



QosManager:1 Erratum

Erratum Number:	<i>Next sequential erratum number</i>	Effective Date:	<i>July 14, 2006</i>
Document and Version:	<i>Document erratum applies to QosManager:1</i>		
Cross References:	<i>List other Erratum's or Documents that this change may apply to or have associated changes with</i>		

This Erratum has been adopted by the UPnP™ Technical Committee and includes the following information:

- A unique Erratum number.
- The date it becomes effective.
- The document version to which this Erratum applies.
- A series of Errata entries (numbered) which list the effected section and page number in the document referenced here, the exiting text in the document with the text to be changed highlighted green and the clarified text with the text modified or added highlighted light blue.

THE UPNP FORUM TAKES NO POSITION AS TO WHETHER ANY INTELLECTUAL PROPERTY RIGHTS EXIST IN THE PROPOSED TEMPLATES, IMPLEMENTATIONS OR IN ANY ASSOCIATED TEST SUITES. THIS ERRATUM IS PROVIDED "AS IS" AND "WITH ALL FAULTS". THE UPNP FORUM MAKES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE PROPOSED SERVICE TEMPLATES INCLUDING BUT NOT LIMITED TO ALL IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, OF REASONABLE CARE OR WORKMANLIKE EFFORT, OR RESULTS OR OF LACK OF NEGLIGENCE.

© 1999-2006 Contributing Members of the UPnP Forum. All Rights Reserved.



QosManager:1 Erratum

Entry

1



QosManager:1 Erratum

- Streaming Control
- Voice
- AV
- Data
- Audio
- Images
- Gaming
- Other
- Background

TspecIndex must be filled in by the Control Point. Other Tspec parameters may be omitted if they are not known.

TrafficHandle

TrafficHandle is a unique identifier associated with a particular instance of TrafficDescriptor; i.e. a particular traffic stream. It is a unique string generated by the QosManager and provided to the Control Point in response to the RequestTrafficQos action. The QosManager must insure that the generated Traffic Handle string is unique for all Traffic Descriptors on the network. In all subsequent communication between the CP and the QosManager service, TrafficHandle is used to reference a particular traffic stream.

AvailableOrderedTspecList

Contains one or more Tspec components. The Tspec index reflects the order of preference. Tspecs with smaller index values are more preferred. AvailableOrderedTspecList is required. A Tspec may be defined via any combination of its components.

ActiveTspecIndex

Active TspecIndex contains an integer which indicates the index of the current active Tspec from the TspecList. Tspec and Tspec List are defined above.

QosBoundarySourceAddress

If a traffic stream originates outside the home network (e.g. on the Internet), the CP application must provide the QoS boundary point IP address. This address will be treated as the QoS termination point for UPnP QoS. Optional, because it is applicable only to traffic streams originating outside the home network. It should be noted that this address is not part of traffic identifier, because the IP packets will carry the IP address of the actual source address. QosBoundarySourceAddress is used by QosManager for decisions related to path determination and device selection.

QosBoundaryDestinationAddress

If a traffic stream terminates outside the home network (e.g. on the Internet), the CP application must provide the QoS boundary point IP address. This address will be treated as the QoS termination point for UPnP QoS. Optional, because it is applicable only to traffic streams terminating outside the home network. It should be noted that this address is not part of traffic identifier, because the IP packets will carry the IP address of the actual destination address. QosBoundaryDestinationAddress is used by QosManager for decisions related to path determination and device selection.

TrafficId

TrafficId contains information about identifying / classifying packets that belong to that particular traffic stream. It consists of an XML structure consisting of: SourceIP; DestinationIP; SourcePort; DestinationPort; Protocol.



QoSManager:1 Erratum

Any of these arguments that are not known are represented as null.

UserName

UserName field contains a string associated with the user who requested a particular traffic stream. This field, if present, shall identify the name of the user initiating the UPnP QoS Action. The field must be ≤ 64 UTF-8 characters. At present this specification does not define the semantics for this field. This field represents the name of the user initiating the QoS action. Examples include, “Dad”, “Jimmy”, “CN#1234567”. The latter is an example of a name provided by a service provider to uniquely identify its individual customers.

CpName

CpName field contains a string associated with the Control Point requesting the traffic stream. This field, if present, shall identify the name of the control point initiating the UPnP QoS Action. The field must be ≤ 64 UTF-8 characters. At present this specification does not define the semantics for this field. The CpName may specify the brand name of the control point or it may indicate the location of the control point such as “living room”.

VendorApplicationName

VendorApplicationName field contains a single URI string associated with an application. This field, if present, shall identify the name of the application initiating the UPnP QoS Action. For applications specified by vendors, the value of this field must begin with “urn:”, followed by an ICANN domain name owned by the vendor, followed by “:application:”, followed by an application name. i.e. the value of this field should be constructed as follows:

urn:domain-name:application:appl-name

domain-name: Application vendor domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be ≤ 64 UTF-8 characters.

appl-name: Name of the application provided by a vendor. This must follow the syntax specified for Name Specific String (NSS) in the RFC 2141. The *appl-name* field must be ≤ 64 UTF-8 characters. The *application-name* must not contain the colon character.

When requesting QoS it may be useful to know the manufacturer and name of the application software client or server that will process the traffic stream.

PortName:

PortName field contains a single URI string associated with the fixed port used by an application for its connection set-up. This field, if present, shall identify the port number used by the application. The value of this field must begin with “urn:”, followed by an ICANN domain name owned by the vendor, followed by “:port:”, followed by the port name. i.e. the value of this field should be constructed as follows

urn:domain-name:port:port-number

domain-name: Application vendor domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be ≤ 64 UTF-8 characters.

port-number: A fixed port number used by the application either for connection set-up. This must be an integer value in the range 1 to 65535. If the application uses a range of port numbers then the starting port number of that range shall be specified. This value is either a vendor specific port used for the application or an IANA assigned port number for the application. The list of all the IANA assigned port numbers is maintained by IANA and is available at <http://www.iana.org/assignments/port-numbers>.



QosManager:1 Erratum

These fields indicate the port that is used by an application for its connection set-up. The use of IANA assigned port numbers is encouraged, but if not available for a particular application, a vendor assigned port number may be used.

ServiceProviderServiceName

ServiceProviderServiceName field contains a single URI string associated with a service offered by a service provider. This field, if present, shall identify the name of the service offered by a service provider.. For services provided by service providers, the value of this field must begin with “urn:”, followed by an ICANN domain name owned by a service provider , followed by “service:”, followed by a service name. i.e. the value of this field should be constructed as follows:

urn:domain-name:service:serv-name

domain-name: Service Provider domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be <= 64 UTF-8 characters.

serv-name: Name of the service provided by a service vendor. This must follow the syntax specified for Name Specific String (NSS) in the RFC 2141. The *serv-name* field must be <= 64 UTF-8 characters. The *serv-name* must not contain the colon character.

This represents the name of a particular service being offered by a service provider. Note that while multiple services provided by the same service provider may use the same software application, different Service offerings may be given different priorities (traffic importance).

TrafficLeaseTime

This field contains the lease-time associated with a particular traffic stream. The lease time is expressed in ‘Seconds’. When not specified, it indicates an indefinite lease.

MediaServerConnectionId:

This is an optional field. This field may be useful in case of UPnP AV based streaming with multiple Tspecs. MediaServerConnectionId may be used by the QosManager to identify the traffic stream that is being setup by the UPnP AV Control Point which is requesting Qos. Please refer to UPnP QoS Architecture document for more details. The connection Id is obtained from the Media Server via CM::PrepareForConnection action.

MediaRendererConnectionId:

This is an optional field. This field may be useful in case of UPnP AV based streaming with multiple Tspecs. MediaRendererConnectionId may be used by the QosManager to identify the traffic stream that is being setup by the UPnP AV Control Point which is requesting Qos. Please refer to UPnP QoS Architecture document for more details. The connection Id is obtained from the Media Renderer via CM::PrepareForConnection action.

New Text

2.2.2 A_ARG_TYPE_TrafficDescriptor

This state variable contains information about a particular traffic stream. A Traffic Descriptor consists of an XML structure consisting of:

- TrafficHandle
- TrafficID
- AvailableOrderedTspecList
- QosBoundarySourceAddress



QosManager:1 Erratum

- TrafficImportanceNumber
- QosBoundaryDestinationAddress
- UserName
- CpName
- VendorApplicationName
- PortName
- ServiceProviderServiceName
- TrafficLeaseTime

Please refer to the detailed TrafficDescriptor XML schema in Appendix A. Here are some details about information in Traffic Descriptor:

Tspec

Tspec contains a description of Content QoS Requirements. Tspec is represented in the form of a XML structure. In UPnP AV scenario, this information is extracted from the Content Directory Service of the Media Server. In the Content Directory Service, Tspec is represented either as a string containing escaped XML structure, or an URI pointing to the escaped XML structure. UPnP AV Control Point uses CDS:Browse and/or CDS:Search action calls to acquire the Tspec(s) associated with the content and creates an AvailableOrderedTspecList field in the Traffic Descriptor.

