Creating a User Interface to Analyse Network Topology

The LHC-OSG Network Monitoring Area gathers and makes accessible various network metrics related to network performance in order to debug complex network problems more effectively. There are over 230 OSG/WLCG perfSONAR toolkits (used for network measurement) each testing to one another many times a day. It can be challenging to identify network problems and, most importantly, their location, which is needed to be able to quickly get the problem fixed. This is a unique dataset with continuous measurements of thousands of research and education network paths that suffers from lack of a good user interface to extract useful information for researchers and network engineers.

My project will be to create a network topology analysis interface that will have four functions to aid with debugging and localizing network performance issues:

- Identify common hops between multiple source and destination pairs, and how these have changed with respect to time. Essentially, this gets the common hops right before a specific time, and right after it. So, if there is a problem noticed at a time, we can see what has changed, and what hops are 'problematic'. This is useful if we notice a common issue between some pairs (such as high packet loss or low throughput).
- The above function can further be used to check how similar the path between two hosts is in 'backwards and forwards' directions. If this is very asymmetric, it could indicate a problem. This functionality could be made separate in the user-interface.
- Given an IP address, return all paths containing that address within a specific time range. If we know a certain address has issues, we can see what paths might be disrupted by this.
- For a source and destination pair discover all the unique paths across a period and use this to find correlations with packet loss or throughput (for example). The output could be a graph with background colours corresponding to the different paths, on which a metric is plotted against time.

In my previous work with the NSF funded SAND project, I have already completed the common hops code, however, as of now, it only runs locally in a CLI window. A large part of my work will be integrating the code with a web interface and designing the web page itself. For this project, **I plan to collaborate with Shawn McKee** (at the University of Michigan) and the other members working on the SAND project. Below is the timeline for the project:

Week 1: Prototype a user-interface using drawing and design tools.

Week 2: Identify suitable web applications to use to provide the user interface.

Weeks 3-6: Integrate common hops code with the web application. Extensive testing to ensure it runs smoothly and correctly. Add a separate tool for checking symmetry.

Week 7: Write code for the second function (given IP, return all paths containing it), test it, and add it to the web interface.

Weeks 8-11: Implement the third and fourth functions (identify unique paths between a source and destination, and output a graph to correlate it with network metrics), test it, and add it to the web interface.

Week 12: Complete testing, package project and finalize documentation.