

Hardware and Software Prototyping of 5G-and-Beyond Wireless Systems and Rural Broadband

Documentation and Educational Website Final Design Document

Team Number: 31

Client: Dr. Hongwei Zhang

Advisers: Dr. Hongwei Zhang

Team Members/Roles:

Adam Kruger

Danny Cao

Ethan Gabriel

Zachary Miller

Team Email: sdmay23-31@iastate.edu

Team Website: <https://sdmay23-31.sd.ece.iastate.edu/>

Revised: 04/28/2023

Table of Contents

	2
Revised Project Design	3
Introduction	3
Product	3
Development and Progression of Senior Design Project	3
Functional and Nonfunctional Requirements	5
Create and Host a Functional Website:	5
User-Friendly Design and Experience through the Website:	5
Educate and Provide Learning Material	5
Compile 5G and Networking Experiments	5
Relevant Standards	6
3GPP TS 38.300:	6
3GPP TS 38.331:	6
3GPP TS 38.201:	6
Engineering Constraints	6
Performance	6
Maintenance	7
Security Concerns and Countermeasures	7
Implementation Details	8
Testing Process and Results	9
User testing - Feedback from users	9
Experiment Testing - Evaluating Complexity and Validity	10
Website Testing - Sphinx Documentation and Read the Docs	11
Related Products and Literature	13
Appendices	14
Appendix I - Operational Manual	14
Appendix II - Versions	16
Appendix III - Other Considerations	18
Appendix IV - Code	18

Revised Project Design

Introduction

This project serves as a means of addressing the issue of lack of documentation on the subject of 5G as a whole as well as a tool of supplementing supporting information for the 5G ARA network architecture. Our advisor, who is also the receiving client of the team's project, wants to have a means of providing learning material or supporting documentation on both general 5G networking technical details as well as the operation of the ARA wireless living lab, which is being led by the advisor/client. The team then sought to address this issue through our project and product by targeting mainly undergraduate students as the focus of the content of the project website.

Product

To address this issue, we, the team, created an educational and documentation website that would address the topic of the 5G network and provide more details on the inner workings of the ARA wireless living lab.

Throughout the project, the target audience or users can go through structured learning modules to go over the overall architecture of 5G networks. If seeking more detailed information, users can access the learning modules that go over the technical details and learn about the application of the concepts through the learning modules on the ARA wireless living lab. Users will also be able to replicate experiments created to teach a certain network function.

As for the client, he will be able to link the website to his main ARA documentation and website and easily access the website and make any changes or modifications as needed. Adding more content will be hassle-free for the client as the team will utilize an open-source application currently used by the client to host the main ARA documentation website.

Development and Progression of Senior Design Project

At the end of the 491 semester, our team and client were still unsure what content or documentation should be prioritized for the target audience. And thus, during the 492

semesters, the team decided to focus on structuring meetings with the advisor/client to focus more on target deliverables rather than starting to create the project design framework. This allowed the team to compose learning modules and develop a core part of the project. The core part is the experiments that users can do through testbeds offered by the college to showcase concepts learned or provide tools for users to test 5G applications. It should be noted that while the experiments were a crucial part of the deliverables, the content regarding 5G was still prioritized higher as per the advisor/client's request. This then allowed the team to begin the creation of the documentation and educational website.

Thus, the team began to create the website's framework before the content population through coding with HTML. The team then presented this rough outline and framework of the initial website design to the advisor/client before proceeding with the content population. The advisor/client had requested the use of the open-source application "ReadtheDocs" to better sync versions and updates and update the project website in later futures for personal use while under a familiar code scheme once the team was no longer employed. This resulted in the team scrapping the HTML framework of the initial website design and restarting utilizing the open-source application. However, this benefitted the team as the application allowed for a more streamlined process of updating the project website.