Tspec XML structure consists of the following:

- ❑ TspecIndex: Unique numerical index associated with a particular Tspec. The value of Tspecindex indicates preference (as defined by the application or the end user). A Tspec with smaller Index is more preferred compared to a Tspec with larger Index.
- ❑ AVTransportURI: This string field contains a unique URI associated with the UPnP AV content item for which QoS is being requested.
- ❑ AVTransportInstanceID: This integer field contains a unique Instance ID associated with the UPnP AV Transport service associated with the content item for which QoS is being requested.
- ❑ TrafficClass: The traffic class associated with the traffic stream. This is an enumerated variable that can be assigned to one of the following list of values:
 - Network Control
 - Streaming Control
 - Voice
 - AV
 - Data
 - Audio
 - Image
 - Gaming
 - Other
 - Background

TspecIndex must be filled in by the Control Point. Other Tspec parameters may be omitted if they are not known. TspecIndexes need not be consecutive numbers. The Tspec with lower TspecIndex is more preferred than a higher TspecIndex. In this example, the Tspec with 2 is more preferred than that with 300. The order of Tspec structures in the AvailableOrderedTspecList is not relevant. All Tspecs must have unique TspecIndex value within an AvailableOrderedTspecList.

```
<AvailableOrderedTspecList>
  <Tspec>
    <TspecIndex>300</TspecIndex>
    <TrafficClass>AV</TrafficClass>
```



QosManager:1 Erratum

```
</Tspec>  
<Tspec>  
  <TspecIndex>2</TspecIndex>  
  <TrafficClass>Audio</TrafficClass>  
</Tspec>  
</AvailableOrderedTspecList>
```

TrafficHandle

TrafficHandle is a unique identifier associated with a particular instance of TrafficDescriptor; i.e. a particular traffic stream. It is a unique string generated by the QosManager and provided to the Control Point in response to the RequestTrafficQos action. The QosManager must insure that the generated Traffic Handle string is unique for all Traffic Descriptors on the network. In all subsequent communication between the CP and the QosManager service, TrafficHandle is used to reference a particular traffic stream.

AvailableOrderedTspecList

Contains one or more Tspec components. The Tspec index reflects the order of preference. Tspecs with smaller index values are more preferred. AvailableOrderedTspecList is required. A Tspec may be defined via any combination of its components.

ActiveTspecIndex

Active TspecIndex contains an integer which indicates the index of the current active Tspec from the TspecList. Tspec and Tspec List are defined above.

QosBoundarySourceAddress

If a traffic stream originates outside the home network (e.g. on the Internet), the CP application must provide the QoS boundary point IP address. This address will be treated as the QoS termination point for UPnP QoS. Optional, because it is applicable only to traffic streams originating outside the home network. It should be noted that this address is not part of traffic identifier, because the IP packets will carry the IP address of the actual source address. QosBoundarySourceAddress is used by QosManager for decisions related to path determination and device selection.

QosBoundaryDestinationAddress

If a traffic stream terminates outside the home network (e.g. on the Internet), the CP application must provide the QoS boundary point IP address. This address will be treated as the QoS termination point for UPnP QoS. Optional, because it is applicable only to traffic streams terminating outside the home network. It should be noted that this address is not part of traffic identifier, because the IP packets will carry the IP address of the actual destination address. QosBoundaryDestinationAddress is used by QosManager for decisions related to path determination and device selection.

TrafficId

TrafficId contains information about identifying / classifying packets that belong to that particular traffic stream. It consists of an XML structure consisting of: SourceIP; DestinationIP; SourcePort; DestinationPort; Protocol.

Any of these arguments that are not known are represented as null.

UserName



QoSManager:1 Erratum

UserName field contains a string associated with the user who requested a particular traffic stream. This field, if present, shall identify the name of the user initiating the UPnP QoS Action. The field must be ≤ 64 UTF-8 characters. At present this specification does not define the semantics for this field. This field represents the name of the user initiating the QoS action. Examples include, “Dad”, “Jimmy”, “CN#1234567”. The latter is an example of a name provided by a service provider to uniquely identify its individual customers.

CpName

CpName field contains a string associated with the Control Point requesting the traffic stream. This field, if present, shall identify the name of the control point initiating the UPnP QoS Action. The field must be ≤ 64 UTF-8 characters. At present this specification does not define the semantics for this field. The CpName may specify the brand name of the control point or it may indicate the location of the control point such as “living room”

VendorApplicationName

VendorApplicationName field contains a single URI string associated with an application. This field, if present, shall identify the name of the application initiating the UPnP QoS Action. For applications specified by vendors, the value of this field must begin with “urn:”, followed by an ICANN domain name owned by the vendor, followed by “:application:”, followed by an application name. i.e. the value of this field should be constructed as follows:

urn:domain-name:application:appl-name

domain-name: Application vendor domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be ≤ 64 UTF-8 characters.

appl-name: Name of the application provided by a vendor. This must follow the syntax specified for Name Specific String (NSS) in the RFC 2141. The *appl-name* field must be ≤ 64 UTF-8 characters. The *application-name* must not contain the colon character.

When requesting QoS it may be useful to know the manufacturer and name of the application software client or server that will process the traffic stream.

PortName:

PortName field contains a single URI string associated with the fixed port used by an application for its connection set-up. This field, if present, shall identify the port number used by the application. The value of this field must begin with “urn:”, followed by an ICANN domain name owned by the vendor, followed by “:port:”, followed by the port name. i.e. the value of this field should be constructed as follows

urn:domain-name:port:port-number

domain-name: Application vendor domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be ≤ 64 UTF-8 characters.

port-number: A fixed port number used by the application either for connection set-up. This must be an integer value in the range 1 to 65535. If the application uses a range of port numbers then the starting port number of that range shall be specified. This value is either a vendor specific port used for the application or an IANA assigned port number for the application. The list of all the IANA assigned port numbers is maintained by IANA and is available at <http://www.iana.org/assignments/port-numbers>.

These fields indicate the port that is used by an application for its connection set-up. The use of IANA assigned port numbers is encouraged, but if not available for a particular application, a vendor assigned port number may be used.



QosManager:1 Erratum

ServiceProviderServiceName

ServiceProviderServiceName field contains a single URI string associated with a service offered by a service provider. This field, if present, shall identify the name of the service offered by a service provider.. For services provided by service providers, the value of this field must begin with “urn:”, followed by an ICANN domain name owned by a service provider , followed by “service:”, followed by a service name. i.e. the value of this field should be constructed as follows:

urn:domain-name:service:serv-name

domain-name: Service Provider domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be <= 64 UTF-8 characters.

serv-name: Name of the service provided by a service vendor. This must follow the syntax specified for Name Specific String (NSS) in the RFC 2141. The *serv-name* field must be <= 64 UTF-8 characters. The *serv-name* must not contain the colon character.

This represents the name of a particular service being offered by a service provider. Note that while multiple services provided by the same service provider may use the same software application, different Service offerings may be given different priorities (traffic importance).

TrafficLeaseTime

This field contains the lease-time associated with a particular traffic stream. The lease time is expressed in ‘Seconds’. When not specified, it indicates an indefinite lease.

MediaServerConnectionId:

This is an optional field. This field may be useful in case of UPnP AV based streaming with multiple Tspecs. MediaServerConnectionId may be used by the QosManager to identify the traffic stream that is being setup by the UPnP AV Control Point which is requesting Qos. Please refer to UPnP QoS Architecture document for more details. The connection Id is obtained from the Media Server via CM::PrepareForConnection action.

MediaRendererConnectionId:

This is an optional field. This field may be useful in case of UPnP AV based streaming with multiple Tspecs. MediaRendererConnectionId may be used by the QosManager to identify the traffic stream that is being setup by the UPnP AV Control Point which is requesting Qos. Please refer to UPnP QoS Architecture document for more details. The connection Id is obtained from the Media Renderer via CM::PrepareForConnection action.

Use the following XML documents as examples for TrafficDescriptor schema.

