**Network Services Framework**

Status of This Document

This document provides information to the Grid community on the service interface between a requesting software agent and a 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 Services Framework describes a framework to support the request and management of Network Services; it allows an application or network provider to request Network Services from other network providers. This framework incorporates the interface, agent and associated services. 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 document and its partner Network Service recommendations make up the complete NSI specification.

Contents

[1. Context and Overview 2](#_Toc266711822)

[2. NSI framework 3](#_Toc266711823)

[2.1 Network Services 3](#_Toc266711824)

[2.2 The Network Service Interface 3](#_Toc266711825)

[2.3 The Network Service Agent 4](#_Toc266711826)

[2.3.1 The Network Resource Manager 5](#_Toc266711827)

[2.4 NSI Sessions 5](#_Toc266711828)

[2.5 NSI service extensibility 5](#_Toc266711829)

[2.6 The NSI Service Plane and Inter-Network model 6](#_Toc266711830)

[2.7 Hierarchical communications model and federation 7](#_Toc266711831)

[3. The NSI Protocol 8](#_Toc266711832)

[3.1 NSI Protocol overview 8](#_Toc266711833)

[3.2 NSI messages 8](#_Toc266711836)

[3.3 NSI Service Definitions 9](#_Toc266711837)

[3.4 Temporal aspects of NSI services 10](#_Toc266711838)

[3.5 Trust and authentication in NSI 10](#_Toc266711841)

[3.6 NSI Service Plane error handling 10](#_Toc266711842)

[4. Representing network resources 12](#_Toc266711843)

[4.1 Describing network topologies 12](#_Toc266711845)

[4.2 Service Termination Points and Service Demarcation Points 13](#_Toc266711846)

[4.3 The Connection Service 15](#_Toc266711847)

[5. Contributors 16](#_Toc266711848)

[6. Glossary 16](#_Toc266711849)

[7. Intellectual Property Statement 18](#_Toc266711850)

[8. Disclaimer 18](#_Toc266711851)

[9. Full Copyright Notice 18](#_Toc266711852)

[10. References 18](#_Toc266711853)

# 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 a framework for the NSI protocol to supports both of these contexts.

The Network Services Framework defines several key architectural elements: a Network, a Network Service, the Network Service Agent (NSA), the Network Service Interface (NSI), and the NSI Protocol. These elements exist in a notional network Service Plane. The framework describes an environment within which network objects are defined as manageable resources. Within the framework, these network resources can be selected, allocated, interrogated, and manipulated by software agents on behalf of requesting users.

Network resources and capabilities are presented to the consumer through a set of ‘Network Services’. The Network Services Framework presents a unified model for interacting with these services. Network services include the ability to create connections, to share topology information, and to do other services needed by a set of federated NSAs.

Federated Network Services are delivered by combining the capabilities of participating providers. To manage federated services, a range of network related functional capabilities such as topology sharing, path finding, resource reservation, hardware provisioning, and other ancillary services and functions are required. These may be formalized in future versions of the NSI protocol.

The NSI protocol is defined by a suite of documents. This informational document describes the NSI Framework. In addition, each Network Service is defined in its own recommendation document.

# NSI framework

## Network Services

The NSI protocol is designed to deliver Network Services. Network Services are services that can monitor, control, interrogate support the network resources available to the provider of the Network. Typically transport network resources will include a range of technologies such as wavelengths delivered using ROADMs, TDM on cross-connect equipment or packets over switches. The NSI framework is designed to support a wide range of Network Services.

An example of a Network Service is the Connection Service, this service is used to request and manage transport connections. Another example is the Topology Service; this is used to exchange topologies between NSAs.

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

Each service offered has an associated Service Definition (SD), this SD 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.

Each service is managed by an exchange of NSI messages between agents. These messages operate using a set of service primitives. These service primitives are the set of instructions 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 Network Services 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

## 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 can support many different NSI Service types. Each service type can have multiple service instances; these instances are created in response to a service request. For example, each NSA shown in Figure 2 includes two instances of a Network Service, these are depicted as the green ‘Network Service’ boxes.