After completing the project website design and layout, it was time for the content population. The team had spent considerable time consolidating all the information and fact-checking the content, following an outline of learning modules that will be pushed to the website. The learning module outline was partially approved by the advisor/client before being on the website. We say partially, as updates to the ARAnetwork were and are still in progress. Because of that, we are challenged to stay updated with technologies as they are still being deployed. The team generated the content per the learning module standards and spent a considerable amount of time designing experiments that could be associated with the learning modules listed on the website. This required time to learn how to use various testbeds such as Powder, GENI, ARA, srsRAN, etc. However, some of these testbeds were dropped in the final product as the team had felt that they could not be utilized in a manner that would be user-friendly to follow or relevant to the current status of ARA, and had felt this affected the goal of providing a means of simple content delivery to users.

After all learning modules were completed and various experiments were designed, the team received feedback from the advisor/client. While the experiments had passed, the learning modules still lacked more technical details regarding the specific operation of the network protocols. This was addressed and added to the project and website. Some learning modules were restructured to better fit the technical detail description. While the end project design was still the proposed documentation and educational website, the team experienced many design changes through website framework design, testbed application changes/removal of experiments, and content structure.

Functional and Nonfunctional Requirements

With the documentation and educational website project, we are expected to meet the standards of requirements set by the client, advisor, and team by means of analyzing effective content delivery and experimental applications.

Create and Host a Functional Website:

- Coding the website content and framework through HTML.
- Utilizing an open-source documentation tool, “Read the Docs,” we can host and automate the building and versioning of the website.
- Using “GitHub” when pushing our code of generated content

User-Friendly Design and Experience through the Website:

- Utilizing layout and balance schemes to give a clear information hierarchy that guides the user’s eye through the page.
- Ensuring a consistent color theme.
- Use of images and graphics to support content and to communicate information to users
- Clear navigational design; users need to be able to find the information they are looking for quickly and easily.

Educate and Provide Learning Material

- To a target audience of undergraduate students, educate users on the 5G network and the fundamentals of core networking material.
- Conform and organize information by working with advisors and referring to provided textbook resources.
- Create core topics and label them as learning modules to condense the information on a subsection of networking.

Compile 5G and Networking Experiments

- Using varieties of testbeds to host experiments: GENI, POWDER, ARAn.
- Create experiments based on learning modules, 5G, and network functionality.
- Document the experiments through instructional steps.
- Experiments are structured in a layout that users can replicate.

Relevant Standards

3GPP TS 38.300:

Technical specification developed by the “Third Generation Partnership Project.” It designs the overall requirements and principles for the 5G NR system regarding topics such as the physical layer, MAC layer, RRC layer, etc. Information and content were generated based on this standard and are noted on the team’s project.

This specification also includes detail regarding the frequency bands that can be used for 5G NR and modulation schemes that can be used for data transmission, all of which are once again logged onto the website.

3GPP TS 38.331:

This technical specification defines the procedures and messages used by the RRC layer. More specifically, it includes details on how the UE and gNB establish and release connections. States for the RRC layers were a core learning module in the project website and relates to this standard, and as such, this standard was used for reference and a guideline for information on the states and RRC layer.

This standard was utilized in one of our testbed experiments. The experiment would test connections between the user equipment and base stations, with GENI using a “virtual network topology.” Both experiments are documented in the ARA testbed and the GENI testbed.

3GPP TS 38.201:

Specification defines the protocol architecture and procedures for the 5G NR system, including the control plane and user plane protocols used to manage the connection between the user equipment and the base station. Three learning modules were outlined following this specification and, thus, were used to guide team members in structuring the content for the modules on this protocol architecture.

Engineering Constraints

Performance

Performance is an important engineering constraint when it comes to website design and development. A website’s performance is typically measured by its speed and responsiveness, which directly impacts the user experience. While the team had thought of previous hosting

services to host our website, we had landed on an open-source application that addresses all keys of performance for a website. These key aspects include:

- **Page Load Time:** Time it takes for a web page to load in a user's browser.
- **Responsiveness:** The speed with which a website responded to user interactions.
- **Server-Side Performance:** Performance of the web server that is serving/hosting the website

Maintenance

This refers to the ongoing activities to ensure that the website continues to function properly and meet the needs of its user. This constraint is also an important factor to consider for the future of this documentation and educational website as future members can continue to update the content or make changes.