Example 1:

```
<TrafficDescriptor
  xmlns="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd
TrafficDescriptorv1.xsd" >
  <TrafficHandle>kiwin1</TrafficHandle>
  <TrafficId>
    <SourceAddress>
      <Ipv4>192.168.1.50</Ipv4>
    </SourceAddress>
    <SourcePort>23</SourcePort>
    <DestinationAddress>
      <Ipv4>192.168.1.53</Ipv4>
    </DestinationAddress>
    <DestinationPort>23</DestinationPort>
```



QosManager:1 Erratum

```
<IpProtocol>1</IpProtocol>
</TrafficId>
<AvailableOrderedTspecList>
  <Tspec>
    <TspecIndex>300</TspecIndex>
    <TrafficClass>AV</TrafficClass>
  </Tspec>
  <Tspec>
    <TspecIndex>2</TspecIndex>
    <TrafficClass>Audio</TrafficClass>
  </Tspec>
</AvailableOrderedTspecList>
<ActiveTspecIndex>300</ActiveTspecIndex>
<TrafficImportanceNumber>5</TrafficImportanceNumber>
<OptionalPolicyParams>
  <CpName>Amy's CP</CpName>
</OptionalPolicyParams>
</TrafficDescriptor>
```

Example 2:

```
<TrafficDescriptor
xmlns="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:prv="http://myPrivate.com"
xmlns:prv2="http://myPrivate2.com"
xsi:schemaLocation="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd
TrafficDescriptorv1.xsd" >
```

```
<TrafficHandle>kiwin1</TrafficHandle>
<TrafficId>
  <SourceAddress>
    <Ipv4>192.168.1.50</Ipv4>
  </SourceAddress>
  <SourcePort>23</SourcePort>
  <DestinationAddress>
    <Ipv4>192.168.1.53</Ipv4>
  </DestinationAddress>
  <DestinationPort>23</DestinationPort>
  <IpProtocol>1</IpProtocol>
</TrafficId>
<AvailableOrderedTspecList>
  <Tspec>
    <TspecIndex>300</TspecIndex>
    <TrafficClass>AV</TrafficClass>
  </Tspec>
  <Tspec>
    <TspecIndex>2</TspecIndex>
    <TrafficClass>Audio</TrafficClass>
  </Tspec>
</AvailableOrderedTspecList>
<ActiveTspecIndex>300</ActiveTspecIndex>
<TrafficImportanceNumber>5</TrafficImportanceNumber>
<OptionalPolicyParams>
  <CpName>Amy's CP</CpName>
</OptionalPolicyParams>
</TrafficDescriptor>
```

Entry	2
Document Section	2.2.6 A_ARG_TYPE_ListOfTrafficDescriptors
Document Page	11

Background

Schema validation is clarified.

Current Text

2.2.6 A_ARG_TYPE_ListOfTrafficDescriptors

This state variable contains one or more instances of the state variable A_ARG_TYPE_TrafficDescriptor.

Here is the XML schema for this state variable:

```
<?xml version="1.0" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.upnp.org/schemas/ListOfTrafficDescriptors.xsd"
  targetNamespace="http://www.upnp.org/schemas/ListOfTrafficDescriptors.xsd"
  elementFormDefault="qualified">

  <xs:annotation>
    <xs:documentation xml:lang="en">
      List of Traffic Descriptors schema.
      Copyright 2004 UPnP(tm). All rights reserved.
    </xs:documentation>
  </xs:annotation>

  <xs:element name="ListOfTrafficDescriptors">
    <xs:complexType>
      <xs:element name="TrafficDescriptor" type="TrafficDescriptor"
        minOccurs="0" />
    </xs:complexType>
  </xs:element>
</xs:schema>
```

New Text

2.2.6 A_ARG_TYPE_ListOfTrafficDescriptors

This state variable contains one or more instances of the state variable A_ARG_TYPE_TrafficDescriptor.

Here is the XML schema for this state variable:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://www.upnp.org/schemas/ListOfTrafficDescriptors.xsd"
  xmlns="http://www.upnp.org/schemas/ListOfTrafficDescriptors.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:td="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd"
  elementFormDefault="qualified" id="ListOfTrafficDescriptors">

  <xs:import namespace="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd"
    schemaLocation="TrafficDescriptorv1.xsd"/>
  <xs:annotation>
    <xs:documentation xml:lang="en">
      List of Traffic Descriptors schema.
      Copyright 2004, 2005 UPnP(tm). All rights reserved.
    </xs:documentation>
```



QosManager:1 Erratum

```
</xs:annotation>
<xs:element name="ListOfTrafficDescriptors" type="ListOfTrafficDescriptorsType"/>
<xs:complexType name="ListOfTrafficDescriptorsType">
  <xs:sequence>
    <xs:element name="TrafficDescriptor" type="td:TrafficDescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

Use the following XML document as an example for ListOfTrafficDescriptors schema.

```
<ListOfTrafficDescriptors
  xmlns="http://www.upnp.org/schemas/ListOfTrafficDescriptors.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/ListOfTrafficDescriptors.xsd
ListOfTrafficDescriptors.xsd">
  <TrafficDescriptor>
    <TrafficHandle>kiwin1</TrafficHandle>
    <TrafficId>
      <SourceAddress>
        <Ipv4>192.168.1.50</Ipv4>
      </SourceAddress>
      <SourcePort>23</SourcePort>
      <DestinationAddress>
        <Ipv4>192.168.1.50</Ipv4>
      </DestinationAddress>
      <DestinationPort>23</DestinationPort>
      <IpProtocol>1</IpProtocol>
    </TrafficId>
    <AvailableOrderedTspecList>
      <Tspec>
        <TspecIndex>300</TspecIndex>
        <TrafficClass>AV</TrafficClass>
      </Tspec>
      <Tspec>
        <TspecIndex>2</TspecIndex>
        <TrafficClass>Audio</TrafficClass>
      </Tspec>
    </AvailableOrderedTspecList>
    <ActiveTspecIndex>1</ActiveTspecIndex>
    <TrafficImportanceNumber>5</TrafficImportanceNumber>
    <OptionalPolicyParams>
      <CpName>Amy's CP</CpName>
    </OptionalPolicyParams>
  </TrafficDescriptor>
  <TrafficDescriptor>
    <TrafficHandle>kiwin2</TrafficHandle>
    <TrafficId>
      <SourceAddress>
        <Ipv4>192.168.1.53</Ipv4>
      </SourceAddress>
      <SourcePort>23</SourcePort>
      <DestinationAddress>
        <Ipv4>192.168.1.55</Ipv4>
      </DestinationAddress>
      <DestinationPort>23</DestinationPort>
      <IpProtocol>1</IpProtocol>
    </TrafficId>
    <AvailableOrderedTspecList>
      <Tspec>
        <TspecIndex>300</TspecIndex>
        <TrafficClass>AV</TrafficClass>
      </Tspec>
      <Tspec>
        <TspecIndex>2</TspecIndex>
        <TrafficClass>Audio</TrafficClass>
      </Tspec>
    </AvailableOrderedTspecList>
  </TrafficDescriptor>
```



QosManager:1 Erratum

```
</AvailableOrderedTspecList>  
<ActiveTspecIndex>1</ActiveTspecIndex>  
<TrafficImportanceNumber>5</TrafficImportanceNumber>  
<OptionalPolicyParams>  
  <CpName>Amy's CP</CpName>  
</OptionalPolicyParams>  
</TrafficDescriptor>  
</ListOfTrafficDescriptors>
```



QosManager:1 Erratum

Entry	3
Document	
Section	2.4.1 RequestTrafficQos
Document	
Page	13

Background

The various error conditions due to bad input are explained.

Current Text

2.4.1 RequestTrafficQos

Control Point can invoke this action for setting up the Quality of Service for a particular traffic stream.

New Text

2.4.1 RequestTrafficQos

Control Point can invoke this action for setting up the Quality of Service for a particular traffic stream. At least one of the elements of SourceAddress or DestinationAddress or SourcePort or DestinationPort or IpProtocol must be specified in the TrafficId structure.

If Control Point supplies TrafficImportanceNumber in TrafficDescriptor to QosManager when calling this action QosManager must return error 721.

If Control Point supplies TrafficDescriptor to QosManager when calling this action, the QosManager must return error code 722.

QosManager must include a valid ActiveTspecIndex value in the TrafficDescriptor when it returns a non error value in response to this action.

If Control Point supplies a non-null TrafficHandle to QosManager when calling this action, the QosManager must return error code 701.

When there are multiple Tspec(s) in the TrafficDescriptor provided by the Control Point to RequestTrafficQoS at the QosManager, the QosManager must set up the most preferred Tspec first. If the setup fails, then the QosManager must try next preferred Tspec and so on.



