**NSI Service Definitions v1.0 draft**

**The Service Definition Overview**

A “Service Definition” describes, in very formal and declarative terms, the service being offered by a service provider to the consumers of that service.

As network service providers move beyond a single IP service, it becomes necessary to differentiate between services and what they offer. Formal Service Definitions define the scope of each service being offered.

The theory behind SD provides a generalized mechanism to:

1. clearly describe and bound all service attributes
2. provide a “template” for qualifying service requests to fully specify all attributes of a service instance.
3. enable “service matching” to gauge interoperability of services across different networks.

The Service Definition (SD) is a machine readable textual document that identifies each attribute of the service and the range of values that attribute will allow.

A simple example [pseudo-code] Service Definition is described below:

Service Name=”Ethernet Transport Service”

Description=”

ETS transports the payload section of tagged ethernet frames from the Originating (ingress) STP to the destination (egress) STP. Bandwidth can be requested anywhere from 1 to 1000 MBps. Bookahead reservations can be requested for anytime between the time the request is submitted to 10 years in advance. The duration can be from 15 minutes up to a year. The lost/dropped frames will not exceed one frame in 10,000,000. Note: These are unidirectional connections. “

{

OrigSTP := NSISTP(OrigSTP)==True; /\* is this a real STP? \*/

DestSTP := NSISTP(DestSTP)==True;

Bandwidth := 1..1000 Mbps, default=10 Mbps;

AccessFraming:= 802.1, /\* payloads in untagged frames \*/

802.1q, /\* payloads in tagged frames \*/

802.1ah,/\* tagged frames are the payload \*/

default=802.1q; /\* only tagged payloads \*/

StartTime := now()..now()+10yr, default=now();

EndTime := now()+10yr, default=StartTime+15 min;

Auth := eduroam(), local();

Policed := True;

MaxFS := =< 9125 bytes; /\* Frame Size or MTU \*/

MaxBS := =< 9300 B; /\* Burst Size, at least frame size… \*/

MaxFLR := =< 8E-8; /\* Frame Loss Rate 1per10M= error’d sec \*/

MaxCES := 3; /\* Max Consecutive Err Secs = Failure Alarm \*/

}

(Note: the pseudo language used above is just for illustrative purposes – any suitable specification can be used for internal NSA processing. In the long term, a common description format (ala XML and XML Schema) will allow inter-operable automated distribution and processing.)

Incorporating formal Service Definitions in the Network Service Interface accomplishes several things:

First, it clearly defines the service attributes that define and bound the service. In the example above:

* The connection endpoints OrigSTP and DestSTP must be valid NSI service termination points, and are required (i.e. no default value in the Service Definition).
* The user can select a particular access framing from a closed set of options. By agreed convention, these symbolic labels indicate the payload to be carried in the connection. These labels should be well documented in the Service Definition to insure all networks offering this service and all requesters requesting these services clearly and objectively understand what each symbolic label means.
* The user’s ingress frames can be any size up to 9125 bytes. Note that the access framing and the payload are not necessarily the same.
* If the number of lost or dropped frames exceeds 1 in 10 million, an “errored second” is logged. When 3 consecutive error’ed seconds occur, an alarm is raised.

This SD provides a clear agreement between the service requester and the network provider as to what the service instance will provide.

By stating specifically what the service can provide, and committing to provide a service instance meeting the specific service parameters found in the request, there is clear delineation between a properly performing service instance and one that is *not* meeting the committed performance levels. Unlike conventional best-effort services where any experienced performance meets the somewhat low commitment of “I made my best effort to deliver your packets promptly and intact.”, SDs enable a very clear service level expectation and fault threshold. The service parameters in the confirmed service request establish a contract of sorts between the requesting agent and the provider agent.

The second aspect of the SD is what it *doesn’t* define. Everything not explicitly specified in the Service Definition is explicitly *undefined.* I.e. if some aspect of a service is not explicitly identified and defined in the Service Definition, the service provider makes no commitment regarding that service attribute. For instance, in the above example, there is no characteristic specifying prioritization, protection, latency bounds, etc. Therefore, the user has no basis to expect any particular behavior regarding these aspects of the service. Indeed, this does not mean these properties are not present, just that there is no commitment on the part of the network to provide any particular guaranty regarding those service attributes. The implication is that some or all such characteristics may be present on one service instance, and not present on another, or they may change. (One can imaging protection switching could affect latency. But such switching should still conform to the Max Consective Errored Seconds attribute that *is* explicitly defined.)

