that means utilizing the outdoor testing environment. The outdoor environment will allow for testing of unexpected variables such as radio noise, propagation issues, and any other variables seen by cellular towers. For prototyping, both the indoor ORBIT platform as well as the GENI platform will be used.

1.4 Requirements

For a self healing network, there are three main requirements to be fulfilled. The network needs to be able to detect an outage, diagnose the outage, and compensate for the damage while attempting a fix. Although there are no “hard requirements” the goal of the project is to improve upon the current generation systems. This requires a detection and mitigation time under a second. Ultimately a perfect solution will leave the end user unaware that there were any issues with the network at all while being able to perform at a much lower cost of operation.

1.5 Intended Users and Uses

5G is designed to do a variety of tasks which improve our lives. For example, fast download speeds, low latency, and billions of connections to devices such as virtual reality and artificial intelligence.

Self healing 5G networks are used when there is a failure in the network. The self healing 5G network can resolve the problem without humans involved. It is a tool which can detect, remediate outages, failure and breaches.

The main user base of this is mobile networks, as this is the basis of 5g. These mobile network users should not experience any loss of service to partial outages.

1.6 Assumptions and Limitations

Assumptions:

- The end users of neighbouring base stations will not be affected by base station failure.
- The project should be working in an Indoor Radio Grid Testbed and across a small-scale open-access wireless network.
- There will be no presence of interference issues.
- The algorithm of the project must have the functionality of being able to decide which base station can be involved in the healing process.

Limitations:

- The financial cost for this project has to be none, since we are using free platforms to conduct experiments on this project.
- The software development standards for this project should be restricted to using python or ruby for coding to implement the self-healing algorithms on the testbeds and linux to operate the testbed servers.

1.7 Expected End Product and Deliverables

The expected end product in the first semester is to implement a simple self-healing algorithm between two access points with a single user.

Our final product will be an algorithm that works on a large network with many users and access points. This algorithm should be able to handle many types of network outages including single node failure as well as grid failure.

2 Project Plan

2.1 Task Decomposition

- Research Orbit Platform
    - Begin testing on simulation platform Geni
    - Run test scripts on real Orbit platform devices
- Research Self-Healing Algorithms
    - Do a research search for important self-healing algorithm descriptions
    - Break these topic down into important concepts
    - Choose desired algorithm and map out key block diagram
- Implement a self-healing algorithm on GENI
    - Code algorithm through Linux shell with simple client and server
- Implement a self-healing algorithm on Orbit platform
    - Code algorithms in Python
    - Implement algorithm on two access points
    - Implement algorithm on real-world outdoor test network
- Assess self-healing algorithm
    - Assess quality of chosen algorithm implementation
    - Decide if this algorithm meets our desired benchmarks
    - Make necessary changes to algorithm and retest
- Project Presentation
    - Prepare presentation on outcome of self-healing algorithm implementation
    - Give presentation

2.2 Risks And Risk Management/Mitigation

Implement self-healing algorithm on Orbit platform:

The Orbit platform may not work well for our implementation of a self-healing algorithm. Because this platform is well known and used for applications such as ours, the probability of this is fairly low at around .2. If this is the case we may need to select a different algorithm, or we may need to choose a different platform such as Geni.

Assess self-healing algorithm

The second risk factor is that our algorithm may not meet our expectations. This is fairly likely at about .6. If this is the case, then our plan will be to go back to the research stage and choose an algorithm with the knowledge gained through our first implementation. We would then need to go through the implementation task again, finally reaching the assess task once more. This risk can be slightly mitigated through a more lengthy research process, however, it can never be eliminated until we actually test the algorithm on the platform. We have factored into our timeline that this may occur, so the loss in time is not drastic.

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Research Orbit Platform :

The research on Orbit platform has been carried out. It takes 2 to 3 weeks to receive a response from the organization. Because of this, our team has planned to use another platform called GENI which is much similar to the Orbit platform. On GENI, team members are allowed to create their own slices (experiments) with different nodes (server or client).

Research about what self-healing is:

Our team has been doing a lot of research and reading IEEE articles on what self-healing is and also presenting powerpoint slides to our advisor to get feedback or additional information from him biweekly. We learnt that the most important thing about self-healing is that the end user should not have to experience any network failure from the base stations. Self-healing is a fundamental process in any network.