QosManager:1 Erratum

Entry	4
Document Section	2.4.2 UpdateTrafficQos
Document Page	14

Background

The various error conditions due to bad input are explained.

Current Text

2.4.2 UpdateTrafficQos

Control Point can invoke this action for updating the Quality of Service for a particular traffic stream.

New Text

2.4.2 UpdateTrafficQos

Control Point can invoke this action for updating the Quality of Service for a particular traffic stream.

If Control Point supplies TrafficImportanceNumber in TrafficDescriptor to QosManager when calling this action QosManager must return error 721.

If Control Point supplies TrafficDescriptor to QosManager when calling this action, the QosManager must return error code 722.

QosManager must include a valid ActiveTspecIndex value in the TrafficDescriptor when it returns a non error value in response to this action.

If the Control Point does not supply a TrafficHandle when calling this action the QosManager must return error code 700.

The Control Point may update TrafficLeaseTime, UserName and/or CpName during UpdateTrafficQos action on the QosManager. All other non-any parameters must not be updated by the Control Point, and such an input must result in error code 714.

In response to the Control Point calling the UpdateTrafficQos action, the QosManager may update ActiveTspecIndex, and/or TrafficImportanceNumber in the TrafficDescriptor structure. All other non-any parameters must not be updated by the QosManager and result in Error Code 714.

Entry	5
Document Section	2.5 Theory of Operation
Document Page	13

Background

Correct spelling mistakes.

Current Text

2.5 Theory of Operation

This service-type enables modeling of ‘Quality of Service Manager’ capabilities. A UPnP QosManager is responsible for setting up, updating, revoking and in general controlling the Quality of Service assigned by networking devices to various traffic streams. A QosManager acts as a control point to UPnP QosDevice services. A QosManager discovers the QosDevice services on the local area network, and may use ‘QD:GetQosDeviceCapabilities’¹ and ‘QD:GetQostate’ actions exposed by the QosDevice service to get information about the device. ‘QD:GetQosDeviceCapabilities’ action returns capabilities of QoS enabled devices (and the details on the various network interfaces present on the device). ‘QD:GetQosState’ action returns the current state of the UPnP QoS enabled device with information such as the TrafficDescriptors that are currently active.

The UPnP QosManager exposes the ‘QM:BrowseAllTrafficDescriptors’² action to the control point. Upon receiving this action request, the QosManager can report back the details on various traffic streams admitted on the UPnP QoS devices on the network. QosManager also discovers the QosPolicyHolder services on the local area network. It is expected that there will be one such PolicyHolder service instance on the network in normal condition. If the QosManager fails to discover any PolicyHolder service, or if QosManager discovers more than one PolicyHolder service, the QosManager assumes the default policy of first-come-first serve admission control with default priorities based on 802.1D Annex G. The default value for the AdmissionPolicy is “Enabled”.

The following table describes how the QosManager derives default priorities to be used in the Traffic Importance Number field. The QosManager takes the value of Traffic Class from the Tspec found inside the Traffic Descriptor structure, and looks up the corresponding default priority from the following table:

Traffic Class	802.1D (Annex G) Traffic Type	Default Priority (Traffic Importance Number)
Network Control	NC	7
Streaming Control	NC	7
Voice	VO	6
Gaming	VO	6
AV	VI	5
Audio	VI	5
Image	EE	3
Data	BE	0



QosManager:1 Erratum

In the context of UPnP AV, the AV Control Point invokes QosManager (from the UPnP AV Control Point) to perform the functions related to setting up Quality of Service a particular traffic. In case of non UPnP AV scenario, any application (acting as a Control Point) can invoke the QosManager service for setting up the Quality of Service for a particular traffic. In either case, the Control Point uses the 'QM:RequestTrafficQos' action exposed by the QosManager service to invoke the QosManager service. The control point puts the initial TrafficDescriptor as an input argument in the RequestTrafficQos action. For example, activeTspegIndex and parts of TrafficID (e.g. port numbers) may be unknown to the Control Point at the time of setup. These details may get filled in after the traffic QoS is implemented in the UPnP QoS Device. The control point may also include: UserName, CpName, VendorApplicationName, PortName, and ServiceProviderServiceName. These fields are conveyed by the QoS Management entity to the QosPolicyHolder using the GetTrafficPolicy action. The QosPolicyHolder may use any information in the TrafficDescriptor to determine the TrafficImportanceNumber and UserImportanceNumber for the traffic stream that QoS is being requested for. It is recommended that the control point provides as much of information as available in the TrafficDescriptor XML fragment.

Upon receiving the 'QM:RequestTrafficQos' action, the QosManager Qos Management Entity obtains the TrafficImportanceNumber and the UserImportanceNumber for that TrafficDescriptor by querying the QosPolicyHolder. Traffic Importance Number (0-7) will be provided to QoS Device by the QosManager Qos Management Entity. User Importance Number (0-255) will be used by QosManager for traffic admission. In addition to User Importance Number and Traffic Importance Number, the QosPolicyHolder also returns a flag that indicates whether admission policy is enabled or disabled.

By invoking 'QD:GetPathInformation' action on QosDevice service instances on the network, the QosManager may determine which intermediate devices are on the path from the source to the sink. Some UPnP QosDevice service instances may also expose the 'QD:GetQosDeviceInfo' action. QoS management entity may invoke this action on the source and/or sink device to find out the port number and protocol information associated with that particular Traffic Descriptor. The QosManager Qos Management Entity issues the 'QD:GetQosState' action to every QoS Device on the path of the traffic stream and gets the dynamic status about that device. If QosManagement entity decides that there are insufficient resource, it may try another Tspec listed in the TrafficDescriptor. If QosManagement entity decides that there are enough resources available on all the devices on the path of the traffic, the QosManager issues 'QD:SetupTrafficQos' actions to those devices on the path forwhich AdmissionControlSupported equals No. If SetupTrafficQos action is successfully executed on all the QoS Devices involved, then the updated TrafficDescriptor created by the QosManager is returned to the control point as an output argument to the 'QM:RequestTrafficQos' action. Otherwise, QosManager returns appropriate error in response to the 'QM:RequestTrafficQos' action.

QosStateId is a state variable defined in the QosDevice that enumerates the state of the device when it was queried by a QosManager with QD:GetQosState action. QosDevice returns QosStateId in response to QD:GetQosState as part of QosDeviceState argument. Subsequently, the QosManager repeats this string as input argument to QD:SetupTrafficQos. Please see QosDevice Service Definition for more details.

When a control point needs to either change or release the Quality of Service associated with a particular traffic, it uses the 'QM:UpdateTrafficQos' or 'QM:ReleaseTrafficQos' actions respectively. Upon receiving 'QM:UpdateTrafficQos' action, the QosManager repeats the admission control process described above for the revised TrafficDescriptor. Upon receiving 'QM:ReleaseTrafficQos' the QosManager invokes the 'ReleaseTrafficQos' action on all the devices on the path of that traffic.

A_ARG_TYPE_PathInformation state variable of QoS Device service is optional, but it is evented with moderation. Any time there is a change in Path information, the QoS device may issue an event and send the updated A_ARG_TYPE_PathInformation variable in the body of the event. QosManager may subscribe to these events. Upon receiving an event, QosManager may take an action that it deems necessary e.g. recomputing the topology.

Finally, it should be noted that UPnP QoS defines services (QosManager, QosDevice, and QosPolicyHolder) but it does not define new device type. Since Quality of Service problem needs to be solved across the board for multiple usage scenarios, it is expected that vendors may use any UPnP device as a container for the services defined in UPnP QoS. The control points and QoS Management Entities must look for UPnP QoS services embedded in all UPnP device types.



New Text

2.5 Theory of Operation

This service-type enables modeling of ‘Quality of Service Manager’ capabilities. A UPnP QosManager is responsible for setting up, updating, revoking and in general controlling the Quality of Service assigned by networking devices to various traffic streams. A QosManager acts as a control point to UPnP QosDevice services. A QosManager discovers the QosDevice services on the local area network, and may use ‘QD:GetQosDeviceCapabilities’³ and ‘QD:GetQosState’ actions exposed by the QosDevice service to get information about the device. ‘QD:GetQosDeviceCapabilities’ action returns capabilities of QoS enabled devices (and the details on the various network interfaces present on the device). ‘QD:GetQosState’ action returns the current state of the UPnP QoS enabled device with information such as the TrafficDescriptors that are currently active.

