**Network Service Interface Framework Architecture**

Status of This Document

This document provides information to the Grid community on the service interface between a requesting software agent and the provider software agent that offers and delivers a Network Service. It is intended to describe the processes and environment in which software agents interact to deliver the service(s). Representing applications or other networks, these agents may request certain services of other network agents. Distribution is unlimited.

Copyright Notice

Copyright © Open Grid Forum (2008-2010). All Rights Reserved.

Trademark

OGSA is a registered trademark and service mark of the Open Grid Forum.

Abstract

The Network Service Interface (NSI) is defined to be the set of protocols and parameters that are used between a software agent requesting a Network Service and the software agent providing that Network Service. This NSI Framework Architecture describes a framework to support a set of Network Services. These services allow an application or operator to request and managing Network Services from other operators. This document and its partner documents describe each of the NSI Network Services (GWD-R) that form the NSI specification.

Contents

[1. Context and Overview 2](#_Toc265674227)

[2. NSI architecture 3](#_Toc265674228)

[2.1 Network services 3](#_Toc265674229)

[2.2 The Network Service Interface 3](#_Toc265674230)

[2.3 The Network Service Agent 4](#_Toc265674231)

[2.3.1 The Network Resource Manager 5](#_Toc265674232)

[2.4 NSI Sessions 5](#_Toc265674233)

[2.5 NSI service extensibility 6](#_Toc265674234)

[2.6 The NSI Service Plane 6](#_Toc265674235)

[2.7 Hierarchical communications model and federation 7](#_Toc265674236)

[3. The NSI Protocol 9](#_Toc265674237)

[3.1 NSI Protocol overview 9](#_Toc265674238)

[3.2 NSI messages 9](#_Toc265674239)

[3.3 NSI Service Instances and Primitives 9](#_Toc265674240)

[3.4 NSI Service Definitions 10](#_Toc265674241)

[3.5 Temporal aspects of NSI services 10](#_Toc265674242)

[3.6 Trust and authentication in NSI 11](#_Toc265674243)

[3.7 NSI Service Plane error handling 11](#_Toc265674244)

[4. Representing network resources 13](#_Toc265674245)

[4.1 virtualization 13](#_Toc265674246)

[4.2 Describing network topologies 13](#_Toc265674247)

[4.3 Service Termination Points 14](#_Toc265674248)

[5. Contributors 16](#_Toc265674249)

[6. Glossary 16](#_Toc265674250)

[7. Intellectual Property Statement 18](#_Toc265674251)

[8. Disclaimer 18](#_Toc265674252)

[9. Full Copyright Notice 18](#_Toc265674253)

[10. References 18](#_Toc265674254)

# Context and Overview

Over the last decade, global networks have begun delivering high performance transport services directly to applications that require performance levels or capabilities unavailable in conventional best-effort IP networks. The ability to create connections between a fixed set of ports worldwide, with specific, predictable, and often demanding performance characteristics, enables emerging global collaborations to establish well-defined and highly customized network environments to support the end users and their applications. This is particularly true within the Research and Higher Education space and the growing Grid community.

Connections across these transport networks have been historically reserved and provisioned in a variety of ways. The most common approach is manual provisioning – typically performed by a network engineer. More recently, some networking communities have developed tools and protocols to automate the process of network resource allocation and to allow the user or application to participate directly in the path creation process. These new approaches to automating transport connection provisioning are the basis for the standardization effort being described in this recommendation.

Automated connection-oriented transport provisioning capabilities are currently being deployed by Research & Education (R&E) providers as well as by commercial providers, and could eventually be implemented in home/ retail networks as deployment progresses. These automated provisioning systems, while being developed independently by different groups, all have common elements. They have developed software based control and/or management agents to regulate access to these resources, to schedule and reserve resources, to trigger or control timely provisioning of the network resources, and to monitor and release resources. These controllers are deployed in two different contexts. One context is application (or Grid) centric, where a network provides a resource to an application or middleware. The other context is network centric, where network resources are collaboratively shared among networks to expand or improve network performance or reach. In the former context, a user or application agent is requesting the service of a network provider. In the latter context, one network is interacting with other network(s) to manage these resources and deliver a comprehensive and well integrated service portfolio to the user community. This informational document defines an architecture for the NSI protocol which suppoorts of these contexts.