Research Self-Healing algorithm :

The research on Self-Healing Algorithms has been carried out. On the 5G network, our team found out that the most failure in the connection of the 5G network is because the high frequency signal cannot travel through obstacles and it can be absorbed by plants and buildings.. Because of this, our team will be working on an algorithm on how to connect to devices which lose connection to the 5G network.

Implement a Self-Healing algorithm on GENI:

The implementation of the self healing algorithm on GENI platform included the use of several virtual machines set up to emulate a cell network. We were successful in writing a script that would detect an outage and reroute the client node so that a successful connection could once again be established. GENI allowed for a controlled environment that could be easily manipulated and would be very reliable.

Implement a Self-Healing algorithm on ORBIT:

The implementation of a self healing network on the ORBIT platform will be the first real world test of the developed algorithm. Using the ORBIT platform will be a step above the GENI tool as it allows for actual radio communication rather than the use of virtual machines. With ORBIT the opportunity for tests including variables such as radio noise is now possible.

2.4 Project Timeline/Schedule

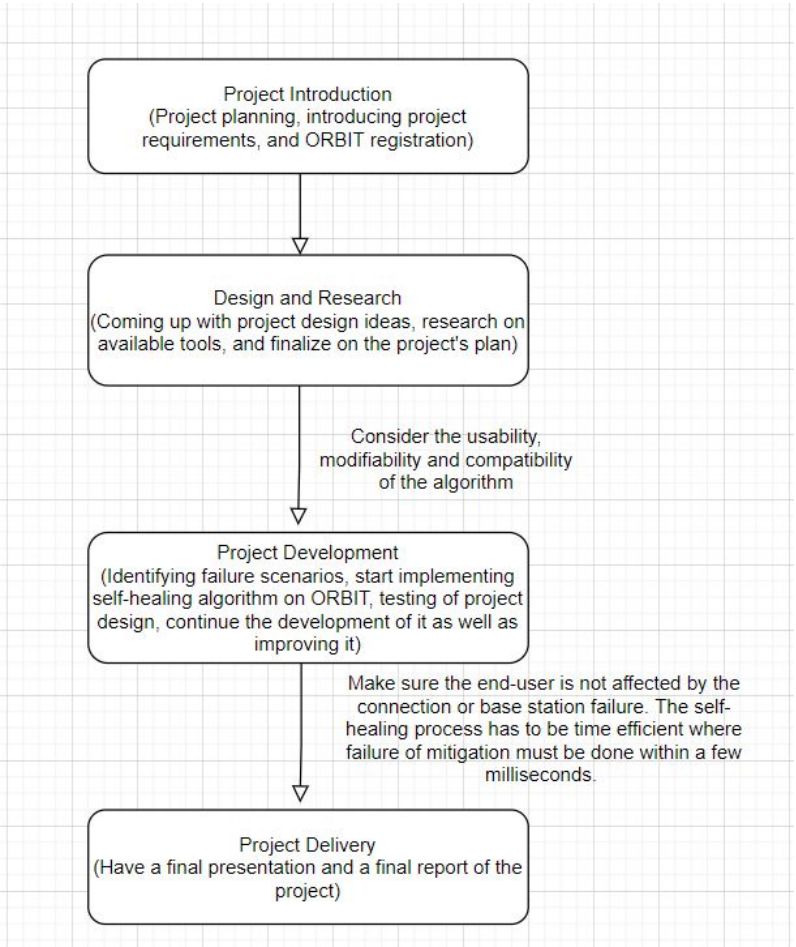

Figure 1.