The UPnP QosManager exposes the ‘QM:BrowseAllTrafficDescriptors’⁴ action to the control point. Upon receiving this action request, the QosManager can report back the details on various traffic streams admitted on the UPnP QoS devices on the network. QosManager also discovers the QosPolicyHolder services on the local area network. It is expected that there will be one such PolicyHolder service instance on the network in normal condition. If the QosManager fails to discover any PolicyHolder service, or if QosManager discovers more than one PolicyHolder service, the QosManager assumes the default policy of first-come-first serve admission control with default priorities based on 802.1D Annex G. The default value for the AdmissionPolicy is “Enabled”.

The following table describes how the QosManager derives default priorities to be used in the Traffic Importance Number field. The QosManager takes the value of Traffic Class from the Tspec found inside the Traffic Descriptor structure, and looks up the corresponding default priority from the following table:

Traffic Class	802.1D (Annex G) Traffic Type	Default Priority (Traffic Importance Number)
Network Control	NC	7
Streaming Control	NC	7
Voice	VO	6
Gaming	VO	6
AV	VI	5
Audio	VI	5
Image	EE	3
Data	BE	0
Other	BE	0
Background	BK	1

In the context of UPnP AV, the AV Control Point invokes QosManager (from the UPnP AV Control Point) to perform the functions related to setting up Quality of Service a particular traffic. In case of non UPnP AV scenario, any application (acting as a Control Point) can invoke the QosManager service for setting up the Quality of Service for a particular traffic. In either case, the Control Point uses the ‘QM:RequestTrafficQos’ action exposed by the QosManager service to invoke the QosManager service. The control point puts the initial TrafficDescriptor as an input argument in the RequestTrafficQos action. For example, activeTspecIndex and parts of TrafficID (e.g. port numbers) may be unknown to the Control Point at the time of setup. These details may get filled in after the traffic QoS is implemented in the UPnP QoS Device. The control point may also include: UserName, CpName, VendorApplicationName, PortName, and ServiceProviderServiceName. These fields are conveyed by the QoS Management entity to the QosPolicyHolder using the GetTrafficPolicy action. The QosPolicyHolder may use any information in

³ QD: is QosDevice service.

⁴ QM: is QosManager service



***QosManager:1* Erratum**

the TrafficDescriptor to determine the TrafficImportanceNumber and UserImportanceNumber for the traffic stream that QoS is being requested for. It is recommended that the control point provides as much of information as available in the TrafficDescriptor XML fragment.

Upon receiving the 'QM:RequestTrafficQos' action, the QosManager Qos Management Entity obtains the TrafficImportanceNumber and the UserImportanceNumber for that TrafficDescriptor by querying the QosPolicyHolder. Traffic Importance Number (0-7) will be provided to QoS Device by the QosManager Qos Management Entity. User Importance Number (0-255) will be used by QosManager for traffic admission. In addition to User Importance Number and Traffic Importance Number, the QosPolicyHolder also returns a flag that indicates whether admission policy is enabled or disabled.

By invoking 'QD:GetPathInformation' action on QosDevice service instances on the network, the QosManager may determine which intermediate devices are on the path from the source to the sink. Some UPnP QosDevice service instances may also expose the 'QD:GetQosDeviceInfo' action. QoS management entity may invoke this action on the source and/or sink device to find out the port number and protocol information associated with that particular Traffic Descriptor. The QosManager Qos Management Entity issues the 'QD:GetQosState' action to every QoS Device on the path of the traffic stream and gets the dynamic status about that device. If QosManagement entity decides that there are insufficient resource, its may try another Tspec listed in the TrafficDescriptor. If QosManagement entity decides that there are enough resources available on all the devices on the path of the traffic, the QosManager issues 'QD:SetupTrafficQos' actions to those devices on the path forwhich AdmissionControlSupported equals No. If SetupTrafficQos action is successfully executed on all the QoS Devices involved, then the updated TrafficDescriptor created by the QosManager is returned to the control point as an output argument to the 'QM:RequestTrafficQos' action. Otherwise, QosManager returns appropriate error in response to the 'QM:RequestTrafficQos' action.

QosStateId is a state variable defined in the QosDevice that enumerates the state of the device when it was queried by a QosManager with QD:GetQosState action. QosDevice returns QosStateId in response to QD:GetQosState as part of QosDeviceState argument. Subsequently, the QosManager repeats this string as input argument to QD:SetupTrafficQos. Please see QosDevice Service Definition for more details.

When a control point needs to either change or release the Quality of Service associated with a particular traffic, it uses the 'QM:UpdateTrafficQos' or 'QM:ReleaseTrafficQos' actions respectively. Upon receiving 'QM:UpdateTrafficQos' action, the QosManager repeats the admission control process described above for the revised TrafficDescriptor. Upon receiving 'QM:ReleaseTrafficQos' the QosManager invokes the 'ReleaseTrafficQos' action on all the devices on the path

Entry	6
Document	
Section	5. Appendix A TrafficDescriptor XML Schema
Document	
Page	

Background

Input traffic descriptor restrictions described using schema validation and clarified text.

Current Text

5. Appendix A TrafficDescriptor XML Schema

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd"
  targetNamespace="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd"
  elementFormDefault="qualified">
  <xs:element name="TrafficDescriptor" minOccurs="1" maxOccurs="1">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="TrafficHandle" type="xs:string" minOccurs="1"
maxOccurs="1" />
        <xs:element name="TrafficId" minOccurs="1" maxOccurs="1">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="SourceAddress" type="IpAddress" minOccurs="0"
/>
              <xs:element name="SourcePort" type="IpPortNumber" minOccurs="0"
/>
              <xs:element name="DestinationAddress" type="IpAddress"
minOccurs="0" />
              <xs:element name="DestinationPort" type="IpPortNumber"
minOccurs="0" />
              <xs:element name="IpProtocol" type="IpProtocolType" minOccurs="0"
/>
              <xs:any minOccurs="0" maxOccurs="unbounded" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>

        <xs:element name="AvailableOrderedTspecList" minOccurs="1"
maxOccurs="1">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Tspec" type="TrafficSpecification"
minOccurs="1" maxOccurs="unbounded">
                </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>

        <xs:element name="ActiveTspecIndex" type="xs:integer" minOccurs="1"
maxOccurs="1" />
        <xs:element name="TrafficImportanceNumber" type="TrafficImportance"
default="0"
minOccurs="1" maxOccurs="1" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```



QosManager:1 Erratum

```
<xs:element name="QosBoundarySourceAddress" type="IpAddress"
minOccurs="0" />
<xs:element name="QosBoundaryDestinationAddress" type="IpAddress"
minOccurs="0" />

<xs:element name="MediaServerConnectionId" type="xs:integer"
minOccurs="0" maxOccurs="1" />
<xs:element name="MediaRendererConnectionId" type="xs:integer"
minOccurs="0" maxOccurs="1" />
<xs:element name="TrafficLeaseTime" type="xs:integer"
minOccurs="0"
maxOccurs="1" />
<xs:any minOccurs="0" maxOccurs="unbounded" />

<xs:element name="OptionalPolicyParams" minOccurs="0" maxOccurs="1">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="UserName" type="xs:string" minOccurs="0"
maxOccurs="1" />
      <xs:element name="CpName" type="xs:string" minOccurs="0"
maxOccurs="1" />
      <xs:element name="VendorApplicationName" type="xs:string"
minOccurs="0"
maxOccurs="1" />
      <xs:element name="PortName" type="xs:string" minOccurs="0"
maxOccurs="1" />
      <xs:element name="ServiceProviderServiceName" type="xs:string"
minOccurs="0"
maxOccurs="1" />
      <xs:any minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:complexType name="TrafficSpecification">
  <xs:sequence>
    <xs:element name="TspecIndex" type="xs:integer" minOccurs="1"
maxOccurs="1" />
    <xs:element name="AvTransportUri" type="xs:string" minOccurs="0"
maxOccurs="1" />
    <xs:element name="AvTransportInstanceId" type="xs:integer"
minOccurs="0"
maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:element name="TrafficClass" type="TrafficClass" minOccurs="0"
maxOccurs="1" />
<xs:any minOccurs="0" maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>

<xs:simpleType name="TrafficClass">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="Network Control" />
    <xs:enumeration value="Streaming Control" />
    <xs:enumeration value="Voice" />
    <xs:enumeration value="AV" />
    <xs:enumeration value="Data" />
    <xs:enumeration value="Audio" />
    <xs:enumeration value="Image" />
  </xs:restriction>
</xs:simpleType>
```



