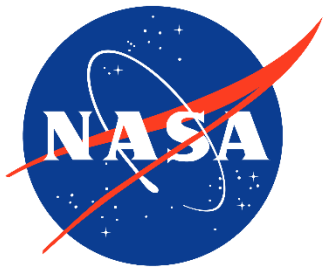
Network Sniffer and Evaluator Task

Julia Marie Odden
NASA Kennedy Space Center
Academic Major: Computer Science
NASA Internship Summer 2020
Jill Giles
28 August 2020



Abstract

I worked as a National Aeronautics and Space Administration (NASA) Interns and Fellowships (NIF) intern in the NE-XS division under the mentorship of Jill Giles and Stephen Aument during the summer of 2020. My assignment addressed an outstanding need to better understand the flow of traffic within the command and control network. The network is a black box in the sense that understanding the generation of data on one end of the bus and unpacking it on the other are simple, but the packaging and dissemination of the data in the middle are difficult to monitor without interruption. Having a better picture of the in-between stages is crucial to ensuring efficient data transfer and to sanity-check the functionality of the system. A participant-style traffic monitor could force the network to route copies of everything passing through the network to another participant, but that participant would add to the load, causing more network traffic and possibly delaying the transfer of mission-critical data to the firing rooms.

To build a network traffic monitor that does not add additional load to the system, I first spent some time studying the basics of network science and constructing a very simple monitor that just registered the presence of data on the network. As the summer progressed, I learned how to dissect packets and data in-app without incorporating external decoding software, meaning that I could efficiently gather statistics about the data my application was monitoring. Finally, I conducted tests on a variety of NASA servers provided by my technical lead.

The resulting application is a network sniffer as opposed to a participant. The tool sits below the level of the data distribution and listens in on the data flow without interrupting it (analogous to sticking a twig into a stream to measure the current with minimal disruption). As packets run by, the tool examines each one, extracts the relevant data for analysis, then returns live statistics on the data flow. The tool runs through a graphical user interface and is compatible with multiple operating systems. The tool requires minimal CPU and is intended to be run in the background of other tasks to help system engineers monitor the network as they use it. Once reviewed, tested, and integrated, the application will help reveal inconsistencies between the working understanding of the network and its underlying protocols, allow engineers to monitor the network in real-time, and lay the groundwork for a better understanding of the network.

Acknowledgements

I would like to thank my mentors, Jill Giles and Stephen Aument, for their support during this process and my technical lead, Anthony Ciavarella, for his endless patience with my

questions as I worked to understand and complete the project. I admire the flexibility and compassion shown by each of my mentors and by the Education team regarding the strange circumstances of this summer's intern program.