Automated GOLE Dashboard  
Version 0.4, March 10, – Hans Trompert and Gerben van Malenstein

# Introduction

Based on the requirements document produced by UvA and experience of Hans and John in managing daily operations of the Automated GOLE a first sketch of what a ‘dashboard’ could look like is made in this document. The term dashboard is translated into a central web page that displays system-wide status of control plane and data plane.

The dashboard consists of information that is queried/harvested/pinged from a central agent. Different update intervals will be used for the different types of information, a trade of will be made between how often information is refreshed and the amount of resources that are consumed to generate the information. In the future a distributed way of gathering and sharing measurement information across domains may be chosen.

Both the status of the control- and data plane will be showed on the dashboard, in order to quickly spot errors/faults in the fabric. More detailed faultfinding is not included in the first version of the dashboard (e.g. specific segment testing), this may be added later on.

# Control plane monitoring

The dashboard provides a user the status of the control plane:

* **C1**. By displaying a graph of all NSAs and the features they support, and
* **C2**. By displaying the peering relationship(s) between all NSAs (peersWith);
* **C3**. By displaying the reachability of the NSA host (ping);
* **C4**. By displaying a table containing all NSA ID/peersWith mismatches:
  + **C4**.1. peersWith statements with unknown NSA IDs
  + **C4**.2. NSAs that are not referenced in any peersWith
  + **C4**.3. Unmatched RA->PA/PA-RA peersWith statements.
* **C5**. By displaying if the NSA is functioning properly (could do a simple NSI query but do not want to setup full mesh of control plane peerings between monitoring system and every NSA, maybe this is only possible with a `NSI monitoring protocol’, ideas are welcome here)

In order to discover and construct the NSA control plane graph it is assumed that there is way in which the dashboard host has access to the NSA description documents of all participating NSAs, for example through the use of the DDS, or by the ability to directly collect the documents from every NSA, or any other method.

# Data plane monitoring

The dashboard provides a user the status of the worldwide data plane:

* **D1**. By displaying a graph of all network topologies, and also displays all connections between these topologies. The user may change this view into table view;
* **D2**. By displaying a table containing all isAlias mismatches;
* **D3**. By displaying (either in a table or as a part of the graph) the results of various path segment reservation requests. For this purpose every network topology will provide at least one STP. Suggested reservation requests:
  + **D3.1** The dashboard host will try to reserve path segments between any two neighboring topologies using a set of at least two aggregators.
  + **D3.2** The dashboard host will try to reserve paths between any set of leaf topologies using a set of at least two aggregators.
* **D4**. By displaying the results of actual ping/iperf test through paths that have been setup to fully verify the data plane.

In order to discover and construct the data plane graph it is assumed that there is way in which the dashboard host has access to the topology documents of all participating networks, for example through the use of the DDS, or by the ability to directly collect the documents from every NSA, or any other method.

# Implementation

In order to quickly show the first control and data plane status information and to gain experience in collecting the needed information and displaying the results it is suggested to start with a simple set of scripts that make use of components that are already available in this community (either directly or after some small modification). In a later stadium the use of frameworks like CMon or perfSONAR or others can be considered.

# Dashboard example

Example of what the dashboard may look like: