Schema documentation for NSI-CS-SD-Ethernet Transport Service version 1.0.xsd

april 10, 2011

Table of Contents

Namespace: "" 1
Schema(s) 1
Main schema `NSI-CS-SD-Ethernet Transport Service version 1.0.xsd` 1
Element(s) 1
Element `ServiceDefinition / ServiceName` 1
Element `ServiceDefinition / ServiceAttributeList` 2
Element `serviceAttributeListType / Orig` 2
Element `serviceAttributeListType / Dest` 2
Element `serviceAttributeListType / Bandwidth` 3
Element `serviceAttributeListType / MaximumBurstSize` 3
Element `serviceAttributeListType / ScheduleStart` 4
Element `serviceAttributeListType / ScheduleEnd` 4
Element `serviceAttributeListType / AuthenticationInfo` 5
Element `serviceAttributeListType / AuthenticationInfo / simpleUser` 5
Element `serviceAttributeListType / AuthenticationInfo / eduRoam` 6
Element `serviceAttributeListType / AuthenticationInfo / x.509cert` 6
Element `serviceAttributeListType / FixedAttrShapingWindow` 6
Complex Type(s) 7
Complex Type `ServiceDefinition` 7
Complex Type `serviceAttributeListType` 7
Simple Type(s) 9
Simple Type `serviceNameType` 9

Namespace: ""

Schema(s)

Main schema `NSI-CS-SD-Ethernet Transport Service version 1.0.xsd`

Namespace	No namespace

Annotations	`NSI Service Definition Schema THis is the template schema describing a Ethernet connection service. The Service definition has two key components: The SD Name, and the SD Service Attribute List. Information describing the service itself should be provided with each service attribute defined in the Attribute List. If some aspect is not clearly deliniated in the service attributes, then additional specification should be added to the SD. An aspect not explicitly set out in the Service Attributes *is explicity undefined*. The Attribute List is a set of Attribute names, and the type and range of each attribute that is allowed. Service deployments may vary slightly in the range of values they support for each attribute, but the attribute set must be the same. See the NSI CS Protocol Specification for further info.`
	`THe Service Name is an agreed upon textual title for the service. "Etehrnet Transort Service", or "General Connection Service Version 1.0" All leading and trailing whitespace characters are removed and whitespace within the title is collapsed to a single space. The Service Name is used to distiguish different services definitions.`

Properties	attribute form default: `unqualified` element form default: `unqualified`

Element(s)

Element `ServiceDefinition / ServiceName`

Namespace	No namespace
Diagram	
Type	serviceNameType
Properties	content: `simple`

Facets	whiteSpace collapse minLength 4 maxLength 32
Source	`<xsd:element name="ServiceName" type="serviceNameType"/>`

Element ServiceDefinition / ServiceAttributeList

Namespace	No namespace
Diagram	
Type	serviceAttributeListType
Properties	content: complex
Model	Orig , Dest+ , Bandwidth , MaximumBurstSize , ScheduleStart , ScheduleEnd , AuthenticationInfo , FixedAttrShapingWindow
Children	AuthenticationInfo, Bandwidth, Dest, FixedAttrShapingWindow, MaximumBurstSize, Orig, ScheduleEnd, ScheduleStart

Instance	`<ServiceAttributeList>`
	  `<Orig>{1,1}</Orig>`
	  `<Dest>{1,unbounded}</Dest>`
	  `<Bandwidth>{1,1}</Bandwidth>`
	  `<MaximumBurstSize>{1,1}</MaximumBurstSize>`
	  `<ScheduleStart>{1,1}</ScheduleStart>`
	  `<ScheduleEnd>{1,1}</ScheduleEnd>`
	  `<AuthenticationInfo>{1,1}</AuthenticationInfo>`
	  `<FixedAttrShapingWindow>{1,1}</FixedAttrShapingWindow>`
	`</ServiceAttributeList>`

Source	`<xsd:element name="ServiceAttributeList" type="serviceAttributeListType"/>`

Element serviceAttributeListType / Orig

Namespace	No namespace
Annotations	The Orig atribute specifies an NSI endpoint name that identifies the ingress point of the connection. The NSI endpoint name provided is a reference into th etopologyDB where the physical characteristics of this endpoint can be found and validate against the service capabilities, e.g. the Ethernet service should only recognize Ethernet capable endpoints.
Diagram	
Type	restriction of xsd:string
Properties	content: simple minOccurs: 1 maxOccurs: 1
Facets	pattern [a-zA-Z0-9$]+:[a-zA-Z0-9$]+