The NSI architecture defines several key elements: the Network Service Interface (NSI) protocol, the Network Service Agent (NSA), , the supporting topology model, and a set of basic services. These concepts are assigned to a conceptual Service Plane. The Network Service Interface (NSI) architecture describes an environment within which network capabilities and components are abstracted and manipulated as “network resources.” Within the NSI architecture, network resources can be selected, allocated, interrogated, and manipulated by software agents on behalf of networked applications. Network resources and capabilities are presented to the consumer through a set of “network services.” The Network Service Interface defines a framework for providing these services.

Network Services are delivered by the capabilities of participating operators. They will make use of a range of network related functional capabilities such as topology sharing, path finding, resource reservation, hardware provisioning, and other ancillary services and functions.

The NSI Architecture is described in more than one document. This informational document describes the NSI Framework, this is a flexible architecture designed to support many Network Services. In addition, each Network Service is described in its own recommendation document.

Version 1.0 of these NSI supports only one Network Service – the Connection Service.

# NSI architecture

## Network services

The NSI protocol is designed to deliver Network Services. Network services are built from the network resources available to the operator of the Network. These resources may cover a range of technologies such as wavelength delivered using ROADMs, TDM on cross-connect equipment or packets over switches among others. The NSI framework is designed to support a wide range of Network Services; version 1.0 of the NSI includes only the Connection Service, however other services are envisioned.

Service requests may originate from an application or grid middleware or a network operator. A service can be requested by any application that has implemented an NSI compliant agent and interface. Similarly, the request can be delivered by any network operator who has implemented an NSI compliant agent and interface.

The specific details of the services offered by a given agent are scoped by the Service Definition (SD). This sets the scope of the service and identifies any parameter that need to be completed for the request to be fulfilled.

The NSI protocol deals with an abstracted model of transport services. This abstraction reduces or hides many of the real-world complexities of delivering a particular transport service. For example, Network Services may manipulate and present these as abstracted resources based on a complex data transport resources. This abstract concept is a simplified and convenient means of presenting the key functional aspects of the service object while hiding most or all of technical details that are in general not relevant to the application.

The abstraction of a service presents a basic set of service primitives. These service primitives form a set of commands that allow the requestor set up and manage a service.

Each service request will result in the allocation of a service id and the creation of a new service instance. The responsibility for allocating the service id lies with the Requester NSA.

In the remaining part of this section, the architectural components that make up the NSI framework are described.

## The Network Service Interface

The Network Service Interface (NSI) provides secure and reliable sessions for service related communication between two NSAs. An instance of the NSI exists at the boundary between two communicating software agents: the Requester NSA and the Provider NSA. These agents interact to realize the delivery of a Network Service intrinsic to the network infrastructure. In this model, the Requester NSA requests some service, and the Provider NSA attempts to deliver it (see Figure 1).

Figure 1: NSI interface

Change this diagram to refer to requestor and provider agents?

## The Network Service Agent

The NSA is central to the NSI architecture since all NSI processes are invested in the Network Service Agent (NSA).

The NSA assumes three possible roles – Requester,Provider and Federating NSA. As a Requester, the NSA requests network resources and as a Provider it delivers these network resources to create a service. The Network Service Agent may at times act as a requester over one NSI interface while acting as a provider at a different NSI interface. A Federating NSA acts as a gateway to other providers, in this role the NSA can forward requests to other Provider NSAs. These modes are shown in Figure 2.

The NSA can support many different NSI Service types. Each service type will have many service instances; these instances are created in response to a service request. For example, Figure 2 depicts two instances of a Network Service.

The processes required to deliver the NSI service has two parts; a part that resides in the Requester NSA and a part that resides in the Provider NSA. In Figure 2 these are shown as the green ‘Network Service’ instances.

The NSI architecture allows many Network Service Agents (NSAs) to be interconnected with NSI interfaces. This is done with the use of a Federating NSA; examples of federation of networks are described in paragraph 2.7.

Figure 2: Network Service Agent modes

Also present in the NSA, but not shown, are some additional supporting functions such as path-finding function or an NSA forwarding look up service. These may local or remote, the definition of these functions is out of scope of the NSI.

### The Network Resource Manager

The provider and Federating NSAs may incorporate a Network Resource Manager (NRM). The NRM manages the part the Network Service implemented locally; two examples of NSAs with NRMs are shown in Figure 3.

Figure : NRMs in an NSA

## NSI Sessions

The NSI Protocol provides an extensible framework for the definition and incorporation of network services. These NSI Services exist in the form of service specific agents known to, or incorporated within, the NSA.

The NSI protocol supports two types of secure session. A NSA-to-NSA session provides a common session between NSAs. The service-to-service sessions provide per NSI service instance session. This is depicted in the following diagram.

Figure : Transport Plane and Service Plane

## NSI service extensibility

The NSI Interface provides a common framework in which Network Services can be delivered. To achieve this aim, the NSI Architecture is extensible; it inherently supports the ability to add new Network Services as they emerge. Examples of anticipated services include a Topology Service to distribute topology information and a Connection Monitoring service. The Network Service Agents must support these services and functions in order to provide the integrated service envisioned.

Version 1.0 of the NSI supports only a Connection Service, this service allows connection-oriented circuits to be requested and managed.

## The NSI Service Plane

For the purposes of defining layering, this architecture assigns the NSI to a conceptual Service Plane. Here we define the Service Plane as incorporating participating NSAs and the associated NSI sessions between these NSAs. This is depicted in Figure 5.

In general, the NSI Service Plane relies on the capabilities of the Control Plane and/or Management Plane (not shown) to effect changes in the Transport Plane, where the control and management planes follow conventional definitions. The transport resources and the physical instance of the Connection reside on the Transport Plane.

Figure : Transport Plane and Service Plane

The NSI Service Plane operates on an abstracted representation of the transport plane resources. In this abstracted view, transport resources within a particular Network are treated as a single opaque object under the control of a Provider NSA. Representation of transport resources is described further in section 4.

Considering the example of the NSI Connection Service, the service managed on the NSI Service Plane is the ability to create a Connection, with a set of simple, well defined manipulation rules that operate within a network and are defined by the NSI. These abstractions are exchanged and processed in the NSI Service Plane. The NSI Service Plane contains information about owners, permissions, performance parameters, schedule, and Path information.

Connection should be thought of as an NSI Service Plane construct that binds high level service planning results (such as path selection, authorization, reservation, and scheduling) to the low level configuration and management information necessary to provision the transport physical instances that make up path in the transport plane.

## Hierarchical communications model and federation

The NSI protocol is intended to allow services to be delivered across a chain of multiple participating Networks. To facilitate this, flexible NSI message forwarding is supported. This section describes the communications models supported for NSI message handling.

The NSI message handling model is supports both the tree and chain model of path-finding as described in theConnection Service definition. It is important to note that this means that the NSA communications path is independent of any routing of connections on the transport plane.

No assumptions are made about the reachability of participating NSAs, an NSA may be directly reachable or reachable only via a gateway NSA. For instance, an arbitrary set of Networks may band together under NSI rules and peer exclusively with a single parent “Federating NSA”. The parent federation NSA may have no transport resources of its own – just those resources under management of the children NSAs. Service requests will flow along the trusted sessions hierarchically among NSAs, even if the transport connection itself takes a different route through the transport infrastructure.

Figure 6 shows an example of the hierarchical model of communications. In the case of a federation of NSA, the federating NSA becomes a communications parent for its child NSAs. An example of this is shown where NSA A communicates with NSA D via NSA B.

Figure a: Hierarchical communications model

It should be noted that in the case of highly meshed NSAs, a destination NSA may be reachable by more than one path. An example of this is shown in the next figure. v1.0 of the NSI places no constraints on how to forward NSI messages. For example NSI A wishing to control a resource at E may choose to do this via intermediate NSAs B or C.

\*\*\* This example will lead to a resource conflict it applied to a single service instance. I suggest that this architecture should be avoided by operators to prevent ambiguity. This is best done by sticking to a hierarchical architecture, i.e no meshing of NSI interfaces.\*\*\* Guy

Figure 6b: Complex communications model

# The NSI Protocol

## NSI Protocol overview

Network Services are delivered with the use of the NSI Protocol. This protocol defines the constructs, state machines, messages, and parameters associated with the NSI services model. An NSA, by definition, is an agent that implements the NSI Protocol. The details of this protocol are out of scope of this document; the NSI Protocol recommendation (GWD-R-XXX) provides a detailed definition of the NSI framework protocol. In addition, each Network Service has its own recommendation document.

The NSI Protocol requires a mechanism to provide: secure communication between authenticated NSAs.

.

## NSI messages

Each session is established between a pair of NSAs.

The NSI is based on a exchange of messages between the requestor and provider, the details are defined by the NSI service.