This is done through content management as part of the maintenance constraint. Another aspect to consider within the maintenance constraint would be user support and feedback as the website should be able to meet user experience and this can be addressed by indicating a channel to contact for user feedback such as an email.

Security Concerns and Countermeasures

Physical Security - No physical security concerns or countermeasures were necessary for the duration of our work on this project.

Cybersecurity - Since our website is hosted on ReadTheDocs, network security is mostly handled by their service. However, ReadTheDocs can not host documentation for private repositories, so the main directory and files of our project are public. This means that anyone can access our code and documentation files. Since they cannot edit these files unless given permission, our only concern of ours would be plagiarism.

Implementation Details

In this section, we outline the implementation details of our final product: a comprehensive documentation website focused on 5G-and-Beyond Wireless Systems and Rural Broadband. The implementation process involved using Sphinx and Read the Docs to create the website, leveraging various resources and collaborating with experts to develop learning modules, and conducting user testing to ensure the quality and relevance of the content. Additionally, we maintained regular communication with the client to gather requirements and feedback throughout the project.

1. Website implementation using Sphinx and Read the Docs:

We chose Sphinx, a powerful documentation generation tool, as the foundation for our website due to its flexibility, ease of use, and support for a wide range of output formats. We created our documentation using reStructuredText (reST) and organized the content into sections using directories and subdirectories. To configure the website's appearance and settings, we utilized the configuration file (`conf.py`) and the index file (`index.rst`).

For hosting our documentation website, we relied on Read the Docs, a popular platform that offers seamless integration with Sphinx-generated documentation. We connected our repository to Read the Docs and created a `.readthedocs.yaml` configuration file to define the build settings. The platform automatically built and deployed our website whenever updates were pushed to the connected repository.

2. Developing learning modules using various resources:

To create comprehensive and relevant learning modules for our website, we leveraged a wide range of resources, including texts related to 5G, online articles, and the expertise of members from the ARA team. We compiled and synthesized information from these sources to develop content that covered essential 5G concepts, technologies, and use cases. This approach allowed us to create learning modules that provided an in-depth and accurate understanding of the subject matter.

3. User testing for content quality assurance:

To ensure the quality and relevance of our learning modules, we conducted user testing with individuals of varying levels of familiarity with 5G concepts. We gathered feedback from users on the clarity, comprehensibility, and overall learning experience provided by our website. Based on the insights gained from user testing, we iteratively refined the content, structure, and design of our learning modules, ultimately ensuring that they met the needs and expectations of our target audience.

4. Client correspondence and requirement gathering:

Throughout the implementation process, we maintained regular communication with our client to gather requirements, share updates, and seek feedback. By keeping the client informed and involved, we were able to make data-driven decisions and adjust our implementation strategy as needed. This approach ensured that our final product aligned with the client's objectives and

delivered a valuable educational resource on 5G-and-Beyond Wireless Systems and Rural Broadband.

In summary, the implementation of our documentation website involved using Sphinx and Read the Docs for website creation and hosting, leveraging various resources to develop comprehensive learning modules, conducting user testing for quality assurance, and maintaining close communication with our client to ensure the successful delivery of a valuable educational resource on 5G-and-Beyond Wireless Systems and Rural Broadband.

Testing Process and Results

User testing - Feedback from users

Our testing process extended beyond the technical aspects of the website to include user testing and gathering feedback from individuals attempting to learn from our educational resources. By incorporating quizzes and a wealth of information on various 5G concepts, we aimed to provide an engaging and comprehensive learning experience for our users.