Source	`<xsd:element name="Orig" minOccurs="1" maxOccurs="1">`
	  `<xsd:annotation>`
	    `<xsd:documentation>The Orig atribute specifies an NSI endpoint name that identifies the ingress point of the connection. The NSI endpoint name provided is a reference into th etopologyDB where the physical characteristics of this endpoint can be found and validate against the service capabilities, e.g. the Ethernet service should only recognize Ethernet capable endpoints.</xsd:documentation>`
	  `</xsd:annotation>`
	  `<xsd:simpleType>`
	    `<xsd:restriction base="xsd:string">`
	      `<xsd:pattern value="[a-zA-Z0-9$]+:[a-zA-Z0-9$]+"/>`
	    `</xsd:restriction>`
	  `</xsd:simpleType>`
	`</xsd:element>`

Element serviceAttributeListType / Dest

Namespace	No namespace
Annotations	The Dest atribute specifies an NSI endpoint name that identifies the egress point of the connection. The NSI endpoint name provided is a reference into the topologyDB where the physical characteristics of this endpoint can be found and validated against the service capabilities, e.g. the Ethernet service should only recognize Ethernet capable endpoints.

Diagram	
Type	restriction of xsd:string
Properties	content: simple minOccurs: 1 maxOccurs: unbounded
Facets	pattern [a-zA-Z0-9$]+:[a-zA-Z0-9$]+

Source	<xsd:element name="Dest" minOccurs="1" maxOccurs="unbounded">
	  <xsd:annotation>
	    <xsd:documentation>The Dest atribute specifies an NSI endpoint name that identifies the egress point of the connection. The NSI endpoint name provided is a reference into the topologyDB where the physical characteristics of this endpoint can be found and validated against the service capabilities, e.g. the Ethernet service should only recognize Ethernet capable endpoints.</xsd:documentation>
	  </xsd:annotation>
	  <xsd:simpleType>
	    <xsd:restriction base="xsd:string">
	      <xsd:pattern value="[a-zA-Z0-9$]+:[a-zA-Z0-9$]+"/>
	    </xsd:restriction>
	  </xsd:simpleType>
	</xsd:element>

Element serviceAttributeListType / Bandwidth

Namespace	No namespace
Annotations	Bandwidth := The Megabits per second (Mbps) requested for a guarantied capacity connection. A zero value indicates a "best effort" connection with no guarantied capacity is being requested. Note - this does not commit that any user data will be delivered! Only that a best effort will be made for zero-cap connections. Note similarly, that a non-zero bw value does not imply that offered load in excess of the guaranty will be delivered at all. The nuanced difference in a zero vs non-zero bw request is that the service does commit to provide a best-effort to deliver frames associated with a connection that has no bw guarranty, where as no such commitment is made for connections that have requested a guaranteed capacity. This means a guaranteed bw connection may have its excess load strictly policed at ingress and dropped, or quietly buffered and delivered if possible...This service deinfition makes no commitment and leaves it to individual netwoprks to resolve internally.
Diagram	
Type	restriction of xsd:integer
Properties	content: simple
Facets	maxInclusive 1000 minInclusive 1

Source	<xsd:element name="Bandwidth">
	  <xsd:annotation>
	    <xsd:documentation>Bandwidth := The Megabits per second (Mbps) requested for a guarantied capacity connection. A zero value indicates a "best effort" connection with no guarantied capacity is being requested. Note - this does not commit that any user data will be delivered! Only that a best effort will be made for zero-cap connections. Note similarly, that a non-zero bw value does not imply that offered load in excess of the guaranty will be delivered at all. The nuanced difference in a zero vs non-zero bw request is that the service does commit to provide a best-effort to deliver frames associated with a connection that has no bw guarranty, where as no such commitment is made for connections that have requested a guaranteed capacity. This means a guaranteed bw connection may have its excess load strictly policed at ingress and dropped, or quietly buffered and delivered if possible...This service deinfition makes no commitment and leaves it to individual netwoprks to resolve internally.</xsd:documentation>
	  </xsd:annotation>
	  <xsd:simpleType>
	    <xsd:restriction base="xsd:integer">
	      <xsd:minInclusive value="1"/>
	      <xsd:maxInclusive value="1000"/>
	    </xsd:restriction>
	  </xsd:simpleType>
	</xsd:element>

