



GNI API Task Force

GLIF Workshop Winter 2010

CERN, Geneva, Switzerland

Evangelos Chaniotakis, ESnet Network Engineer

Lawrence Berkeley National Lab



GNI API Task Force Overview



- Motivation:
 - There exist several different guaranteed bandwidth services, that do not interoperate due to diverging APIs.
 - We can fix this.
- Targets:
 - Bring network resource management developers together,
 - Develop a common interface for guaranteed bandwidth reservation requests.
 - Develop a software framework (Fenius) to facilitate translation
- Non-Objectives:
 - To become a standard.
 - To get everything 100% correct.
- Participating developers from:
 - G-lambda, IDC, Argia, Harmony, DRAC, KISTI, NCSU

Agenda



- Review progress
- Identify open issues
- Plans for the Task Force
- Plans for the GNI API interface
- Plans for Fenius

Progress



- New GNI API interface developed over the summer
- Developed / improved Fenius for:
 - Argia
 - DRAC (native support for GNI API; no translation)
 - IDC
 - G-Lambda
- Deployed Fenius at several networks / GOLEs
 - StarLight, NetherLight, MANLAN/ION, AIST, JGN2+,
 - NORDUnet, CERNLight, UvA
- Used to control actual data plane (demo later today)

Open Issues



- Service architecture:
 - Is now centralized superagent – not scalable
 - Should evolve to something more flexible
- Topology:
 - Format: current is JSON, should be NDL / NML
 - Pathfinding architecture / topology exchange: DTOX group
- Access control & security
 - AuthN / AuthZ
 - Policy & how it relates to pathfinding



Plans for the Task Force

- We have been around for almost 3 years,
- We should decide in which direction to plan (long term)
- Will we continue working on:
 - Extending the API?
 - Improving software?
 - Producing “lessons learned” documents?
 - Service demos?
- Shall we try and become a de facto standard, or rather focus our work towards standards forums (NSI)?
- Is Fenius going to stay demo-ware or should we harden and operationalize it?
- How fast (if at all do) we want to expand usage of the new API?

Plans for the GNI API interface



- We have now a flexible interface with a few clever features in the schema:
 - Can potentially support multilayer, multipoint topologies, asymmetrical connections, plus some other clever things such as expansion of existing connections (thx to Takahiro Miyamoto)
 - All based on a small, good set of primitives (reserve(), release(), list(), query(), isAvailable())
- How do we want to extend this (if at all)?
 - Fancier resource scheduling (i.e. recurring schedules, independent schedules per “edge”, different bandwidths over time, etc)
 - More technology-specific things (i.e. IPv4 / 6 parameters for endpoints, SDH params, ...)
 - More facilities for the user: allow negotiation of reservation parameters (ie bandwidth, time, VLANs, etc)
- Or should we start *removing* features?

Plans for the Fenius software package



- The Fenius software project now contains:
 - The translation implementations for IDC, Argia, G-Lambda
 - A Web UI / SuperAgent with pathfinding capabilities
 - A client package with some scripting features
- Which direction(s) should we proceed towards?
 - Harden the translators,
 - Implement more functions (i.e. isAvailable())
 - Add more translators (DynamicKL, AutoBAHN, Sherpa, ..)
 - Improve the Web UI,
 - Develop client for iPad ;)



Special Thanks

- Frank Blankman
- Scott Campbell
- Joan Garcia Espin
- John Graham
- Jeroen van der Ham
- Takatoshi Ikeda
- Nils Jacobson
- Gigi Karmous-Edwards
- Tomohiro Kudoh
- Kavitha Kumar
- Ali Lahlou
- Andrew Lake
- Tom Lehman
- Mathieu Lemay
- John MacAuley
- Gerben van Malenstein
- Edoardo Martelli
- Takahiro Miyamoto
- Azher Mughal
- Fumihiko Okazaki
- Jan Radil
- Jordi Ferrer Riera
- Sandor Rosza
- Ryousei Takano
- Thomas Tam
- Jin Tanaka
- Alan Verlo
- John Vollbrecht
- Fred Wan
- Xi Yang