The user testing process involved the following steps:

1. **Identify target users:** We selected a diverse group of users with varying levels of familiarity with 5G concepts to ensure that our website effectively catered to a wide audience. This included individuals with minimal knowledge of the subject matter, as well as those with more advanced understanding.
2. **Provide access to the website:** We granted the selected users access to the documentation website, allowing them to explore the content and interact with the quizzes and other learning materials.
3. **Collect feedback:** We encouraged users to provide feedback on their experience, focusing on aspects such as the website's usability, content comprehensibility, and overall learning experience. This feedback was collected through various channels, such as online surveys, email correspondence, and one-on-one interviews.
4. **Analyze the feedback:** We carefully reviewed and analyzed the feedback received from users, identifying common themes and areas for improvement. This analysis helped us understand the strengths and weaknesses of our website, enabling us to make data-driven decisions for enhancing its effectiveness.

5. **Implement improvements:** Based on the insights gathered from user feedback, we iteratively refined our website by making necessary changes to the content, structure, and design. This included modifying the explanations of complex concepts, adjusting the difficulty of quizzes, and improving the overall user experience.
6. **Repeat the process:** We continued to gather feedback and make improvements until we were satisfied that our website met the needs and expectations of our target users.

By incorporating user testing and feedback into our testing process, we ensured that our educational website not only functioned correctly but also provided a valuable and enjoyable learning experience for users exploring various 5G concepts.

Experiment Testing - Evaluating Complexity and Validity

In addition to user testing and feedback, we conducted experiment testing to evaluate the complexity and validity of the hands-on experiments provided on our website. These experiments were designed to help beginners gain a deeper understanding of the interworking of 5G and its various components. To ensure that these experiments were both accessible to our target audience and effective in conveying key concepts, we implemented a rigorous testing process.

The experiment testing process involved the following steps:

1. **Selection of experiments:** We carefully curated a range of experiments that covered essential 5G concepts, such as network architecture, radio access technologies, and core network functionalities. The selected experiments were designed to provide users with a hands-on learning experience that facilitated a deeper understanding of 5G systems.
2. **Internal testing:** Before presenting the experiments to users, we conducted internal testing to evaluate their complexity and validity. This involved running through each experiment step-by-step, examining the clarity of instructions, the ease of execution, and the relevance of the experiment's outcomes to the underlying 5G concepts.
3. **Adjustments and refinements:** Based on the internal testing results, we made necessary adjustments to the experiments, simplifying the instructions or modifying the experiment's design to better align with the targeted learning objectives. This iterative process ensured that the experiments were both accessible to beginners and effective in conveying the intended concepts.

4. External user testing: With the refined experiments, we conducted external user testing by inviting a group of users with varying levels of expertise in 5G to participate in the experiments. We provided them with access to the experiment materials and monitored their progress to assess their understanding, engagement, and overall learning experience.
5. Feedback collection: Following the completion of the experiments, we collected feedback from the users, focusing on aspects such as the clarity of instructions, the difficulty level, and the overall effectiveness of the experiments in conveying the intended concepts. This feedback was gathered through online surveys, email correspondence, and one-on-one interviews.
6. Final refinements: Based on the insights gleaned from user feedback, we made further refinements to the experiments, adjusting their complexity and ensuring their validity in teaching 5G concepts. This process was repeated until we were satisfied that the experiments provided an optimal learning experience for our target audience.

By implementing a thorough experiment testing process, we were able to ensure that the hands-on experiments offered on our website were accessible to beginners, effectively conveyed key 5G concepts, and provided a valuable learning experience for users aiming to better understand the interworking of 5G systems.

Website Testing - Sphinx Documentation and Read the Docs

Testing a Sphinx documentation website is a crucial part of the development process to ensure that the website is functioning properly, is fast, and delivers an optimal user experience. The testing process can be divided into two main phases: testing on localhost and testing and rebuilding the website through Read the Docs.

1. Testing on localhost:

During the development of the Sphinx documentation, it is essential to test the website on the local development environment, also known as localhost. This allows for immediate feedback on the structure, appearance, and functionality of the documentation without needing to deploy it to a public server.

To test the Sphinx documentation on localhost, use the following steps:

- a. Build the documentation: Run the sphinx-build command, specifying the input directory (where your source files are located) and the output directory (where the generated output should be saved). This command will parse the reStructuredText files and generate the HTML output in the specified output directory.

```
"sphinx-build -b html <source_directory> <output_directory>"
```

- b. Preview the documentation: Open the generated index.html file in the output directory using your preferred web browser. This allows you to view the website as it would appear to end-users, providing an opportunity to verify the content, structure, and appearance of the documentation.
- c. Check for errors and broken links: Navigate through the website, checking for any errors in content or formatting, as well as broken links or images. This can be done manually or by using tools like sphinx-linkcheck to automatically identify broken links.
- d. Iterate the process: Make any necessary changes to the source files and rebuild the documentation using the sphinx-build command. Continue iterating until you are satisfied with the results.

2. Testing and rebuilding the website through Read the Docs:

Once the documentation is ready for deployment, it is essential to test and rebuild the website using Read the Docs, an online platform that provides hosting services for Sphinx-generated documentation.

- a. Connect your repository: First, sign in to Read the Docs and connect your GitHub, GitLab, or Bitbucket repository containing the Sphinx documentation.
- b. Configure the build process: Create a .readthedocs.yaml configuration file in your repository to define the settings for Read the Docs, such as the Python version, Sphinx version, and build requirements.
- c. Trigger a build: Read the Docs will automatically trigger a build of the documentation whenever changes are pushed to the connected repository. You can also manually trigger a build by navigating to the "Builds" tab on your Read the Docs project page.
- d. Review the build output: Once the build is complete, review the build output on the Read the Docs project page to ensure that there are no errors or warnings.
- e. Test the live website: Visit the public URL of your documentation on Read the Docs, and navigate through the website to verify that it works correctly and performs well.

By following this testing process, you can ensure that your Sphinx documentation website is functional, fast, and delivers a seamless user experience, both during local development and when hosted on Read the Docs.

Related Products and Literature

In regard to related products, it's a little challenging to consider what would classify as a related product in our project scope. Our project was designed to mesh with a client's request of providing information on both 5G networks and relating it to the ARA wireless living lab. There isn't really a related product that meets that specific description, but in terms of overall product in reference to a learning website, funny enough, that would just be classified as the internet at that point.

However, related literature may be a different story. As our modules taught 5G networks concepts and base networking fundamentals, and as such, we reinforced our learning about these topics through the use of some textbooks when discussing some of the concepts taught in the learning modules. These textbooks that were used as a reference to design a learning module outline included:

- 5G Mobile Networks: A Systems Approach (Open-source book)
- 5G New Radio in Bullets
- IEEE: Wireless Communication
- IEEE: Communications

Appendices

Appendix I - Operational Manual

While multiple systems are included within our project as a whole, we will go over the steps to operate the main system we've put in place, which is our documentation website.

To begin, the following steps are necessary to create an environment where one can work on, deploy, and edit the code within the project.

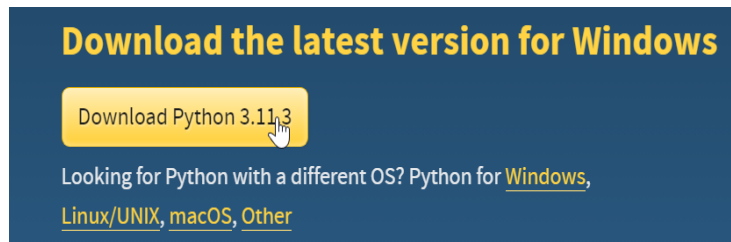
1. Download and install Python3.

We recommend adding python to the environment variables.

The folder where python was installed is determined by whether the user upon setup selected “install for all users”, or not. The folder

corresponding to python310

will be what is added to the environment variables on the host machine.