Element serviceAttributeListType / MaximumBurstSize

Namespace	No namespace

Annotations	MaximumBurstSize := the total size of contiguous frames that can be transmitted on this circuit before ingress policing will begin dropping frames. This parameter allows the source to transmit a stream of pdus up to this size and expect the network to buffer them and transmit them over time. Implicitly associated with this parameter is the buffer drain rate which is assumed to be a "fine grained constant rate" output flow at the requested bandwidth. If the input queue excedes the MBS, the network is allowed to, and should be expected to, police the ingress flow and drop offending frames. This parameer is only meaningful for asynchronous packet based technologies such as Ethernet. the Maximum Burst Size is measured in units of MegaBytes (2^20 Bytes)
Diagram	
Type	restriction of xsd:integer
Properties	content: simple default: 100
Facets	maxInclusive 1000 minInclusive 0

Source

```
<xsd:element name="MaximumBurstSize" default="100">
  <xsd:annotation>
    <xsd:documentation>MaximumBurstSize := the total size of contiguous frames that can
 be transmitted on this circuit before ingress policing will begin dropping frames. This
 parameter allows the source to transmit a stream of pdus up to this size and expect
 the network to buffer them and transmit them over time. Implicitly associated with this
 parameter is the buffer drain rate which is assumed to be a "fine grained constant rate"
 output flow at the requested bandwidth. If the input queue excedes the MBS, the network
 is allowed to, and should be expected to, police the ingress flow and drop offending
 frames. This parameer is only meaningful for asynchronous packet based technologies such
 as Ethernet. the Maximum Burst Size is measured in units of MegaBytes (2^20 Bytes)</
xsd:documentation>
  </xsd:annotation>
  <xsd:simpleType>
    <xsd:restriction base="xsd:integer">
      <xsd:minInclusive value="0"/>
      <xsd:maxInclusive value="1000"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

Element serviceAttributeListType / ScheduleStart

Namespace	No namespace
Annotations	Schedule attributes are required. They are specified using the XML Schema format for data and time. Example: 2012-01-01T06:00:00
Diagram	
Type	xsd:dateTime
Properties	content: simple

Source

```
<xsd:element name="ScheduleStart" type="xsd:dateTime">
  <xsd:annotation>
    <xsd:documentation>Schedule attributes are required. They are specified using the XML
 Schema format for data and time. Example: 2012-01-01T06:00:00</xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

Element serviceAttributeListType / ScheduleEnd

Namespace	No namespace
Annotations	Schedule attributes are required. They are specified using the XML Schema format for data and time. Example: 2012-01-01T06:00:00
Diagram	
Type	xsd:dateTime
Properties	content: simple