QosManager:1 Erratum

```
<xs:enumeration value="Gaming"/>
<xs:enumeration value="Other"/>
<xs:enumeration value="Background"/>
</xs:restriction>
</xs:simpleType>

<xs:simpleType name="TrafficImportance">
  <xs:restriction base="xs:nonNegativeInteger">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="7"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="Ipv6Address">
  <xs:restriction base="xs:hexBinary">
    <xs:length value="32"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="IpPortNumber">
  <xs:restriction base="xs:positiveInteger">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="65535"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="IpProtocolType">
  <xs:restriction base="xs:nonNegativeInteger">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="255"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="IpAddress">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Ipv4" type="IPv4Address"/>
      <xs:element name="Ipv6" type="IPv6Address"/>
    </xs:choice>
    <xs:element name="PrefixLength">
      <xs:simpleType>
        <xs:restriction base="xs:positiveInteger" minOccurs="0">
          <xs:minInclusive value="1"/>
          <xs:maxInclusive value="128"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="IPv4Address">
  <xs:restriction base="xs:string">
    <xs:pattern value="((([1-9]?[0-9]|1[0-9]{0-1}|2[0-4]{0-1}|25[0-5])\.){3}([1-9]?[0-9]|1[0-9]{0-1}|2[0-4]{0-1}|25[0-5]))"/>
  </xs:restriction>
</xs:simpleType>

</xs:schema>
```

Description:



QosManager:1 Erratum

- TrafficHandle: String. Required. Every Traffic Descriptor on the network must have a unique traffic handle. When the value of trafficHandle is not known, it should be represented by a null string.
- TrafficID: The unique identifier for the traffic. Required. The traffic identifier may be defined via any combination of its components. No one single component is required.
- SourceIP, DestinationIP: String (IP Address in the dotted decimal format). Optional.
- SourcePort, DestinationPort: TCP/UDP port number. Optional.
- IpProtocol: These are IANA assigned IP protocol numbers from 0-255. IGD trafficID defines IPProtocol in terms of these numbers. For more information:
<http://www.iana.org/assignments/protocol-numbers>
- AvailableOrderedTspecList: Contains one or more Tspec components. The Tspec index reflects the order of preference. Tspecs with smaller index values are more preferred.
availableOrderedTspecList is Required.
- Tspec: Tspec contains a description of Content QoS Requirements. In UPnP AV scenario, this information is extracted from the Content Directory Service of the Media Server. Tspec is represented in the form of an XML structure in the Content Directory Service. UPnP AV Control Point uses CDS:Browse and CDS:Search action calls to acquire the Tspec(s) associated with the content and inserts the Tspec XML into the Traffic Descriptor.

Tspec XML structure consists of the following:

- TspecIndex: Unique numerical index associated with a particular Tspec. The value of Tspec index indicates preference (as defined by the application or the end user). A Tspec with smaller Index is more preferred compared to a Tspec with larger Index.
- AvTransportUri: Unique URI associated with the UPnP AV content item for which QoS is being requested. Mandatory only in case of three box UPnP AV model.
- AvTransportInstanceId: Unique Instance ID associated with the UPnP AV Transport service associated with the content item for which QoS is being requested. Mandatory only in case of three box UPnP AV model.
- TrafficClass: The traffic class associated with the traffic stream. This is an enumerated variable that can be assigned to one of the following list of values:
 - Network Control
 - Streaming Control
 - Voice
 - AV
 - Data
 - Audio
 - Images
 - Gaming
 - Other
 - Background

TspecIndex must be filled in by the Control Point. Other Tspec parameters may be omitted if they are not known.

- ActiveTspecIndex: Integer index of active Tspec. Required when invoking QosDevice service. Optional when invoking QosManager.
- TrafficImportanceNumber: Traffic Importance (0-7) will be provided to QoS Device by the QosManager. In the absence of Policy Holder the QosManager will use default traffic class, as defined in Theory of Operation.
- QosBoundarySourceAddress: If a traffic stream originates outside the home network (e.g. on the Internet), the CP application must provide the QoS boundary point IP address. This address will be treated as the QoS termination point for UPnP QoS. Optional, because it is applicable only to traffic streams originating outside the home network. It should be noted that this address is not part of traffic identifier, because the IP packets will carry the IP address of the actual source



QoSManager:1 Erratum

- address. QoS boundary point is used by QoSManager for decisions related to path determination and device selection
- **QoSBoundaryDestinationAddress:** If a traffic stream terminates outside the home network (e.g. on the Internet), the CP application must provide the QoS boundary point IP address. This address will be treated as the QoS termination point for UPnP QoS. Optional, because it is applicable only to traffic streams terminating outside the home network. It should be noted that this address is not part of traffic identifier, because the IP packets will carry the IP address of the actual destination address. QoS boundary point is used by QoSManager for decisions related to path determination and device selection.
 - **OptionalPolicyParams:** Various optional parameters can be provided. Optional.
 - **UserName:** User Name field contains a string associated with the user who requested a particular traffic stream. This field, if present, shall identify the name of the user initiating the UPnP QoS Action. The field must be ≤ 64 UTF-8 characters. At present this specification does not define the semantics for this field. This field represent the name of the user initiating the QoS action. Examples include, "Dad", "Jimmy", "CN#1234567". The later is an exmple of a name provide by a service provider to uniquely identify its individual customers.
 - **CpName:** CP Name field contains a string associated with the Control Point requesting the traffic stream. This field, if present, shall identify the name of the control point initiating the UPnP QoS Action. The field must be ≤ 64 UTF-8 characters. At present this specification does not define the semantics for this field. The CpName may specify the brand name of the control point or it may indicate the location of the control point such as "living room"
 - **VendorApplicationName:** Vendor ApplicationName field contains a single URI string associated with an application. This field, if present, shall identify the name of the application initiating the UPnP QoS Action. For applications specified by vendors, the value of this field must begin with "urn:", followed by an ICANN domain name owned by the vendor, followed by ":application:", followed by an application name. i.e. the value of this field should be constructed as follows:

urn:domain-name:application:appl-name

domain-name: Application vendor domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be ≤ 64 UTF-8 characters.

appl-name: Name of the application provided by a vendor. This must follow the syntax specified for Name Specific String (NSS) in the RFC 2141. The *appl-name* field must be ≤ 64 UTF-8 characters. The *application-name* must not contain the colon character.

When requesting QoS it may be useful to know the manufacturer and name of the application software client or server that will process the traffic stream.

- **PortName:** Port Name field contains a single URI string associated with the fixed port used by an application for its connection set-up. This field, if present, shall identify the port number used by the application. The value of this field must begin with "urn:", followed by an ICANN domain name owned by the vendor, followed by ":port:", followed by the port name. i.e. the value of this field should be constructed as follows

urn:domain-name:port:port-number

domain-name: Application vendor domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be ≤ 64 UTF-8 characters.

port-number: A fixed port number used by the application either for connection set-up. This must be an integer value in the range 1 to 65535. If the application uses a range of port numbers then the starting port number of that range shall be specified. This value is either a vendor specific port used for the application or an IANA assigned port number for the application. The list of all the IANA assigned port numbers is maintained by IANA and is available at

<http://www.iana.org/assignments/port-numbers>.



QosManager:1 Erratum

These files indicate the port that is used by an application for its connection set-up. The use of IANA assigned port numbers is encouraged, but if not available for a particular application, a vendor assigned port number may be used.

- **ServiceProviderServiceName:** Service Provider Service Name field contains a single URI string associated with a service offered by a service provider. This field, if present, shall identify the name of the service offered by a service provider.. For services provided by service providers, the value of this field must begin with “urn:”, followed by an ICANN domain name owned by a service provider , followed by “service:”, followed by a servicename. i.e. the value of this field should be constructed as follows:

urn:domain-name:service:serv-name

domain-name: Service Provider domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be <= 64 UTF-8 characters.

serv-name: Name of the service provided by a service vendor. This must follow the syntax specified for Name Specific String (NSS) in the RFC 2141. The *serv-name* field must be <= 64 UTF-8 characters. The *serv-name* must not contain the colon character.

This represents the name of a particular service being offered by a service provider. Note that while multiple services provided by the same service provider may use the same software application, different Service offerings may be given different priorities (traffic importance).

- **TrafficLeaseTime:** **String** representing the lease time in seconds. Optional.
- **MediaServerConnectionId:** This is an optional field. The connection Id is obtained from the Media Server via CM::PrepareForConnection action. This field may be useful in case of UPnP AV based traffic streaming with multiple Tspecs. MediaServerConnectionId may be used by the QosManager to identify the traffic stream that is being setup by the UPnP AV Control Point which is requesting Qos. Please refer to UPnP QoS Architecture document for more details.
- **MediaRendererConnectionId:** This is an optional field. The connection Id is obtained from the Media Renderer via CM::PrepareForConnection action. This field may be useful in case of UPnP AV based streaming with multiple Tspecs. MediaRendererConnectionId may be used by the QosManager to identify the traffic stream that is being setup by the UPnP AV Control Point which is requesting Qos. Please refer to UPnP QoS Architecture document for more details.