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. This is the case in a Federating NSA acts as a gateway to other providers; in this role the NSA can forward requests to other Provider NSAs. Federations of networks are described further in section 2.7. These three modes of operation are depicted in Figure 2.

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 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 Figure 4.

Figure : NSI sessions

## NSI service extensibility

The Network Services Framework provides a common platform on 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.

## The NSI Service Plane

This architecture assigns the NSI to a notional Service Plane. Here we define the Service Plane as incorporating participating NSAs and the associated NSI sessions between these NSAs. The transport equipment (switches, X-connects etc) resides in the Transport Plane. This is depicted in Figure 5.

In general, the NSI Service Plane relies on the capabilities of the Control Plane and/or Management Plane 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

## Hierarchical communications model and federation

The Network Services Framework 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.

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 Federating NSA. The Federating 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.

Figure 6 shows an example of the hierarchical model of communications. In the case of a federation of NSAs, 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 : 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. The NSI protocol places no constraints on how to forward NSI messages. For example NSA A wishing to control a resource at E may choose to do this via intermediate NSAs B or C.

Figure : Complex communications model

The following is a legitimate scenario that could apply to the complex communications model shown in Figure 7:

1. The upper left Requester NSA requests a Connection X-Z from NSA A.
2. NSA A divides the request in two parts and creates two new requests:  
    - a request for Connection X-Y is sent to NSA B  
    - a request for Connection Y-Z is sent to NSA C
3. NSA E can provide both a part of Connection X-Y and Connection Y-Z. So both NSA B and NSA C forward their request to NSA E.  
    - NSA B sends a request for a part of Connection X-Y to NSA E  
    - NSA C sends a request for a part of Connection Y-Z to NSA E
4. NSA E provides the Connections of both requests independently and simultaneously.

# 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.

## NSI messages

The NSI protocol describes an exchange of NSI messages between the requestor and provider, the details of these messages are defined by the NSI service. Each message contains:

* Identification of the Network Service type. (eg Connection Service, Topology Service, etc)
* Identification of the Network Service version (v1.0, v2.0 etc).
* Identification of a specific service instance
* Identification of a message thread
* A service primitive.

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. The service and its associated Service Definition defines the full set of capabilities that are offered to requesters and 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. For example, 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.

An NSI Message also 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 Messages include a mechanism to ensure that ordering is maintained in a NSI Message Thread.

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.

Each service instance must have a locally unique identifier.

## NSI Service Definitions

The concept of Service Definitions is introduced to allow network providers 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. Any service which supports advance reservation must maintain its own real-time clock and it is necessary for the requester and provider clocks to be aligned.

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.

## 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 described in the NSI protocol recommendation.

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 - NSA-NSA and service-to-service, these are depicted in Figure 4.

NSA-to-NSA relationship: The Requestor and Provider NSAs establish a secure session between Agents.

Service-to-Service relationship: Secure sessions are established between the requester and provider parts of a Service. Standard methods for securing these sessions are described in the protocol document. These include piggybacking trust on the NSI trusted connection and using separate ids and keys for the services.

There are other trust issues that may need to be supported by the NSI interface; these are noted, but not addressed in this version of the framework:

* Trust between attribute provider and policy server (attribute user), where the two are not in adjacent NSAs.
* Trust between connection reservations authorized on the service plane by provider NSA and provisioned connections on control or data plane.

## NSI Service Plane error handling

The Network Service Framework is based on 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 resources 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

## 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 as an input to functions such as path-finding and resource reservation.

For the purposes of the Network Services Framework, two topologies are identified; these are the intra-Network and inter-Network topologies. Only the inter-Network topology is in-scope for the NSI protocol.

The *intra-Network* topology refers to the topology of the resources within a Network, where a Network is defined as the group of Network resources managed by a single operator and a single NSA. The network operator is expected to have a preexisting management or control system with its own method for network modeling. It is assumed that each NSA has access to its topology information, and no assumptions are made as to how this has been gathered or how it is represented. In other words, the method by which the intra-Network topology is represented is out-of-scope for the Network Service Framework. Many languages and models have been proposed to describe networks; some examples are OGF NML and ITU-T G.805, it is expected that these and others could be used.