Source

```
<xsd:element name="ScheduleEnd" type="xsd:dateTime">
  <xsd:annotation>
    <xsd:documentation>Schedule attributes are required. They are specified using the XML
 Schema format for data and time. Example: 2012-01-01T06:00:00</xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

Element serviceAttributeListType / AuthenticationInfo

Namespace	No namespace
Annotations	AuthenticationInfo defines mechanisms the service recognizes for request authentication. The authentication method specified in the Service Request must match one of the methods offered by the Service Definition. Further, the string presented as part of the method choice is present to an internal authentication model for evaluation.
Diagram	
Properties	content: complex
Model	simpleUser | eduRoam | x.509cert
Children	eduRoam, simpleUser, x.509cert

Instance	<AuthenticationInfo>
	  <simpleUser>{1,1}</simpleUser>
	  <eduRoam>{1,1}</eduRoam>
	  <x.509cert>{1,1}</x.509cert>
	</AuthenticationInfo>

Source	<xsd:element name="AuthenticationInfo">
	  <xsd:annotation>
	    <xsd:documentation>AuthenticationInfo defines mechanisms the service recognizes for request authentication. The authentication method specified in the Service Request must match one of the methods offered by the Service Definition. Further, the string presented as part of the method choice is present to an internal authentication model for evaluation.</xsd:documentation>
	  </xsd:annotation>
	  <xsd:complexType>
	    <xsd:choice>
	      <!-- NSA has a flat file with allowed "simple users" listed...good for testing -->
	      <xsd:element name="simpleUser">
	        <xsd:annotation>
	          <xsd:documentation>The simpleUser authentication is a string representing a user. The Provider Agent will look this up in an internally populated simpleUsers table. A match will validate the user. This authenitcation is inherently insecure and should only be used for testing. THis authentication method can be turned off by not providing simpleUsers to polulate the table, or by removing the simpleUser element from the service definition</xsd:documentation>
	        </xsd:annotation>
	        <xsd:simpleType>
	          <xsd:restriction base="xsd:string">
	            <xsd:minLength value="8"/>
	            <xsd:maxLength value="32"/>
	            <xsd:whiteSpace value="collapse"/>
	          </xsd:restriction>
	        </xsd:simpleType>
	      </xsd:element>
	      <!-- example of maybe another auth method... -->
	      <xsd:element name="eduRoam" type="xsd:string">
	        <!-- -->
	      </xsd:element>
	      <!-- This is intended to mimic current OSCARS style auth -->
	      <xsd:element name="x.509cert" type="xsd:string">
	        <!-- -->
	      </xsd:element>
	    </xsd:choice>
	  </xsd:complexType>
	</xsd:element>

Element serviceAttributeListType / AuthenticationInfo / simpleUser

Namespace	No namespace
Annotations	The simpleUser authentication is a string representing a user. The Provider Agent will look this up in an internally populated simpleUsers table. A match will validate the user. This authenitcation is inherently insecure and should only be used for testing. THis authentication method can be turned off by not providing simpleUsers to polulate the table, or by removing the simpleUser element from the service definition
Diagram	
Type	restriction of xsd:string
Properties	content: simple
Facets	whiteSpace collapse minLength 8

	maxLength 32

Source	<xsd:element name="simpleUser">
	  <xsd:annotation>
	    <xsd:documentation>The simpleUser authentication is a string representing a user. The Provider Agent will look this up in an internally populated simpleUsers table. A match will validate the user. This authenitcation is inherently insecure and should only be used for testing. THis authentication method can be turned off by not providing simpleUsers to polulate the table, or by removing the simpleUser element from the service definition</xsd:documentation>
	  </xsd:annotation>
	  <xsd:simpleType>
	    <xsd:restriction base="xsd:string">
	      <xsd:minLength value="8"/>
	      <xsd:maxLength value="32"/>
	      <xsd:whiteSpace value="collapse"/>
	    </xsd:restriction>
	  </xsd:simpleType>
	</xsd:element>

Element serviceAttributeListType / AuthenticationInfo / eduRoam

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="eduRoam" type="xsd:string"> <!-- --> </xsd:element>

Element serviceAttributeListType / AuthenticationInfo / x.509cert

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="x.509cert" type="xsd:string"> <!-- --> </xsd:element>

Element serviceAttributeListType / FixedAttrShapingWindow

Namespace	No namespace