Third, the SD provides a template for the NSA to consult to insure that all service attributes are fully and properly specified in the service request. Formally, a service request must fully specify all attributes associated with a service before the network can allocate the resources. The set of attributes that must be specified is the set of attributes defined in the SD. The SD defines which attributes must be explicitly specified by the requester (i.e. they have no default value specified in the SD), and which may be omitted by the requester (the SD specifies a default value for the attribute.) The SD specifies the range of valid values, or the choice of values an attribute may specify.

The Service Definition does not confirm or reject a request – it simply states the [possible] attributes of a service instance. A user may still request a set of attribute constraints that cannot be met end to end, for instance due to resource exhaustion. The SD provides clarity in the service request specification, and the fault threshold associated with a committed service instance.

Traditional models of circuit services and control planes adopt a single very tightly defined data plane technology, and then hard code the service attributes (signaling parameters) into the control plane protocols. Emerging multiprotocol services will need to leverage many data transport technologies and will need to recognize a wider array of service attributes. An extensible service definition framework allows a single service plane protocol suite (such as NSI) to present different services and different transport capabilities to the user. A different control plane protocol is not required every time a new service variant is introduced. The NSI Service Definition separates Connection Service Protocol and the Connection Services themselves. The NSI Framework and the CS protocol operate on the base abstracted notion of a “connection” allowing particular connection services to be defined separately from the provisioning protocol itself. The Service Definitions allow the service plane to quickly incorporate new service specific parameters.

Finally, Service Definitions, by merit of the fact they clearly define the service specifics, provide a means for network management and engineering teams to understand exactly what must be engineered into the infrastructure in order to support that service.

**The *Common* Service Definition**

In the NSI inter-domain context, several network operators define a single “common Service Definition” (CSD) for a service that they all want to deploy. They can each then take that CSD back home and engineer it into their respective networks and be assured then that the services they deploy are consistent and inter-operable across their domains – thus substantially increasing inter-operability and reach of the end-to-end services.

A Common Service Definition is a sort of ideal service: ideally, all participating networks will conform fully to the CSD. But perhaps not all networks can meet the full range of the service attributes defined in the CSD. (for instance some aspects of a CSD may require forklift upgrades to meet the full capability.) The CSD allows networks to offer certain variations on the service without sacrificing interoperability. For instance, perhaps some networks can only offer 1 Gbps maximum capacity, while other may offer 10 Gbps. This difference is easily specified in the local Service Definition and yet the service remains compatible across the domains. Those characteristics that a network cannot meet can be adjusted in their local SD to insure that potential users know which aspects are inter-operable and which are not.

While a network can adjust the range of a service attribute, it cannot chose to ignore the attribute altogether and remain compatible – it must provide all of the service attributes to some degree. When two networks compare their service definitions, the intersection of the service attributes defines the interoperability range of the two services. I.e. the intersection defines the range of service requests that can be provisioned across both networks. The larger the common ranges, the more likely service requests will be compatible with both service offerings. So it behooves the service providers to maximize the degree to which service attributes meet or exceed the ranges specified in the CSD. Doing so provides broader more interoperable services.

For example: the Ethernet MTU varies by switch equipment and configuration. The ideal CSD may specify 9180 as the MTU, but some networks may not be able to support the full 9180 ideal MTU. A network supporting only 1500 byte MTU can still inter-operate on connection requests that don’t need a large MTU even though that same network will reject requests requiring a larger MTU. But a network cannot elect to delete the MTU altogether from a conforming Service Definition, but it could reduce it to 1500 Bytes. Indeed, two such slightly differing services could still successfully meet most service request.

Finally, within the context of the NSI Framework, processing of Service Definitions requires a certain enhanced sophistication on the part of the Network Service Agent. The NSA must be able to do two things: First, it must be able to find and parse the Service Definition and use it to qualify service requests, insuring that all request parameters are fully specified. Second, the NSA must then be able to intelligently relate those requested service parameters to the attributes of the resources that are available in the network to choose a candidate path.