The *inter-Network* topology refers to the topology of interconnected Networks. The inter-Network topology is only concerned with describing the way in which Networks are interconnected and an aggregated set of Network capabilities. This Network Service Framework defines a representation of the inter-Network topology that should be used by the NSI. This is referred to as the NSI inter-Network topology or the inter-Network topology.

The inter-Network topology describes objects known as Service Termination Points (STPs) which are the edge points of a Network. STPs are grouped into a single topology object called a Network. A Network may have an associated internal transfer function matrix between STPs; this matrix describes the aggregated connectivity inside the Network.

By aggregating detailed transport topology into a single Network, or by grouping several Networks together to form a Federating Network object, the global network topology may be reduced substantially. 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.

Each operator can advertize a set of STPs. An STP may contain a set of port labels, for example an Ethernet port may contain a range of available VLANs. Further, an STP may be grouped with an STP in an adjacent network to create a Service Demarcation Point (SDP).

Figure 9 depicts an example of an inter-Network topology. It shows an example of how a Networks and STPs/SDPs can be used to describe an aggregated representation of a conventional Network model such as OGF-NML.

Figure 9: Inter-Network Topology

From a global perspective, the use of the Intra-Network topology to aggregate detailed transport topology within a 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.

It should not be assumed that SDPs are congruent with the NSA reachability on the service plane. I.e. the presence of an interconnection between two Networks does not imply that a matching NSA-to-NSA session is available for these Networks.

## Service Termination Points and Service Demarcation Points

The NSI Architecture adopts generalized notions of Service Termination Points (STPs) and Service Demarcation Points (SDPs). An STP names a topological location that is the ingress/egress point of a Network. An SDP names a topological location that connects two Networks. For the purposes of the Connection Service, the SDP also forms the point at which Connections are concatenated. This is the junction between the ingress of a Connection in one Network and the egress of a Connection in the next Network.

### Service Termination Points

A prerequisite for an STP is the existence of a physical connection into a Network. This preexisting capability (typically made up of a physical port on a 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 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.

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, in this example 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.

STPs may be uni or bi-directional and are typically associated with Ports in the topology. In the uni-directional case, the STP/SDP 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.

### Service Demarcation Points

Two adjacent networks agree on the connectivity capability between the two networks. The process for this agreement is out-of-scope of the NSI. When two STPs in adjacent networks with matching capabilities are paired, the resulting object is known as a Service Demarcation Point (SDP). The SDP capability is the information that describes all possible SDPs that can be instantiated between them. Constraints are also included in the information.

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

Once allocated, an SDP 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. combining multiple switch ports) can be modeled as a property of SDP.

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

Figure : STPs and SDPs

Assume there are two networks, X and Y, and possible SDP 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.

It is important to note that the NSI inter-network topology model is composed of Networks interconnected by SDPs. 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.

## Representing Connections in the Intra-Network topology

The Network Services Framework supports many services. The first of these is the Connection Service. The purpose of this service is to manage Connections. A Connection is defined to be the connectivity between STP/SDPs. Connections may be concatenated at SDPs to create longer Connections.

Figure : Representing Connections

The following diagram depicts an example of a connection service between hosts, one internal to Network W and the other attached to an STP of Network Z. The Connection is created by requesting a Connection in Networks X, Y and Z. In Network W the Host in internally connected, this internal connection is out-of-scope for the NSI since it requires an Intra-Network knowledge and control. In Network Z the host is connected to an STP it may be reached directly by using an NSI connection to STP k.

An SDP can function as both an ingress point on one side and an egress point on the other. Two such connections that share a single SDP 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 SDP becomes intermediate transit-point of a path or connection, i.e a routing point through which the connection must pass.

# 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 Services

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.

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 Services Framework

The Network Services 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 Demarcation Point (SDP)

Two STPs are grouped in adjacent Networks to form a Service Demarcation Point (SDP); this is a point that allows Connections to be Concatenated.

Service Termination Point (STP)

An STP is an edge point in a Network, i.e. an edge port.

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

The Topology resides in the Service Plane.  The Topology describes 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 transport equipment and associated resources that carry user data through the network.

# 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