Annotations	The Standard Shaping Window is a timing window mesured in milliseconds (10^-3 seconds) that defines the period over which all connections are configured in order to "smooth" out flows... This is a parameter the source of a connection can use to manage their offered traffic. For instance: A 200 Mbps flow on a 1 GE link consumes 20% of the link capacity. Therefore, that connection can expect to burst traffic for 20% * ShapingWindow. If the shaping window is 100 milliseconds, then the connection can expect that .2*100=20 millisecond bursts would be acceptable shaping for the flow. And every 100 miliseconds, the source could burst another 20 mils of data for that circuit.
Diagram	
Type	xsd:integer
Properties	content: simple default: 100
Source	<xsd:element name="FixedAttrShapingWindow" type="xsd:integer" default="100"> <xsd:annotation> <xsd:documentation>The Standard Shaping Window is a timing window mesured in milliseconds (10^-3 seconds) that defines the period over which all connections are configured in order to "smooth" out flows... This is a parameter the source of a connection can use to manage their offered traffic. For instance: A 200 Mbps flow on a 1 GE link consumes 20% of the link capacity. Therefore, that connection can expect to burst traffic for 20% * ShapingWindow. If the shaping window is 100 milliseconds, then the connection can expect that .2*100=20 millisecond bursts would be acceptable shaping for the flow. And every 100 miliseconds, the source could burst another 20 mils of data for that circuit.</xsd:documentation> </xsd:annotation> </xsd:element>

Complex Type(s)

Complex Type ServiceDefinition

Namespace	No namespace
Diagram	
Model	ServiceName , ServiceAttributeList
Children	ServiceAttributeList, ServiceName

Source	<xsd:complexType name="ServiceDefinition">
	  <xsd:sequence>
	    <xsd:element name="ServiceName" type="serviceNameType"/>
	    <xsd:element name="ServiceAttributeList" type="serviceAttributeListType"/>
	  </xsd:sequence>
	</xsd:complexType>

Complex Type serviceAttributeListType

Namespace	No namespace
Diagram	
Used by	Element ServiceDefinition/ServiceAttributeList
Model	Orig , Dest+ , Bandwidth , MaximumBurstSize , ScheduleStart , ScheduleEnd , AuthenticationInfo , FixedAttrShapingWindow
Children	AuthenticationInfo, Bandwidth, Dest, FixedAttrShapingWindow, MaximumBurstSize, Orig, ScheduleEnd, ScheduleStart

Source	<xsd:complexType name="serviceAttributeListType">
	  <xsd:sequence>
	    <!--**
	            Orig := Ingress STP in forward direction, and egress STP in
	            reverse direction.
	        -->
	    <xsd:element name="Orig" minOccurs="1" maxOccurs="1">
	      <xsd:annotation>
	        <xsd:documentation>The Orig atribute specifies an NSI endpoint name that
	identifies the ingress point of the connection. The NSI endpoint name provided is a
	reference into th etopologyDB where the physical characteristics of this endpoint can
	be found and validate against the service capabilities, e.g. the Ethernet service should
	only recognize Ethernet capable endpoints.</xsd:documentation>
	      </xsd:annotation>
	      <xsd:simpleType>
	        <xsd:restriction base="xsd:string">
	          <xsd:pattern value="[a-zA-Z0-9$]+:[a-zA-Z0-9$]+"/>
	        </xsd:restriction>
	      </xsd:simpleType>
	    </xsd:element>
	    <!--**
	            Dest := Egress STP in forward direction, and ingress STP in
	            reverse direction.
	        -->
	    <xsd:element name="Dest" minOccurs="1" maxOccurs="unbounded">
	      <xsd:annotation>
	        <xsd:documentation>The Dest atribute specifies an NSI endpoint name that
	identifies the egress point of the connection. The NSI endpoint name provided is a
	reference into the topologyDB where the physical characteristics of this endpoint can be
	found and validated against the service capabilities, e.g. the Ethernet service should
	only recognize Ethernet capable endpoints.</xsd:documentation>
	      </xsd:annotation>
	      <xsd:simpleType>
	        <xsd:restriction base="xsd:string">
	          <xsd:pattern value="[a-zA-Z0-9$]+:[a-zA-Z0-9$]+"/>
	        </xsd:restriction>
	      </xsd:simpleType>
	    </xsd:element>
	    <!--**
	
	        -->
	    <xsd:element name="Bandwidth">
	      <xsd:annotation>
	        <xsd:documentation>Bandwidth := The Megabits per second (Mbps) requested for a
	guarantied capacity connection. A zero value indicates a "best effort" connection with
	no guarantied capacity is being requested. Note - this does not commit that any user
	data will be delivered! Only that a best effort will be made for zero-cap connections.
	Note similarly, that a non-zero bw value does not imply that offered load in excess
	of the guaranty will be delivered at all. The nuanced difference in a zero vs non-zero
	bw request is that the service does commit to provide a best-effort to deliver frames