The NSI protocol is made up of NSI messages. Each message contains sufficient information to associate the message with a specific service instance and to identify the service primitive.

.

An NSI Message  allows identification of the Network Service type. (eg Connection Service, Topology Service, etc) and will allow identification of the Network Service version (v1.0, v2.0 etc). NSI Messages  include a mechanism to ensure that ordering is maintained in a NSI Message Thread.

An NSI Message  includes a mechanism to associate it with an NSI Message Thread to allow differentiation of message streams associated with simultaneous and asynchronous service functions occurring between pairs of NSAs

## NSI Service Instances and Primitives

The base NSI protocol handler recognizes NSI messages between NSAs. The protocol examines each message received for its Service Identifier and forwards that message to the appropriate service specific handler.

While the service and its associated Service Definition defines the full set of capabilities that are offered to requesters, the service instance defines one specific instance of the service.

Each NSI service defines a service instance which is an independent, uniquely identifiable deliverable unit of the service. The NSI Connection Service refers to a particular connection as a service instance. A topology distribution service may define an instance to be a particular topology graph, or a topology transaction such as a full dump or incremental update.

Each Network Service type includes set of service primitives. These primitives form a set of instructions that pass from the requester to the provider. In general, a service specific state machine allocated and associated with each service instance, and the service primitives drive the transitions of that state machine. A service primitive may require a sequence of messages or even its own state machine to affect an exchange of messages.

Service Instances are processed asynchronously with respect to other service instances. For example, one connection may transition from reserving, to scheduled, to in-service, to release at a vastly different speed than another connection established by the same service agent.

The network service instanced must be locally unique.

## NSI Service Definitions

Jerry S. to rework this to make it more generic to all service, not just connection services.

The concept of Service Definitions is introduced to allow operators to formally identify and define the characteristics associated with each service offering.

The Service Definition consists of a set of attributes that formally and explicitly define the complete scope of a service offering. In particular, the NSI Connection Service uses the Service Definition as a baseline set of parameters to bound the scope of the service that will be offered to requesters.

The Service Definition specifies the set of service parameters that completely specify a service instance. For example, the Service Definition might identify “capacity”, “mtu\_size”, and “maximum\_frame\_loss\_rate” as three aspects of the service. The Service Definition also describes the *range* of allowed values for each service parameter, and a default value can be specified. In the context of the previous example, the range of allowed values for the “capacity” parameter may be 50 mbps to 10 gbps in increments of 150 mbps. Or the “mtu\_size” may be defined to be 1500 Bytes to 9000 Bytes with a default of 1500. The parameters in the Service Definition form a kind of template that the service request must fill in. I.e. A service request must fill in the template with an explicit value - or a default value taken from the service definition - for each parameter of the service it is requesting. A service request is fully specified when all parameters associated with that service have been determined either by explicit user specification or by implicit default values found in the Service Definition. This fully specified request is then processed by the NSA and, if all service specifications can be satisfied, a service instance is created (and reserved).

The Service Definition is an integral component of the NSI architecture in that it is key to vetting service requests against the multi-domain service offerings encountered along a candidate path.

The Service Definition is a public document that can serve as a both a human readable guide to available service capabilities and a machine readable file that can be processed by automated agents in the NSI Architecture.

## Temporal aspects of NSI services

Services, in which resources are dynamically requested, reserved and provisioned, require temporal aspects to be understood and deterministic. For the purposes of advance reservation Connection Service process must maintain its own real-time clock, and it is necessary for these clocks to be aligned. (Move this to the connection service section?)

When resources are sought by a requester NSA from a provider NSA, a service instance is created and an identifier is assigned to that service instance. Then, according to the parameters of the request (i.e. its Service Definition), the provider NSA identifies and reserves a set of available resources which satisfy the request and associates them to the instance. The resources are provisioned and released at some point on the temporal axis. The time information and signaling are used to specify the time boundary of the requested connection in-service period. It is the responsibility of the Provider NSA to attempt to deliver the connection as close to the start and end times as it is able. It should be noted that this may have some uncertainty as typically the duration of the provisioning phase cannot be precisely predicted.

In the case of the Connection Service, advance reservation requests will specify the required resources and the provisioning start and end time. The request is processed by a scheduler, and the scheduler finds a set of resources available for the requested duration and allocates them to the request to create a reservation. If the scheduler cannot find an available set of resources which satisfies the request, the request is denied. This scheduling process is part of path finding in the connection service. A reservation database (i.e. calendar) should be maintained by the scheduler or resource managers, and referred and updated by the scheduler. The detail of advance reservation is covered in detail in the Connection Service recommendation.