2. After Python3 has been installed, a documentation generator, which is written in python, named Sphinx, must be installed. The command “pip install sphinx” will allow the necessary dependencies and libraries to be installed to carry out compilation of the documentation website.
3. The next step is to be able to clone the repository that the website is hosted on. This involves using and installing the git CLI or installing git on the machine. Once git has been installed, the repository can be cloned into a folder that has been created by using the command: “git clone <<GithubRepositoryHTTPURL>>”. Once this has been completed, the full directory, or repository that will contain the documentation website will be available locally.
4. In order to compile and test the website on a local machine, it is necessary to first have the directory which contains all of the relevant website files. The directory will contain many folders and subfolders, containing .rst and .html files, as well as styling and other configuration files. Within the main directory, there exists a folder named ‘docs’ which contains much, if not all of the content that is required for the website. In order to build the website, and render html files from the .rst files, a command ‘make html’ must be run in this directory. There also exists another path, docs>>build>>html. This is the directory where the built html files from the previous command can be hosted from by running the following command: ‘python -m http.server’. Once these steps have been completed,

navigate to “<http://localhost:8000/>”, and see the website that has been generated from the .rst files.

```
PS C:\se492\docs\build\html> python -m http.server
Serving HTTP on :: port 8000 (http://[:]:8000/) ...
```

5. The previous step is what must be done to ensure proper testing on our own machines before officially sending any changes or files to GitHub, or the official website hosting service itself. However, there are further steps to take for the website to be hosted publicly. Changes made on a local machine after rigorous verification and testing can then be deployed to Github. The commands: “git add .”, “git commit -m “Sample Message”, and “git push” can be used to update the repository with new information and updates. This ensures that there are no outdated/changed additions or deletions that exist on the official website repository.

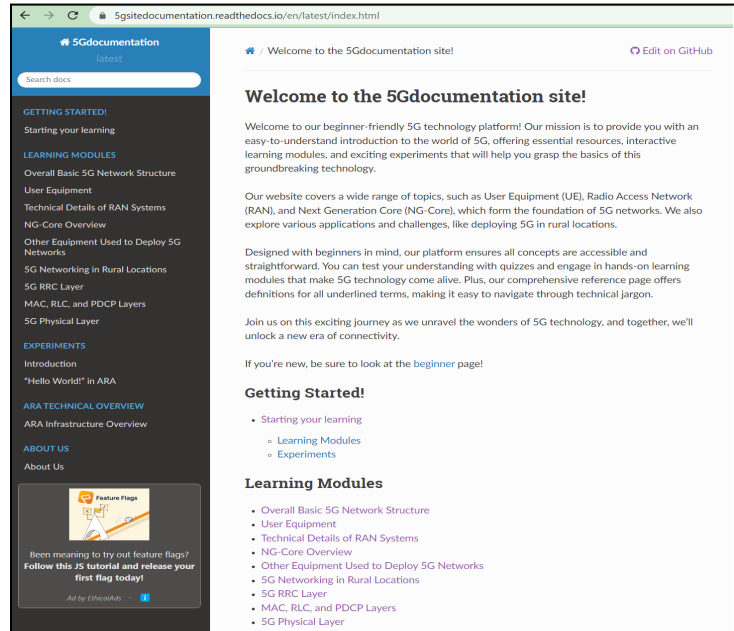
6. Once the repository is updated and contains information that is ready to be released to the public, an account must be created on ReadTheDocs.org. This account on ReadTheDocs.org can then be connected to the *public* Github repository. Once these two services are connected, the website can be built on Readthedocs.org, which scans for the corresponding “conf.py” file in the Github repository to correctly build the project.