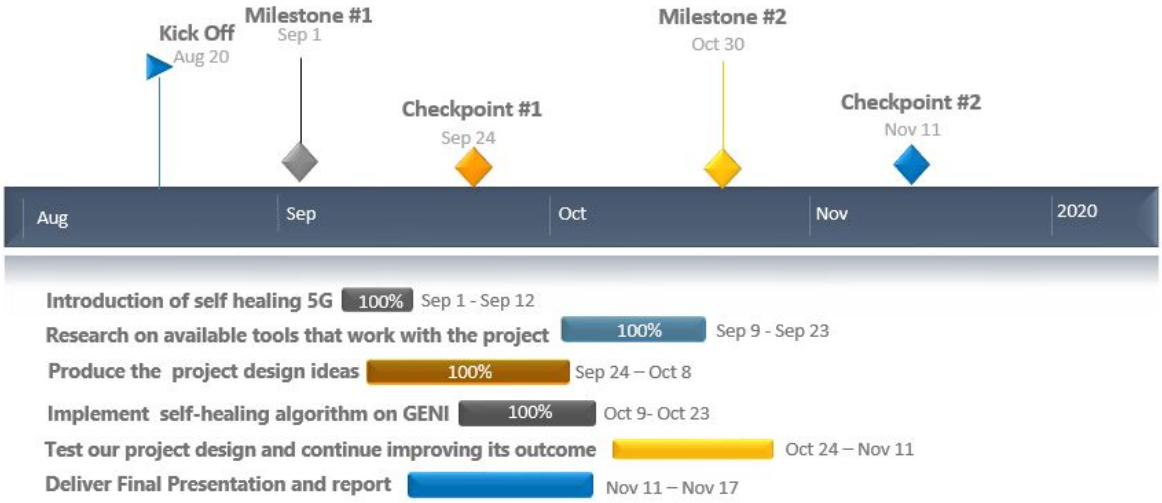

Figure 2. Gantt Chart on the Self-Healing project

The flowchart portrays the project's timeline for two semesters. First and foremost, we are going to start off with a project introduction where we learn what requirements are needed in terms of skills, knowledge and experience that we can contribute to this project. We also need to register for accounts in the free platforms which are GENI and the ORBIT testbed. In this stage, project planning is an essential task as it ensures that we are on track in this project.

After this, we will proceed with designing and doing a lot of research on the project. We will have to come up with project design ideas, do research on the available resources to make this project successful and then finalize on the project's plan. The project design ideas are crucial because it has to include the project's functional requirements which are the effects of a network outage on network users should be mitigated and the self-healing of the network should be time efficient where the failure mitigation must be done within a few milliseconds. Other than that, usability, modifiability and compatibility of the algorithm has to be ensured. The algorithm should be usable for most 5G network applications, modifiable to suit specific network needs and compatible with common network node hardwares.

Next is the program development where we start implementing the self-healing algorithm on the ORBIT testbed, identifying failure scenarios through experiments. testing the project's design and continuing the development of it as well as improving it. This stage is fundamental in the process of the project because this is where the most progress can be viewed and certain adjustments or modifications can be made in our project. The testing for this project would include both virtual and real world tests. This stage is also where we can fix mistakes and enhance the algorithm's efficiency and functionality. We have to make sure that the end-user is not affected by the

connection or base station failure. The self-healing process has to be time efficient where the mitigation failure must be done within a few milliseconds.

Last but not least is the project delivery. This would be done at the end of the second semester where we would have a final presentation and a final report of the project.

2.5 Project Tracking Procedures

We're using the GENI portal created by the professor to keep track of our Self-healing-5G project. In the GENI portal, we can decide who will be the project leader, leading other members in presenting the project status to the professor each week. All members can create slices for project explanation and also be able to check the update of the project's process.

2.6 Personnel Effort Requirements

Individuals need to have a basic knowledge of coding on Linux commands and Python to implement Self-Healing algorithms on testbed and operate testbed servers. During the first semester, all members need to learn how to use GENI experiment network deployment because it serves as a proof of concept experiment platform for wireless aspects.

2.7 Other Resource Requirements

1.Radio Node is a Primary platform for user experiments controlled by a chassis manager through an ethernet interface to reset power on/off.

2. Instrumentation subsystem is a critical component of an automation system create to measures radio signal levels and creates artificial RF interference

3. WLAN monitor system provides a network layer view of the grid's components using several WLAN "observers" spread across the system

4. Support Servers includes front-end servers for web services and backend for experimentation and data storage

2.8 Financial Requirements

There are no financial requirements for the project because we will be using free platforms such as GENI and ORBIT. Experiments will be conducted with those platforms and coding will be done mostly in Python and Linux.

3 Design

3.1 Previous Work And Literature

1. Self-healing methods and base station for obtaining the basic configuration parameters in self-configuration process of base station

   A kind of base station self-configuring process is obtained by the self-healing method and the base station of basic configuration parameters. The present invention relates to the wireless communication field, particularly self-healing method and the base station that a kind of base station self-configuring process is obtained the basic configuration parameter in LTE (Long Term Evolution, Long Term Evolution) system's self-organizing network.