## Trust and authentication in NSI

This section describes the approach taken to trust and authentication in the NSI protocol. The detailed mechanisms for providing security and authentication are not defined here; the protocol recommendation will define standard ways of delivering secure connections.

Each NSA establishes NSI sessions with one or more other NSAs . An NSA will know for example that it is physically connected to its neighbor NSA; it may also have an identifier for that neighbor. Two types of trust are identified; these are NSA-NSA and service-to-service, these are depicted in

NSI – NSI relationship: The NSI requestor and provider have a secure session.

Service – Service relationship: Here services share message sequences with each other. These message sequences must provide between the services. The method for doing this is not described in this recommendation. Standard methods are described in the protocol document. These include piggybacking trust on the NSI trusted connection and using separate ids and keys for the services.

Formalize terms: ‘NSI session’ and ‘NSI Service session’

There are other trust issues that may need to be supported by the NSI interface which are not considered in version one of the recommendation. There are sections in appendix xxyy give a slightly fuller description. The two issues are

Trust between attribute provider and policy server (attribute user), where the two are not in adjacent NSAs.

Trust between connection reservation authorized on the service plane by provider NSA and provisioned connections on control or data plane.

This is the end of the architecture document – remaining parts belong to the Connection Service document.

## NSI Service Plane error handling

Inder to rework this to make this connection-service non-specific, Connection service specific parts go in the connection service section. The mechanism for recoving state is a connection service specific issue

The NSI architecture is a distributed, multi-agent architecture which is designed to handle error cases in such a way as to ensure predictable and deterministic behavior. This section describes the basis of error handling for this protocol.

Network errors can be broadly categorized as soft or hard failures. A soft failure occurs when two NSA agents lose communication with each other or the NRM loses communications with the transport resources. A hard failure occurs if the NSA software crashes or the underlying server hardware fails. Such failures may cause a loss of information about state information in transit or not committed.

The NSI protocol incorporates mechanisms to recover to a consistent and predictable state after detecting an anomaly. The following architectural principles guide error handling and recovery:

Handling of failures should result in deterministic behavior that is user centric and oriented towards the service model, for example: A failure in the Service Plane should not affect connections that are provisioned and active in the Transport Plane. A failure in the Service Plane should not result in an incomplete service.

Recovery of Transport and/or Service Plane should not be reliant on external entities or mechanisms, for example: An NSA recovering from a hard failure error condition will not depend on peer NSAs to reconstruct its state.

Failures in the Service Plane can result in NSA state faults. Examples of Service Plane errors include: losing communication with an NSA, losing communication with the transport network, corruption/crash in the platform etc. These errors may result in service disruptions until these states can be synchronized, hence the NSI protocol and state machine design should account for such scenarios.

Figure : Local/Remote Failures

Inder to update this diagram, - explain that yellow box indicates object of interest.

Discussion of failure of supporting functions such as NRM or lookup services

Regardless of where the error originates, it is important that the NSA recover to a deterministic state. This means that both the user service state and the resource state should be consistent between NSAs.

The distributed model of servicing user requests using tree/chain model allows each NSA to assume the role of a requester or provider. When Service Plane failures occur, it is possible that an NSA will become entirely disconnected from other NSAs involved in a service instance. This scenario imposes a requirement on the NSA to have a linkage between its Requester and Provider Agent state machines to understand the impact of the failure on the service tree and recover from it. The state machines of Requester and Provider NSAs should be designed so the outcome of a distributed failure ends each state machine in a deterministic state.

# Representing network resources

Parts of section 4 belong in Architecture document:

What is a Network?

What is a Network Service ( now in section 2.1)

What is a Topology

Topology service is part of connection service or NSI architecture? Vangelis thinks that this is unique to the Connection service, is this correct?

## virtualization

put Freeks diagram here and add discussion of virtualization.

## Describing network topologies

A Network topology is an object-oriented representation of Network resources based on a network description language. The Network topology may be used to perform functions such as path-finding and resource reservation. Many languages or models have been proposed to describe networks, such as OGF NML or ITU-T G.805. These are designed to fully describe the components of a network.

The NSI defines a new topology description: the inter-Network topology this is concerned with describing the global interconnection of Networks, and the intra-Network topology concerned with the transport resources within the Network. The formal representation of traditional intra-Network topology is out-of-scope for the NSI. While the choice of representation of intra-Network topology is up to the operator, the inter-Network topology is described here.