The build can take around 30 seconds to a few minutes. When the build is complete, the URL will append the name of the project that exists in “conf.py”, to “.readthedocs.io/en/”. The end result will look something like this:
<https://<<nameOfProject>>.readthedocs.io/en/>



7. After completing all of these steps, the documentation website can be accessed by anyone in the world with an internet connection and access to a web browser!
8. To complete further testing, navigation through the website, and debugging can be done by simply using the website as intended.
- 9.



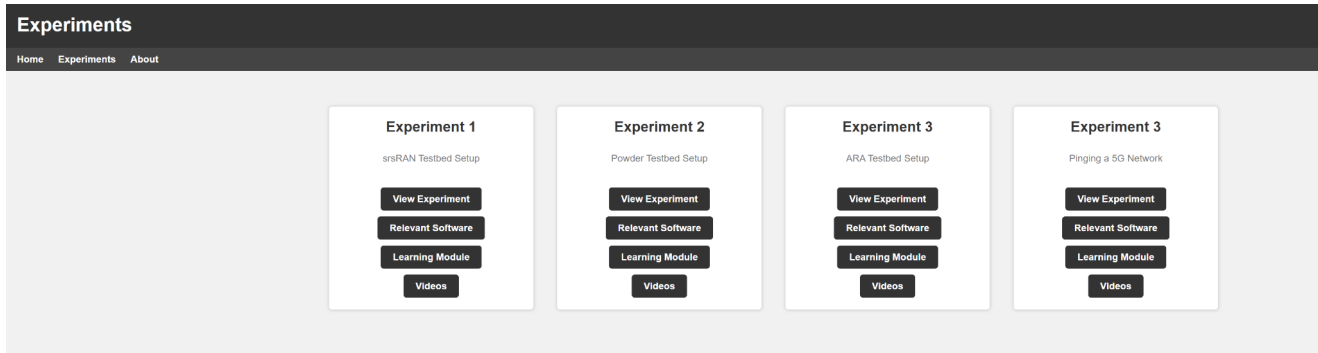
Appendix II - Versions

Versions considered before clients specifications had changed, before learning more about the project, and versions that resulted in failure to achieve specifications:

During our time working on this senior design project, there have been multiple version specification changes and alterations in both content and system structure.

Initially upon structuring, designing and filling in our website, the main structure, navigation, home and other pages appeared as follows:





Computer Network Foundation

Introduction

Pinging a network is the act of essentially sending a relatively small packet of information (data) from one device to another. In our case, we'll be pinging from your computer to a 5G network created and initialized on another device created by the Powder web application. Pinging allows us to test the speed, reliability, and overall status of the network connection between the connected devices. It also allows us to:

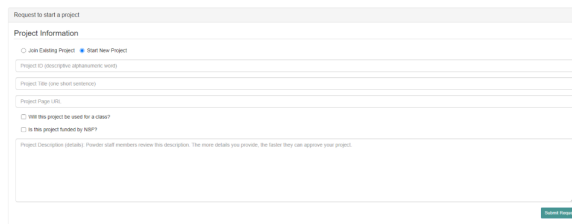
- Troubleshoot Network Problems
- Monitor Network Stability
- Test Network Performance
- Checking Security Breaches
- Verifying Network Configuration

Getting Started

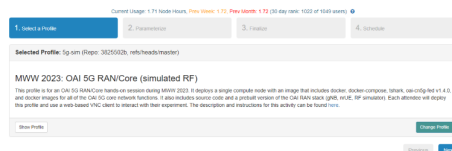
Let's start with our first testbed and experiment as we set up a 5G network simulation on Powder.

Step 1. Launch a web application, go to www.powderwireless.com, and create an account or sign on.

Step 2. Under your username, click on the "Start/Join Project" tab. If this is your first time launching Powder, you'll be required to start a new project and will most likely have to wait for a confirmation of your project once the information has been submitted. Otherwise, join an existing project and continue on.



Step 3. After starting/joining a project, in the top left, click on the tab under "Experiments" and "Start an Experiment." From here, we'll want to simulate a 5G network, so select the profile "5g-sim". You can use the search function to locate the specific profile.