The first step of qualifying the request to insure it lies within the service envelope is relatively easy.

The second step of relating the service characteristics to the available resources is more complex. For instance, if the Service Definition specifies a “Capacity” characteristic with a service range of 10 to 1000 Mbps (megabits per second), and a conforming request is received asking for 500 Mbps, then the NSA resource allocator must be able to intelligently search the topology database (the resources) for those path segments that have an available capacity of 500 Mbps, tentatively reserving resources and releasing them as potential path segments are examined.

This process that compares the service definitions to the resource topology DB is a more advanced capability that requires more study and work before it can be mapped to NSA features. Fortunately, the NSI framework relegates much of this resource specific allocation to the NRMs, which simplifies the NSI layer considerably. Indeed, the NSI layer can do first stage coarse grained inter-domain path selection based solely on intersecting Service Definitions found along the path with the request profile, and then consulting the specific NRMs along the candidate path so constructed will perform the fine grained reservation.

In NSI CS v1.0, the Service Definitions are only used to qualify service requests.

**NSI Service Definitions v1.0**

NSI v1.0 will use the Service Definition to act as a template to fully specify a service request. The basic form of a Service Definition is:

<serviceDefinition>

<serviceNameDecl>

<serviceDesc>

<serviceAttributeList>

</serviceDefinition>

Service attibutes are of the general form:

<AttrName> <AttrValue>

For NSI v1.0, the Service Definition will be encoded as an XML document conformant to an XML Schema Definition. An example of the Service Definition XML follows:

<?xml version="1.0" encoding="UTF-8"?>  
<nsi:schema xmlns:nsi="file://local/Users/jerry/work/NORDUnet/OGF NSI-WG/nsi-SD-v1.0.xsd">  
 <nsi:ServiceDefinition>  
 <nsi:ServiceName>EthernetTransportService</nsi:ServiceName>  
 <nsi:ServiceDesc> Ethernet framed payloads are carried.

</nsi:ServiceDesc>  
 <nsi:ServiceAttributeList>  
 <nsi:Attribute>  
 <nsi:AttrName>OrigSTP</nsi:AttrName>  
 <nsi:STPname/>  
 </nsi:Attribute>  
 <nsi:Attribute>  
 <nsi:AttrName>Bandwidth</nsi:AttrName>  
 <nsi:Rangeof>  
 <min>0</min>  
 <max>1000</max>  
 <step>10</step>  
 <default>10</default>  
 </nsi:Rangeof>  
 </nsi:Attribute>  
 <nsi:Attribute>  
 <nsi:AttrName>Protection</nsi:AttrName>  
 <nsi:Choiceof>  
 <nsi:AttrValue>none</nsi:AttrValue>  
 <nsi:AttrValue>transport</nsi:AttrValue>  
 <nsi:AttrValue>access</nsi:AttrValue>/>  
 </nsi:Choiceof>  
 </nsi:Attribute>  
 </nsi:ServiceAttributeList>  
 </nsi:ServiceDefinition>  
</nsi:schema>

|  |  |
| --- | --- |
| ServiceDefinition | Comprises the specifications that fully and completely define an NSI “Service” |
| ServiceName | A string that distinguishes the service described in this SD from other services that may also be offered in the same NSI network. The name can be any length and may contain any characters printing characters. The name string carries no encoded information within an NSI context. It is simply a string. special characters are explicitly disallowed. Multi-lingual service names may be supported |
| ServiceDesc | A textual description of the service, or alternatively a URN pointing to a file containing the textual description. |
| ServiceAttributeList | A list of attributes that fully specify the service being offered. |
|  |  |
|  |  |
|  |  |
|  |  |

The NSI Service Definition v1.0 XML document is governed by the NSI Service Definition XSD

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<xsd:annotation>

<xsd:documentation xml:lang="en">

NSI Service Definition Schema

</xsd:documentation>

</xsd:annotation>

<xsd:complexType name=”ServiceDefinition” type=”SDType”>

<xsd:element name=”ServiceName” type=”xsd:string” />

<xsd:element name=”ServiceDesc” type=”xsd:string” />

<xsd:

(…in progress..)