   The diagram below shows the steps for the invention of the self-healing methods and base station for obtaining the basic configuration parameters in the self-configuration process of base station.

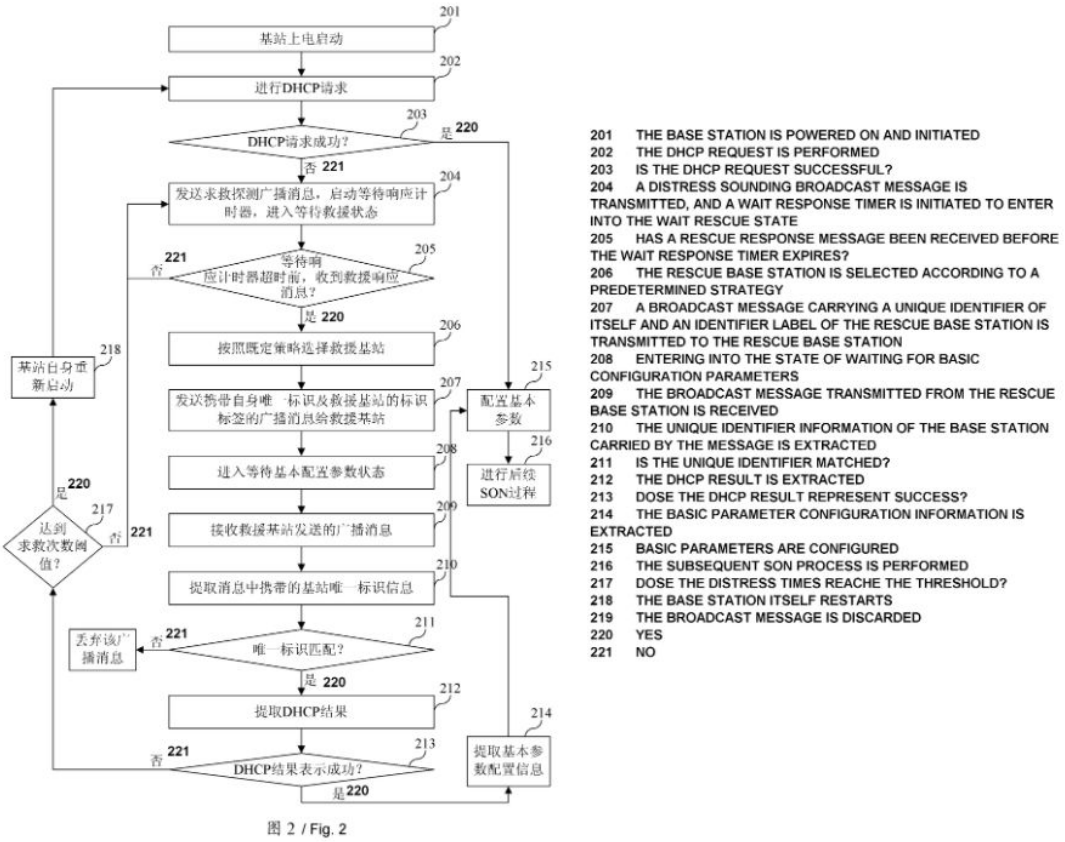

图 2 / Fig. 2

Figure 3.

2. Realizing method and system of self healing of base station cells in long-term evolution system.

   The implementation method of self healing of the base station cell and system in the long evolving system. The present invention relates to base station cell foundation technology in mobile communication Long Term Evolution (LTE, the Long Term Evolution) system, relating in particular to the implementation method and the system of self healing of base station cells in the LTE system.

   This implementation method comprises:

   The base station detects baseband board and occurs judging the vacant base-band resource that self whether has the minimum bandwidth requirement that satisfies sub-district on the unusual baseband board when unusual, if exist, then initiates the sub-district and sets up process and rebuild sub-district on the unusual baseband board; Otherwise the redundancy bandwidth of obtaining the sub-district from normal baseband board is as available base-band resource, and when available base-band resource satisfies the minimum bandwidth requirement of sub-district on the unusual baseband board, rebuilds the sub-district on the unusual baseband board.

3.2 Design Thinking

The majority of the broad design decisions for this project have mostly been laid out before the beginning of this project. The general idea as to how to complete this project falls on the basics of self healing networks which includes outage detection, diagnosis, and compensation.