New Text

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd"
targetNamespace="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd"
elementFormDefault="qualified" id="TrafficDescriptorv1">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      TrafficDescriptor schema.
      Copyright 2004,2005 UPnP(tm). All rights reserved.
    </xs:documentation>
  </xs:annotation>
  <xs:element name="TrafficDescriptor" type="TrafficDescriptorType"/>
  <xs:complexType name="TrafficDescriptorType">
    <xs:sequence>
      <xs:element name="TrafficHandle" type="xs:string" minOccurs="0"/>
      <xs:element name="TrafficId">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="SourceAddress" type="IpAddress"/>
            <xs:element name="SourcePort" type="IpPortNumber" minOccurs="0"/>
            <xs:element name="DestinationAddress" type="IpAddress"/>
            <xs:element name="DestinationPort" type="IpPortNumber" minOccurs="0"/>
            <xs:element name="IpProtocol" type="IpProtocolType" minOccurs="0"/>
            <xs:element name="v2TrafficId" type="v2TrafficIdType" minOccurs="0"/>
            <!-- allow any element except those from target namespace -->
            <xs:choice minOccurs="0" maxOccurs="unbounded">
              <xs:any namespace="##other" processContents="lax"/>
              <xs:any namespace="##local" processContents="lax"/>
            </xs:choice>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```



QosManager:1 Erratum

```
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="AvailableOrderedTspecList">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Tspec" type="TrafficSpecification" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="ActiveTspecIndex" type="xs:integer" minOccurs="0"/>
<xs:element name="TrafficImportanceNumber" type="TrafficImportance" default="0"
minOccurs="0"/>
<xs:element name="QosBoundarySourceAddress" type="IpAddress" minOccurs="0"/>
<xs:element name="QosBoundaryDestinationAddress" type="IpAddress" minOccurs="0"/>
<xs:element name="MediaServerConnectionId" type="xs:integer" minOccurs="0"/>
<xs:element name="MediaRendererConnectionId" type="xs:integer" minOccurs="0"/>
<xs:element name="TrafficLeaseTime" type="xs:integer" minOccurs="0"/>
<xs:element name="v2" type="v2ExtensionType" minOccurs="0"/>
<!-- allow any element except those from target namespace -->
<xs:choice minOccurs="0" maxOccurs="unbounded">
  <xs:any namespace="##other" processContents="lax"/>
  <xs:any namespace="##local" processContents="lax"/>
</xs:choice>
<xs:element name="OptionalPolicyParams" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="UserName" type="xs:string" minOccurs="0"/>
      <xs:element name="CpName" type="xs:string" minOccurs="0"/>
      <xs:element name="VendorApplicationName" type="xs:string" minOccurs="0"/>
      <xs:element name="PortName" type="xs:string" minOccurs="0"/>
      <xs:element name="ServiceProviderServiceName" type="xs:string" minOccurs="0"/>
      <xs:element name="v2OptionalParams" type="v2OptionalParamsType"
minOccurs="0"/>
      <!-- allow any element except those from target namespace -->
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:any namespace="##other" processContents="lax"/>
        <xs:any namespace="##local" processContents="lax"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="TrafficSpecification">
  <xs:sequence>
    <xs:element name="TspecIndex" type="xs:integer"/>
    <xs:element name="AvTransportUri" type="xs:string" minOccurs="0"/>
    <xs:element name="AvTransportInstanceId" type="xs:integer" minOccurs="0"/>
    <xs:element name="TrafficClass" type="TrafficClass" minOccurs="0"/>
    <xs:element name="v2TrafficSpecification" type="v2TrafficSpecificationType"
minOccurs="0"/>
    <!-- allow any element except those from target namespace -->
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:any namespace="##other" processContents="lax"/>
      <xs:any namespace="##local" processContents="lax"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="TrafficClass">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NetworkControl"/>
    <xs:enumeration value="StreamingControl"/>
    <xs:enumeration value="Voice"/>
    <xs:enumeration value="AV"/>
    <xs:enumeration value="Data"/>
    <xs:enumeration value="Audio"/>
    <xs:enumeration value="Image"/>
    <xs:enumeration value="Gaming"/>
    <xs:enumeration value="Other"/>
    <xs:enumeration value="Background"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="TrafficImportance">
  <xs:restriction base="xs:nonNegativeInteger">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="7"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="IpPortNumber">
```



QosManager:1 Erratum

```
<xs:restriction base="xs:nonNegativeInteger">
  <xs:minInclusive value="0"/>
  <xs:maxInclusive value="65535"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="IpProtocolType">
  <xs:restriction base="xs:nonNegativeInteger">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="255"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="IPv6Address">
  <xs:restriction base="xs:hexBinary">
    <xs:length value="32"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="IPv4Address">
  <xs:restriction base="xs:string">
    <xs:pattern value="([1-9]?[0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.([1-9]?[0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="PrefixLength">
  <xs:restriction base="xs:positiveInteger">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="128"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="IpAddress">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Ipv4" type="IPv4Address"/>
      <xs:element name="Ipv6" type="IPv6Address"/>
    </xs:choice>
    <xs:element name="PrefixLength" type="PrefixLength" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="v2ExtensionType">
  <xs:sequence>
    <xs:element name="PolicyHolderId" type="xs:string" minOccurs="0"/>
    <xs:element name="PolicyLastModified" type="xs:string" minOccurs="0"/>
    <xs:element name="PolicyModifyingUserName" type="xs:string" minOccurs="0"/>
    <xs:element name="PolicyHolderConfigUrl" type="xs:string" minOccurs="0"/>
    <xs:element name="v3" type="v3ExtensionType" minOccurs="0"/>
    <!-- allow any element except those from (v2) target namespace -->
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:any namespace="##other" processContents="lax"/>
      <xs:any namespace="##local" processContents="lax"/>
    </xs:choice>
  </xs:sequence>
  <xs:anyAttribute/>
</xs:complexType>
<xs:complexType name="v3ExtensionType">
  <xs:sequence>
    <xs:any namespace="##targetNamespace" processContents="lax" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute/>
</xs:complexType>
<xs:complexType name="v2TrafficIdType">
  <xs:sequence>
    <xs:element name="v3TrafficId" type="v3TrafficIdType" minOccurs="0"/>
    <!-- allow any element except those from (v2) target namespace -->
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:any namespace="##other" processContents="lax"/>
      <xs:any namespace="##local" processContents="lax"/>
    </xs:choice>
  </xs:sequence>
  <xs:anyAttribute/>
</xs:complexType>
<xs:complexType name="v3TrafficIdType">
  <xs:sequence>
    <xs:any namespace="##targetNamespace" processContents="lax" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute/>
</xs:complexType>
<xs:complexType name="v2OptionalParamsType">
  <xs:sequence>
    <xs:element name="v3OptionalParams" type="v3OptionalParamsType" minOccurs="0"/>
    <!-- allow any element except those from (v2) target namespace -->
    <xs:choice minOccurs="0" maxOccurs="unbounded">
```



QosManager:1 Erratum

```
<xs:any namespace="##other" processContents="lax"/>
<xs:any namespace="##local" processContents="lax"/>
</xs:choice>
</xs:sequence>
<xs:anyAttribute/>
</xs:complexType>
<xs:complexType name="v3OptionalParamsType">
  <xs:sequence>
    <xs:any namespace="##targetNamespace" processContents="lax" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute/>
</xs:complexType>
<xs:complexType name="v2TrafficSpecificationType">
  <xs:sequence>
    <xs:element name="v3TrafficSpecification" type="v3TrafficSpecificationType"
minOccurs="0"/>
    <!-- allow any element except those from (v2) target namespace -->
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:any namespace="##other" processContents="lax"/>
      <xs:any namespace="##local" processContents="lax"/>
    </xs:choice>
  </xs:sequence>
  <xs:anyAttribute/>
</xs:complexType>
<xs:complexType name="v3TrafficSpecificationType">
  <xs:sequence>
    <xs:any namespace="##targetNamespace" processContents="lax" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute/>
</xs:complexType>
</xs:schema>
```

Description:

- TrafficHandle: String. Required. Every Traffic Descriptor on the network must have a unique traffic handle. When the value of trafficHandle is not known, it should be represented by a null string.
- TrafficID: The unique identifier for the traffic. Required. The traffic identifier may be defined via any combination of its components. No one single component is required. **At least SourceAddress and DestinationAddress IP addresses must be specified in the TrafficId structure.**
- SourceIP, DestinationIP: String (IP Address in the dotted decimal format). Optional.
- SourcePort, DestinationPort: TCP/UDP port number. Optional.
- IpProtocol: These are IANA assigned IP protocol numbers from 0-255. IGD trafficID defines IPProtocol in terms of these numbers. For more information:
<http://www.iana.org/assignments/protocol-numbers>
- AvailableOrderedTspecList: Contains one or more Tspec components. The Tspec index reflects the order of preference. Tspecs with smaller index values are more preferred. availableOrderedTspecList is Required.
- Tspec: Tspec contains a description of Content QoS Requirements. In UPnP AV scenario, this information is extracted from the Content Directory Service of the Media Server. Tspec is represented in the form of an XML structure in the Content Directory Service. UPnP AV Control Point uses CDS:Browse and CDS:Search action calls to acquire the Tspec(s) associated with the content and inserts the Tspec XML into the Traffic Descriptor.