The method by which the inter-Network topology is assembled out-of-scope for version 1.0 of the NSI. It is assumed that each NSA has access to this topology information, but no assumptions are made as to how this has been gathered. Later version of the NSI are planned to a topology exchange service.

Figure 8: Inter-Network Topology

Need to emphasize that service plane connections are not necessarily congruent with the transport plane connectivity.

Can Network advertize the internal connectivity between STPs?

Need to have separate terms to distinguish transport plane object (Network?) and its service plane representation (Domain?).

NSA reachability is an issue for connection service and is related to pathfinding.

Figure 8 depicts an inter-Network abstracted topology with Networks interconnected at Service Termination/Transit Points (STPs).

NSI inter-Network topology supports the administrative grouping of Transport Plane resources into a single topology object called a Network. Networks interconnect with other Networks via interconnected physical ports. These are modeled as STPs in the service plane, it is the responsibility of the NSAs to define a valid mapping to relate STPs with physical ports.

By aggregating detailed transport topology into a single Network, or by grouping several Networks together to form a federated Network object, the global network topology may be reduced substantially. How such a federation is implemented and the resulting simplified inter- Network topology map is out of scope for at least this version of NSI. Successful implementation for a particular deployment will allow Pathfinders to inexpensively compute coarse grained path(s) between any pair of networks. Each NSA along the candidate path is then consulted to reserve and confirm the resources. (For most of the NSI discussion (certainly for V1.0 inter-domain discussion) we assume each Network has a transfer function that is a simple non-blocking any-port-to-any-port crossbar switching function. Any other path constraints are managed by the Resource Manager.)

It is important to note here that the arrangement of NSI interfaces between NSAs is *not* necessarily congruent with the connectivity between Networks.

From a global perspective, hiding detailed transport topology within an opaque Network object substantially reduces the size and complexity of the topology information base. This has positive implications for coherence and convergence, for dynamic topology distribution, path finding efficiency, and for scalability in the global environment. It has the less desirable effect of reducing optimality – it becomes increasingly difficult to choose a resource efficient path. The trade off is an issue of pragmatism, and will be steered by best practices as the experience base improves.

## Service Termination Points

The NSI Architecture adopts a generalized notion of a Service Termination Point (STP). An STP names a topological location that acts as a concatenation point connecting two Connections. This is the junction between the ingress of a Connection in one Network and the egress of a Connection in the next Network.

A prerequisite for an STP is the existence of a physical connection between Networks. This preexisting capability (typically made up of a physical port on each network) can be advertised by each network to the NSA. Note that the choice about which resources to advertize is subject to local policy. Once advertised, these capabilities may be used by the pathfinding function of the NSA.

An STP capability can be a list of possible STP instances, or a more flexible representation like wildcard and constraints. For example, if there are 10 links (1, 2, 3, …, 10) and any two of these links can be aggregated, there are 90 possible STP instances (1-2, 1-3, …, 9-10). A wildcard like representation may be better than listing all of such STPs individually.

An STP can function as both an ingress point on one side and an egress point on the other. Two such connections that share a single STP in this way are said to be concatenated. These two concatenated connections then appear to the user payload as a single end-to-end transport plane data-path. In this way an STP becomes intermediate transit-point of a path or connection, i.e a routing point through which the connection must pass.

STPs may be uni or bi-directional and are typically associated with Ports in the topology. In the uni-directional case, the STP functions either as an ingress point or an egress point, this is defined by the flow polarity of the associated Port, and which side of the junction is the user side, and which side is the network side.

Two adjacent networks agree on the STP capability between the two networks in advance. The STP capability is the information which describes all possible STPs which can be instantiated between them. Constraints are also included in the information.

Once the STP capabilities are advertised, the STP needs to be dynamically generated for an inter-network connection over the STP to be instantiated.

Once allocated, an STP may have properties such as a framing, bandwidth and a VLAN id. Some of these properties may reflect the requirements specified in the service definition. Labeling (cf. fiber id, wavelength, VLAN id) and aggregation (cf. an STP may be provisioned by aggregating multiple switch ports) can be modeled as a property of STP.

To instantiate an inter-network connection, the requester NSA requests STPs with the same representation to the adjacent networks. Each network then keeps a calendar of reservations.

Figure : STPs

Should this be in architecture or Connection Service document?