The design choices left are all about how we choose to implement the three fundamental steps of self healing. For outage detection a series of expected pings between client and server can be arranged so that if a ping fails there can be further investigation. This leads into diagnosis, once a failure has been reported the algorithm needs to begin testing for the source of the outage (client to server, server to internet etc). From there a survey of available nodes needs to be made to accurately distribute the clients in an organized manner.

3.3 Proposed Design

So far, we have experimented with GENI where we created slices and made nodes. We connected the server and client. We also learned how to ssh and look and how it works as a whole. Other than that, we also experimented with the ORBIT testbed. We reserved nodes and we are still in the process of doing the tutorials as we are facing some difficulties with the ssh to the reserved domains in ORBIT. Right now, we are trying to implement the self-healing algorithm where there is a scenario of a base station failure and the client has to connect to its next closest server. The design of this project which we are currently working on in GENI satisfies the functional requirement of this project since our goal is to mitigate the effect of base station failure by seeking helping from neighbouring base stations, where in this case is the server node in our slices in GENI. In addition, we also satisfy the non-functional requirements of this project in the economical and environmental aspects. We are using both ORBIT and GENI platforms to do our project, and we are working in an Indoor Radio Grid Testbed with ORBIT. There are two standards relevant to our project, which are the software development standards and the networking knowledge. We have been using a lot of Linux commands to help us experiment and test our server/client nodes

connection. The ORBIT tutorials we are currently working on teaches us about the WiFi connection to the domains or reserved nodes.

3.4 Technology Considerations

The strengths of the free platforms is that the connection is very fast and it works efficiently. In terms of reserving nodes in ORBIT testbed and making connections with nodes in GENI, the steps are rather simple and easily understandable. However, we are currently facing issues with the ssh into the reserved nodes in ORBIT testbed despite the step by step tutorial provided by orbit-lab.org. Examples are us expecting to connect to the internal1 domain, where our public key is already uploaded in our profile, but we are connected to external2. Other than that, when we tried to connect to the reserved domain, we always got "permission denied public key". This is the main issue we are currently facing and trying to solve. The weakness of GENI is that whenever we create slices, it will expire in 4 days or so, and then we would have to create another slice in order to ssh certain nodes.

The recommended solution as of now would be to figure out the ssh issue with the login at the reserved domain in the ORBIT testbed. The furthest step we could currently do with ORBIT is configuring the SSH keys. If we can get that figured out, we will be good to go to conduct further testing and experiments with the nodes.

3.5 Design Analysis

Our proposed design is a very strong proof of concept for our overall plans of a general self-healing algorithm. It is very feasible with our current knowledge of networking and the platforms GENI and Orbit. We already have demonstrated our knowledge of routing protocols for our proof of concept, our next step is to implement a shell that executes our protocols, and derives the conditions that cause the need for our protocols.

Once our proof of concept is complete, our next step is to add more nodes and servers to our design. The other addition will be different degrees of network failure and how to handle them. The goal of this is to continue making our algorithm function in a more general manner. After this iterative process is complete, we will have a generalized self-healing network algorithm.

3.6 Development Process

We are following the development of the waterfall, which starts with the requirements (verify that all network stations can be self-healing when one is down, then another station has to support it), then gathers the software we need to implement such as Linux and python (mostly on Linux). We also have to design how fast the station will react when noticing a problem and test if the failure station will automatically transfer clients to other working stations, which prevents clients internet's connection lost.

3.7 Design Plan

This project will need to work with an Indoor Radio Grid Testbed and across large-scale open-access wireless networks. We need to create slides in Geni (wired connection), design, test, and improve the station's automated repair. There will be at least two stations and two clients for this project. In the case of the user's internet connection, there are pings connecting the internet node from client to internet. If the ping fails or there is an outage meaning that the base station is

not functioning, then the network will attempt to heal itself by rerouting the traffic from the failing base station, to the working base station and finally to the broader network. Once this is done, traffic between the Internet and Client will continue. We also need to work with an Orbit Testbed that can be accessed across the wireless network world. Thus, we have to ensure that our stations have proficiency in detecting and compensating for the system's outage. Automating on a smaller scale will transfer well into a larger scale network. Although time saved on a small scale (one node) network will not save many resources, on a larger scale the benefits will be much greater.

