**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 Best Effort service, it becomes necessary to differentiate between services and what they offer. Formal Service Definitions define the characteristics and 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 and fully specifying parameters of a service request
3. enable “service matching” to gauge interoperability of services across different and/or requests.

The Service Definition (SD) is a text document that identifies each characteristic or attribute of the service and specifies the range of values for each such attribute.

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 SD).
* The user can select a particular access framing from a closed set of options. By agreed convention, these labels indicate the payload to be carried in the connection. These labels should be well documented in the CSD to insure all deployments know specifically and clearly what each 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 a specific combination of service parameters in a particular 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 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 attribute is not explicitly identified and defined in the Service Definition, the service provider makes no commitment regarding that service characteristic. 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 nay 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 instance before the network resources can be reserved. These 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 SD 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. The SD provides clarity in specification, and fault thresholding in service.

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 integrate many types of data transport technologies with differing service attributes into a portfolio of services. A common extensible service plane is needed that can adapt to different services and different transport capabilities so that a new control protocols are not required every time a new service variant is introduced. The NSI Framework separates Connection Service Protocol and the Connection Services themselves. The NSI Framework and the CS protocol operate on the abstract notion of a “connection” allowing particular services to be defined and modified 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.

In the NSI inter-domain context, several network operators can get together to define a single “common Service Definition” (CSD) for a service that they 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 have deployed are consistent and inter-operable across the many domains – thus substantially increasing the reach and inter-operability of the end-to-end service.

A Common Service Definition is a sort of ideal service. Perhaps not all networks can meet all the service characteristics in the CSD. 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 its service characteristics’ values, it cannot elect to ignore a characteristic altogether – 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 of the two services. The larger the common attribute 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 9250 as the MTU, but some networks may not be able to support the full 9250 ideal MTU. A network cannot elect to delete the MTU from a conforming Service Definition, but it could reduce it to 1500 Bytes. A network supporting only 1500 byte MTU can still inter-operate on connection requests that don’t need a large MTU even though they will reject requests requiring the larger MTU. Indeed, this could still allow most requests to succeed.

**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> <sAttributeList>

Service attibutes are of the general form:

<sAttrName> “:=” <valueSpec> “;”

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

<serviceDefinition>

 <serviceName>Ethernet Transport Service</serviceName>

 <serviceDesc>

ETS transports the payload section of tagged ethernet frames from the Originating (ingress) STP to the destination (egress) STP.

 </serviceDesc>

 <sAttributeList>

 <sAttr>

 <sAttrName>OrigSTP</sAttrName>

 <sAttrDesc>

 Ingress point for the connection. User required.

 </sAttrDesc>

 <sAttrNSIname></sAttrNSIname>

 </sAttr>

 <sAttr>

 <sAttrName>DestSTP</sAttrName>

 <sAttrDesc>

 Egress point for the connection. User Required.

 </sAttrDesc>

 <sAttrNSIname></sAttrNSIname>

 </sAttr>

 <sAttr>

 <sAttrName>AccessFraming</sAttrName>

 <sAttrDesc></sAttrDesc>

 <sAttrValueList>

 <sAttrVal>.1</sAttrVal>

 <sAttrVal>.1q</sAttrVal>

 <sAttrVal>.1ah</sAttrVal>

 <sAttrDef>.1q</sAttrDef>

 </sAttrValueList>

 </sAttr>

 <sAttr>

 <sAttrName>Bandwidth</sAttrName>

 <sAttrDesc></sAttrDesc>

 <sAttrRange>

 <sAttrRangeMin>1</sAttrRangeMin>

 <sAttrRangeMin>1000</sAttrRangeMin>

 </sAttrRange>

 </sAttr>

 </sAttributeList>

</serviceDefinition>

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..)