Assume there are two networks, X and Y, and possible STP instances between X and Y are: v1, v2, v3 and I1, I5, I9. This information is advertised by X and Y in advance. To request an inter-network connection, the requester NSA requests:

• To network X: connection between somewhere to to\_Y:v1

• To network Y: connection between to\_X:v1 to somewhere.

Each network looks up its own calendar and check availability of v1. The availability may be different by the networks.

When the above intra-network connections are instantiated, they are inherently connected.

An STP is a symbolic reference, i.e. it is comprised of a parsable alphanumeric string containing two components: 1) a Network identifier string in the higher order portion, and 2) a local STP identifier in the lower order portion. An STP must always resolve to a specific topological Port object as defined in the NSI topology. It is an implementation decision as to how to map that NSI Port identification to the corresponding resource in the physical transport plane.

It is important to note that the NSI inter-network topology model is composed of Networks interconnected by STPs. It should be noted that this topology is neither a standard nor does it imply that an NSI implementation must adopt specifically any particular schema for its database in the code.

# Contributors

Joan A. García-Espín, I2CAT

Chin Guok, ESNET

Radek Krzywania, PSNC

Tomohiro Kudoh, AIST

John MacAuley, Surfnet

Takahiro Miyamoto, KDDI R&D Laboratories

Inder Monga, ESnet

Guy Roberts, DANTE

Jerry Sobieski, NORDUNET

Sebastien Soudan, Laboratoire de l'Informatique du Parallèlisme

John Vollbrecht, Internet2

# Glossary

Connection

A Connection is a conduit that transparently moves user information across a Network from an ingress point to an egress point. A Connection has a set of properties (for instance, capacity, or authorization, or start time).  These properties, and their allowed range of values, are defined by a service definition. A Connection instance is a particular Connection, identified by a Connection Identifier.

Connection Identifier

A Connection Identifier is a label unique to an NSI interface which can be used to identify a Connection for the purposes of request, instantiation and management.

Connection Service

A Connection Service is a service that allows a Requester NSA to request and manage a Connection from a Provider NSA

Control and Management Planes

The Control Plane and/or Management Plane are not defined in this document, but follow common usage.

Network  
A Network includes all of the transport resources that are managed by a single NSA.

Network Resource Manager (NRM)

The Network Resource Manager owns a particular set of transport resources and has ultimate responsibility for authorizing and managing the use of these resources.

Network Service

Network Services are the full set of services offered by an NSA. A Network Service is an abstract service that must be implemented by a concrete network service agent (NSA). Each NSA will support one or more Network Services. NSI verison 1.0 supports only one Network Service – the Connection Service.

Network Service Agent (NSA)

The Network Service Agent is a concrete piece of software that sends and receives NSI Messages.  The NSA includes a set of capabilities that allow Network Services to be delivered.

Network Service Interface (NSI)

The NSI is the interface between Requester NSAs and Provider NSAs.  The NSI defines a set of interactions or transactions between these NSAs to realize a Network Service.

Network Service Framework

The NSI framework describes a message based platform capable of supporting a range of Network services.

NSI Message

A NSI Message is a structured unit of data sent between a Requester NSA and a Provider NSA.

Path

A Path is an ordered list of Routing Objects.

Requester/ Provider NSA

An NSA acts in one of two possible roles relative to a particular instance of an NSI.  When an NSA requests a service, it is called a Requester NSA. When an NSA realizes a service, it is called a Provider NSA.

Routing Object

A Routing Object may include the following transport resources.

Service Definition

The Service Definition is the set of attributes associated with connection services (for instance, capacity, or authorization, or start time) and a range of allowed values for these attributes. Each Connection has an associated Service Definition instance.

Service Plane

The Service Plane is a plane in which services are requested and managed; these services include the Network Service. The Service Plane contains set Network Service Agents communicating using Network Service Interfaces.

Topology Rename

The Topology resides in the Service Plane.  The Topology describes both the physical resources and their interconnection as well as the non-physical groupings of various components

Transport Plane

The Transport Plane contains is the set of physical resources that transport user data through the network.  The Transport Plane forms the substrate over which Connections are allocated and provisioned.

# Intellectual Property Statement

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director.

# Disclaimer

This document and the information contained herein is provided on an “As Is” basis and the OGF disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

# Full Copyright Notice

Copyright (C) Open Grid Forum (2008-2010). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the OGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the OGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assignees.

# References