

# Netflow Presentation Frontend

## Background

JKFlow provides the processing of netflow data into numerous RRDs which allows great breadth in the detail of the netflow data. This project focuses on developing an interactive and easy to use web based front ends to such data.

A current prototype of the netflow visualisation can be viewed at <http://iepmbw.bnl.org/netflow/index.html>.

## Project Description

To develop, test and document a flexible web based frontend for the netflow data. This will involve expanding and developing new code to improve the API and performance of the existing perl/cgi server side scripts and the client end javascripts.

The server side code is primarily a means to dynamically gather and present base components of the frontend; for example to display the statistics of from the BNL site to say SLAC. Much work needs to be conducted to expand the presentation consistency and to iron the bugs in the implementation.

Using the above prototype as a framework, learn and use AJAX technologies and methodologies to improve the user experience of the netflow visualisation and offer suggestions of how to better represent the netflow data.

## Specifics

Specific additions to the netflow visualisation include:

### User authentication

Provide a mechanism for Authentication, Authorisation and Auditing of the visitors and users on the webpage.

### Extend and Refactorise backend

Current backend scripts are written in perl/cgi. They should be understood and suggestions given in terms of code refactorisation and redesigned to provide a more uniform api for the frontend to access the information.

It is suggested that the perl Template library be used to separate the presentation of the dynamic webpages from the code. In particular, emphasis should be put on unit testing of individual components.

### Provide a flexible user front end

Provide a tabular interface on the main page in order to provide ajax based pages for:

- 'Custom Reports': a dynamic page of user definable 'template' webpages whereby the user can quickly and easily create custom reports based on selected input criteria (such as Tiers, groups, sources, destinations) where managerial and technical reports can be quickly and easily accessed. Such template pages should be editable by the user and provide live pointers to the information within. Examples include: weekly reports of aggregate throughput.
- 'Top N Reports': Provide data mining information of all flows that have been processed in a top 10 like fashion.
- Extend the interactive plotting frontend (fixing bugs, optimizing on speed etc)

Furthermore, the web site should be made consistent and professional looking, with facilities to allow easy customization.

### Provide robust installation procedures

Such as using CPAN.

## Other

The code should be kept upto date on the SLAC SVN. The project applicant should apply for a computer account at SLAC [here](#) and contribute towards the wiki page (such as [Terapaths Monitoring Front End Developers Manual](#)).

## Contact

Yee-Ting Li