After corresponding with our client, we had come to find that this website design was not up to his professional standards, and he challenged us to make the website look more “professional”. The following week we presented to our client a new design, and he was enthusiastic about the outcome.

Our client also noted throughout the semester that since ARA is approaching deployment, there are some various technologies that we should intend to add once they are finalized and deployed by the ARA organization. While restructuring the website itself and testing/implementing other learning modules, we awaited information regarding new developments to the ARA network. After being informed on these things, though late in the semester, we were instructed to add them to our website with detailed information included.

Due to the hectic timing of ARA's deployment and our intentions to stay included in new developments involving it, there were issues regarding exactly what will be deployed in time for

us to include on our website. Because of this, much rework and correspondence with ARA's team was necessary, even into the very late stages of the semester.

On top of this, we also went through many reviews and tests regarding the learning content on our website by user testing. Upon receiving feedback about the content itself, the layout, and other various criticisms, we updated the website accordingly.

Appendix III - Other Considerations

Other notable considerations -

1. Exact requirements of the product were not explicitly relayed to us by our client, as our client was also unsure of what content we would be able to include on our project. This is because the exact timing of ARA's deployment was in a bit of a gray area, and there was no specific deadline for it.
2. We lost a very intelligent group member of ours, whom we grew to be close with the first semester. This was an adjustment to our team we had to get used to upon starting SD 492.
3. We learned how hectic it can be when working on projects scaled massively to the likes of the ARA network, especially approaching deployment.
4. We learned a ton about how 5G networks work, how they are deployed, and lots of technical inner workings of the process as a whole.
5. We learned what it is like to work with a brilliant team of hardworking, educated contributors and also specialized graduate students.
6. Working under the purview of Hongwei was a special experience that we will surely be able to take with us into our professional careers, if we haven't done so already.

Appendix IV - Code

In the Hardware and Software Prototyping of 5G-and-Beyond Wireless Systems and Rural Broadband project, our main focus was to create comprehensive documentation and an educational website. We briefly explored open-source tools and technologies related to software-defined radio systems, such as srsRAN, ARArAN, and OpenRAN. However, the core of our project was centered around creating documentation using Sphinx.

Sphinx is a powerful documentation generation tool that works seamlessly with Python. It allows us to write documentation using reStructuredText (reST), a lightweight markup language designed specifically for documenting software. Sphinx simplifies the process of generating well-structured, professional-looking documentation, offering features such as cross-referencing, code highlighting, and indexing.

The primary components of a Sphinx project are the configuration file (`conf.py`), the index file (`index.rst`), and the individual source files (written in reST). The configuration file is used to define settings for the project, such as the project's name, author, version, and the desired output format (e.g., HTML, PDF, or ePub). The index file serves as the entry point for the documentation, defining the structure of the document hierarchy and including other source files. The source files contain the actual content of the documentation, written in reST, which is then parsed by Sphinx to generate the desired output format.

The documentation is organized into different sections using directories and subdirectories. This allows for a clear and logical structure, making it easy for readers to navigate the content. Sphinx also supports theming, which enables customization of the documentation's appearance, and extensions, which can add additional functionality to the generated output.

To create our documentation, we began by installing Sphinx and setting up a new project using the provided command-line interface (CLI) tools. We then configured the project's settings and started writing content using reST. As we explored srsRAN, ARARan, and OpenRAN, we documented our findings, experiences, and relevant information in the source files. Throughout the process, we used the CLI to build and preview the generated output, ensuring that our documentation met the desired quality and appearance.

In summary, the core coding procedure for this project involved setting up and using Sphinx to create comprehensive documentation for the Hardware and Software Prototyping of 5G-and-Beyond Wireless Systems and Rural Broadband project. By focusing on the efficient organization and presentation of information, we successfully developed a well-structured, user-friendly educational website that provides valuable insights into the world of 5G-and-beyond wireless systems and rural broadband.