4 Testing

4.1 Unit Testing

Using the Linux command to test Geni and ORBIT Framework, we will test individual node failure and build a self-repaired network. Besides, we are also writing scripts using bash for the self-healing algorithm and then testing it using Linux commands. In these tests, a connection between a server and internet will be broken and the network will attempt to reroute so that the client receives a successful connection to the internet. After successful completion of a single node repair, we will move to a larger scale network for testing.

Aside from that, we also did a wifi experiment on the effect on nearby networks on the 802.11 networks. Using the outdoor orbit testbed, we configured the nodes in groups of two, based on its own wireless network where each group consists of an access point, a server and a client. First, we configured the wireless access points, on both channels, 1 and 2. After designating the nodes as receivers and servers, we can verify if we are able to ping from one station to the other using the IP addresses. The throughput of both the wireless access points from channel 1 and 2, can be measured using the iperf network tool where neighbouring networks are on the same channel.

4.2 Interface Testing

The first interface that must be tested is the proper communication between server and user in our algorithm. These two nodes must be in constant communication in order to correctly detect a failure as well as routing a path to recovery. If the communication between these two nodes is incomplete, a failure may not be detected or routed.

Similarly, the other interface that must be tested is the communication between server nodes. The server nodes must constantly keep tabs among themselves in order to detect network outages and plan a path of recovery. This communication includes current connections, nearby nodes, and overall health of a server node and surrounding nodes. If this communication is faulty, it will be impossible to correctly address failures.

4.3 Acceptance Testing

To prove the capability and success of this project, it is necessary to show progress toward the main pillars of self healing networks which are outage detection, outage diagnosis, and outage compensation. Proving the competency of the project, our system must be able to handle an unexpected loss in signal by first detecting the severity and location of the loss, diagnosing the cause of the loss, and drawing on the resources of surrounding cells in a quick and efficient

manner.

4.4 Results

- The successes we already achieved is that we are able to test the nodes in the slices we created in GENI. We are currently implementing the self-healing algorithm with one scenario where one of the base stations, which in this case is the server node failing to connect to the client, or vice versa. The client node is expected to connect to the next nearest server. This is currently in the works. We are currently facing failures with the ORBIT Testbed as we could not ssh into the reserved domains. We are looking into the basic tutorials with ORBIT where we are learning how to turn the nodes on, and running scripts.
- We learned that there are many options to ssh into the reserved domains and we are currently exploring all those options. If we could not figure out the solutions to these problems, we would have to contact the authorized staff at the orbit lab to help us with this issue or we might have to consult our advisor for a different plan or solution on how to go about this. As we make progress with our project, we would do it step by step, and if it requires any change, we would find other solutions to go about it.

5 Implementation

- Get in contact with an expert Orbit developer that can point us in the direction of appropriate documentation, which will lead us to successfully working with ORBIT

- Be able to configure the behavior of the nodes on self-healing.

- Learn more about the specifications of the specific orbit nodes in order to be able to write the disk images is also necessary

First we must start with basic tests (rerouting etc.) and move to more complex testing. This testing will include real world problems such as radio noise, unexpected outages to simulate a more real testing environment.

Second we start some tests about the effect of nearby networks with different channels. This is used to test how the connection is on different channels in the same radio frequency.

6 Closing Material

6.1 Conclusion

We built a simple self-healing on GENI representing a base station's case losing connection to the broader network caused by a bad storm or an infrastructure failure.

- The internet node represents the broader network.
- the server nodes represent base stations.
- the client nodes represent users.
- The pings represent the Internet node that communicates with clients 1 and 2.

If an outage is detected (connection lost between the Client and internet), the network will attempt to heal itself by rerouting the traffic from the failing base station to the working base station and then to the broader network. Once it goes through all processes (with 100ms of the internet lost), traffic between the internet and Client will be back to normal in which users can connect to the internet again.

6.2 References

List technical references and related work / market survey references. Do professional citation style (ex. IEEE).

Citations:

郑, 纪伟. (2011, December 28). Self-healing method for acquiring basic configuration parameters in the self-configuration process of base station and base station. Retrieved October 25, 2020, from https://patents.google.com/patent/CN102300207A/en

张, 现周, 易, 鸿锋, & 张, 文英. (2012, July 25). Realizing method and system of self healing of base station cell in long-term evolution system. Retrieved October 25, 2020, from https://patents.google.com/patent/CN101772059B/en