Tspec XML structure consists of the following:

- TspecIndex: Unique numerical index associated with a particular Tspec. The value of Tspec index indicates preference (as defined by the application or the end user). A Tspec with smaller Index is more preferred compared to a Tspec with larger Index.
- AvTransportUri: Unique URI associated with the UPnP AV content item for which QoS is being requested. Mandatory only in case of three box UPnP AV model.
- AvTransportInstanceId: Unique Instance ID associated with the UPnP AV Transport service associated with the content item for which QoS is being requested. Mandatory only in case of three box UPnP AV model.
- TrafficClass: The traffic class associated with the traffic stream. This is an enumerated variable that can be assigned to one of the following list of values:
 - Network Control



QosManager:1 Erratum

- Streaming Control
- Voice
- AV
- Data
- Audio
- Image
- Gaming
- Other
- Background

TspecIndex must be filled in by the Control Point. Other Tspec parameters may be omitted if they are not known. Different Tspec(s) for the same traffic could be differentiated using traffic class. The following is a valid example of a Tspec list:

```
<Tspec>
  <TspecIndex>1</TspecIndex>
  <AvTransportUri>http://192.168.0.111/blah.mpg</AvTransportUri>
  <TrafficClass>AV</TrafficClass>
</Tspec>
<Tspec>
  <TspecIndex>2</TspecIndex>
  <AvTransportUri>
http://192.168.0.111/blah2.mp3</AvTransportUri>
  <TrafficClass>Audio</TrafficClass>
</Tspec>
```

- ActiveTspecIndex: Integer index of active Tspec. Required when invoking QosDevice service. Optional when invoking QosManager.
- TrafficImportanceNumber: Traffic Importance (0-7) will be provided to QoS Device by the QosManager. In the absence of Policy Holder the QosManager will use default traffic class, as defined in Theory of Operation.
- QosBoundarySourceAddress: If a traffic stream originates outside the home network (e.g. on the Internet), the CP application must provide the QoS boundary point IP address. This address will be treated as the QoS termination point for UPnP QoS. Optional, because it is applicable only to traffic streams originating outside the home network. It should be noted that this address is not part of traffic identifier, because the IP packets will carry the IP address of the actual source address. QoS boundary point is used by QosManager for decisions related to path determination and device selection. . The “Don’t care” situation that may apply for SourceAddress IP address do not apply here. Hence a Control Point must not fill 0.0.0.0 for the value. If the SourceAddress is not in the subnet and a QosBoundarySourceAddress was not provided, the QosManager must return an error code 752.
- QosBoundaryDestinationAddress: If a traffic stream terminates outside the home network (e.g. on the Internet), the CP application must provide the QoS boundary point IP address. This address will be treated as the QoS termination point for UPnP QoS. Optional, because it is applicable only to traffic streams terminating outside the home network. It should be noted that this address is not part of traffic identifier, because the IP packets will carry the IP address of the actual destination address. QoS boundary point is used by QosManager for decisions related to path determination and device selection. The “Don’t care” situation that may apply for DestinationAddress IP address do not apply here. Hence a Control Point must not fill 0.0.0.0 for the value. If the DestinationAddress is not in the subnet and a QosBoundaryDestinationAddress was not provided, the QosManager must return an error code 752.
- OptionalPolicyParams: Various optional parameters can be provided. Optional.
- UserName: User Name field contains a string associated with the user who requested a particular traffic stream. This field, if present, shall identify the name of the user initiating the UPnP QoS Action. The field must be <= 64 UTF-8 characters. At present this specification does not define the semantics for this field. This field represent the name of the user initiating the QoS action.



QoSManager:1 Erratum

Examples include, “Dad”, “Jimmy”, “CN#1234567”. The later is an example of a name provided by a service provider to uniquely identify its individual customers.

- **CpName:** CP Name field contains a string associated with the Control Point requesting the traffic stream. This field, if present, shall identify the name of the control point initiating the UPnP QoS Action. The field must be ≤ 64 UTF-8 characters. At present this specification does not define the semantics for this field. The CpName may specify the brand name of the control point or it may indicate the location of the control point such as “living room”
- **VendorApplicationName:** Vendor ApplicationName field contains a single URI string associated with an application. This field, if present, shall identify the name of the application initiating the UPnP QoS Action. For applications specified by vendors, the value of this field must begin with “urn:”, followed by an ICANN domain name owned by the vendor, followed by “:application:”, followed by an application name. i.e. the value of this field should be constructed as follows:

urn:domain-name:application:appl-name

domain-name: Application vendor domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be ≤ 64 UTF-8 characters.

appl-name: Name of the application provided by a vendor. This must follow the syntax specified for Name Specific String (NSS) in the RFC 2141. The *appl-name* field must be ≤ 64 UTF-8 characters. The *application-name* must not contain the colon character.

When requesting QoS it may be useful to know the manufacturer and name of the application software client or server that will process the traffic stream.

- **PortName:** Port Name field contains a single URI string associated with the fixed port used by an application for its connection set-up. This field, if present, shall identify the port number used by the application. The value of this field must begin with “urn:”, followed by an ICANN domain name owned by the vendor, followed by “:port:”, followed by the port name. i.e. the value of this field should be constructed as follows

urn:domain-name:port:port-number

domain-name: Application vendor domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be ≤ 64 UTF-8 characters.

port-number: A fixed port number used by the application either for connection set-up. This must be an integer value in the range 1 to 65535. If the application uses a range of port numbers then the starting port number of that range shall be specified. This value is either a vendor specific port used for the application or an IANA assigned port number for the application. The list of all the IANA assigned port numbers is maintained by IANA and is available at <http://www.iana.org/assignments/port-numbers>.

These fields indicate the port that is used by an application for its connection set-up. The use of IANA assigned port numbers is encouraged, but if not available for a particular application, a vendor assigned port number may be used.

- **ServiceProviderServiceName:** Service Provider Service Name field contains a single URI string associated with a service offered by a service provider. This field, if present, shall identify the name of the service offered by a service provider. For services provided by service providers, the value of this field must begin with “urn:”, followed by an ICANN domain name owned by a service provider, followed by “:service:”, followed by a servicename. i.e. the value of this field should be constructed as follows:

urn:domain-name:service:serv-name

domain-name: Service Provider domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be ≤ 64 UTF-8 characters.



QosManager:1 Erratum

serv-name: Name of the service provided by a service vendor. This must follow the syntax specified for Name Specific String (NSS) in the RFC 2141. The *serv-name* field must be <= 64 UTF-8 characters. The *serv-name* must not contain the colon character.

This represents the name of a particular service being offered by a service provider. Note that while multiple services provided by the same service provider may use the same software application, different Service offerings may be given different priorities (traffic importance).

- TrafficLeaseTime: Integer representing the lease time in seconds. Optional.
- MediaServerConnectionId: This is an optional field. The connection Id is obtained from the Media Server via CM::PrepareForConnection action. This field may be useful in case of UPnP AV based traffic streaming with multiple Tspecs. MediaServerConnectionId may be used by the QosManager to identify the traffic stream that is being setup by the UPnP AV Control Point which is requesting Qos. Please refer to UPnP QoS Architecture document for more details.
- MediaRendererConnectionId: This is an optional field. The connection Id is obtained from the Media Renderer via CM::PrepareForConnection action. This field may be useful in case of UPnP AV based streaming with multiple Tspecs. MediaRendererConnectionId may be used by the QosManager to identify the traffic stream that is being setup by the UPnP AV Control Point which is requesting Qos. Please refer to UPnP QoS Architecture document for more details.