```
associated with a connection that has no bw guarranty, where as no such commitment is
made for connections that have requested a guaranteed capacity. This means a guaranteed
bw connection may have its excess load strictly policed at ingress and dropped, or
quietly buffered and delivered if possible...This service deinfition makes no commitment
and leaves it to individual netwoprks to resolve internally.</xsd:documentation>
     </xsd:annotation>
     <xsd:simpleType>
       <xsd:restriction base="xsd:integer">
         <xsd:minInclusive value="1"/>
         <xsd:maxInclusive value="1000"/>
       </xsd:restriction>
     </xsd:simpleType>
   </xsd:element>
   <!--***************************************************************

           -->
   <xsd:element name="MaximumBurstSize" default="100">
     <xsd:annotation>
       <xsd:documentation>MaximumBurstSize := the total size of contiguous frames that
can be transmitted on this circuit before ingress policing will begin dropping frames.
This parameter allows the source to transmit a stream of pdus up to this size and expect
the network to buffer them and transmit them over time. Implicitly associated with this
parameter is the buffer drain rate which is assumed to be a "fine grained constant rate"
output flow at the requested bandwidth. If the input queue excedes the MBS, the network
is allowed to, and should be expected to, police the ingress flow and drop offending
frames. This parameer is only meaningful for asynchronous packet based technologies such
as Ethernet. the Maximum Burst Size is measured in units of MegaBytes (2^20 Bytes)</
xsd:documentation>
     </xsd:annotation>
     <xsd:simpleType>
       <xsd:restriction base="xsd:integer">
         <xsd:minInclusive value="0"/>
         <xsd:maxInclusive value="1000"/>
       </xsd:restriction>
     </xsd:simpleType>
   </xsd:element>
   <!--***************************************************************

               -->
   <xsd:element name="ScheduleStart" type="xsd:dateTime">
     <xsd:annotation>
       <xsd:documentation>Schedule attributes are required. They are specified using the
XML Schema format for data and time. Example: 2012-01-01T06:00:00</xsd:documentation>
     </xsd:annotation>
   </xsd:element>
   <xsd:element name="ScheduleEnd" type="xsd:dateTime">
     <xsd:annotation>
       <xsd:documentation>Schedule attributes are required. They are specified using the
XML Schema format for data and time. Example: 2012-01-01T06:00:00</xsd:documentation>
     </xsd:annotation>
   </xsd:element>
   <!--***************************************************************
           -->
   <xsd:element name="AuthenticationInfo">
     <xsd:annotation>
       <xsd:documentation>AuthenticationInfo defines mechanisms the service recognizes
for request authentication. The authentication method specified in the Service Request
must match one of the methods offered by the Service Definition. Further, the string
presented as part of the method choice is present to an internal authentication model for
evaluation.</xsd:documentation>
     </xsd:annotation>
     <xsd:complexType>
       <xsd:choice>
         <!-- NSA has a flat file with allowed "simple users" listed...good for testing
-->
         <xsd:element name="simpleUser">
           <xsd:annotation>
             <xsd:documentation>The simpleUser authentication is a string representing a
user. The Provider Agent will look this up in an internally populated simpleUsers table.
A match will validate the user. This authenitcation is inherently insecure and should
only be used for testing. THis authentication method can be turned off by not providing
simpleUsers to polulate the table, or by removing the simpleUser element from the service
definition</xsd:documentation>
           </xsd:annotation>
           <xsd:simpleType>
             <xsd:restriction base="xsd:string">
               <xsd:minLength value="8"/>
               <xsd:maxLength value="32"/>
               <xsd:whiteSpace value="collapse"/>
             </xsd:restriction>
           </xsd:simpleType>
         </xsd:element>
         <!-- example of maybe another auth method... -->
```

```
          <xsd:element name="eduRoam" type="xsd:string">
            <!--  -->
          </xsd:element>
          <!-- This is intended to mimic current OSCARS style auth -->
          <xsd:element name="x.509cert" type="xsd:string">
            <!-- -->
          </xsd:element>
        </xsd:choice>
      </xsd:complexType>
    </xsd:element>
    <!-- The following attributes define statically defined values that are explicitly
 fixed as part
of the service -->
    <xsd:element name="FixedAttrShapingWindow" type="xsd:integer" default="100">
      <xsd:annotation>
        <xsd:documentation>The Standard Shaping Window is a timing window mesured in
 milliseconds (10^-3 seconds) that defines the period over which all connections are
 configured in order to "smooth" out flows... This is a parameter the source of a
 connection can use to manage their offered traffic. For instance: A 200 Mbps flow on a
 1 GE link consumes 20% of the link capacity. Therefore, that connection can expect to
 burst traffic for 20% * ShapingWindow. If the shaping window is 100 milliseconds, then
 the connection can expect that .2*100=20 millisecond bursts would be acceptable shaping
 for the flow. And every 100 miliseconds, the source could burst another 20 mils of data
 for that circuit.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```

Simple Type(s)

Simple Type serviceNameType

Namespace	No namespace
Diagram	
Type	restriction of xsd:string
Facets	whiteSpace collapse minLength 4 maxLength 32
Used by	Element ServiceDefinition/ServiceName

Source	`<xsd:simpleType name="serviceNameType">`
	`  <xsd:restriction base="xsd:string">`
	`    <xsd:whiteSpace value="collapse"/>`
	`    <xsd:minLength value="4"/>`
	`    <xsd:maxLength value="32"/>`
	`  </xsd:restriction>`
	`</xsd:simpleType>`