UPnP QosDevice:2
Service Template Version 1.01

For UPnP Version 1.0
Status: Standardized DCP
Date: October 16, 2006
Document Version: 1.00

Authors	Company
Amol Bhagwat	CableLabs
Bruce Fairman	Sony
Daryl Hlasny	Sharp Laboratories of America
Gabe Frost	Microsoft Corporation
Jack Manbeck	TI
John McQueen	Broadcom
Michael van Hartskamp	Philips
Narm Gadiraju	Intel Corporation
Raj Bopardikar	Intel Corporation
Richard Bardini	Sony
Richard Chen	Philips
Stephen Palm	Broadcom

Contents

List of Tables

List of Figures

1. Overview and Scope

This service definition is compliant with the UPnP Device Architecture version 1.0.[Qos Architecture]

This service-type enables modeling of the 'QoS Device' function capabilities. The QosDevice:2 service is a function typically implemented in source, sink and intermediate network elements that are in the path of the traffic. The QosDevice Service is responsible for providing the appropriate network resources to traffic and state of the device as requested by QoS Management Entity as defined in the QosManager:2 Service. [QM]

This document does not address the procedure for end-to-end set up of a new traffic stream nor revoke an existing traffic stream.

1.1. Referenced Specifications

Unless explicitly stated otherwise herein, implementation of the mandatory provisions of any standard referenced by this specification shall be mandatory for compliance with this specification.

1.1.1. Normative References

This section lists the normative references used in this document and includes the tag inside square brackets that is used for each sub reference:

[Annex_G] – IEEE 802.1D-2004, Annex G, *IEEE Standard for Information technology - Telecommunications and information exchange between systems - IEEE standard for local and metropolitan area networks - Common specifications - Media access control (MAC) Bridges*, 2004.

[XML] – *Extensible Markup Language (XML) 1.0 (Second Edition)*, T. Bray, J.Paoli, C. M. Sperberg-McQueen, E Maler, eds. W3C Recommendations, 6 October 2000.

[QM] – UPnP QosManager:2 Service Document: Note that only the schema definition used for the A_ARG_TYPE_TrafficDescriptor is normative for UPnP QosDevice:2 service specification and the schema is defined in the UPnP QosManager:2 document.

[DEVICE] - *UPnP Device Architecture, version 1.0*.

[RFC3339] – Date and Time on the Internet: Timestamps, G. Klyne, July 2002. http://www.ietf.org/rfc/rfc3339.txt

1.1.2. Informative References

This section lists the informative references used in this document and includes the tag inside square brackets that is used for each sub reference:

[Qos Architecture] – UPnP Qos Architecture:2 Document

2. Service Modeling Definitions

2.1. ServiceType

The following service type identifies a service that is compliant with this template:

urn:schemas-upnp-org:service:QosDevice:2

The shorthand QosDevice is used herein to refer to this type of service.

2.2. Namespaces

The XML [XML] in this document should be read as if the following namespace definitions are in effect.

`xmlns="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd"[QM]`

2.3. State Variables

Reader Note: For first-time reader, it may be more insightful to read the action definitions before reading the state variable definitions.

2.3.1. Derived data Types

This section defines some derived data types that are represented as UPnP string data types with special syntax.

2.3.1.1. XML Fragments as UPnP Arguments

The UPnP QoS Framework often uses XML Fragments as arguments in UPnP actions. The containing UPnP data type is a string. This places restrictions on a string's content; it has to represent a well-formed XML fragment (this includes a complete XML document).

In their XML fragments, implementations may use an explicit reference to appropriate name spaces.

At several places in the XML schemas there is room for vendor differentiation or future revisions through the use of the "any"-tag. When extending UPnP-QoS with their own XML tags, vendors should use a name space to prevent collisions of their tags with those of other vendors. It is recommended that implementations are not required to retrieve the corresponding schemas from the Internet.

In order to maintain the extensibility of the namespace, all future modification of the schema definition will be proper supersets. The URN will not change even when the service version number changes.

Finally, an XML fragment, in adherence to the UPnP V1.0 architecture [Qos Architecture], needs to be escaped by using the normal XML rules, [XML] Section 2.4 Character Data and Markup, before embedding it in a SOAP request or response message. The XML escaping rules are summarized from the [XML] reference mentioned above:

- The (<) character is encoded as (&lt;)
- The (>) character is encoded as (&gt;)
- The (&) character is encoded as (&amp;)
- The (") character is encoded as (&quot;)

- The (‘) character is encoded as (&apos;)

Table 2-1: State Variables

Variable Name	Req. or Opt.[1]	Data Type	Allowed Value [2]	Default Value [2]	Eng. Units
A_ARG_TYPE_TrafficDescriptor	R	String (XML fragment)	See §2.3.2	n/a	n/a
A_ARG_TYPE_TrafficDescriptorsPerInterface	R	String (XML fragment)	See §2.3.3	n/a	n/a
A_ARG_TYPE_TrafficHandle	R	String	See §2.3.4	n/a	n/a
A_ARG_TYPE_NumTraffficDescriptors	R	ui4	See §2.3.5	n/a	n/a
A_ARG_TYPE_QosDeviceCapabilities	R	String (XML fragment)	See §2.3.6	n/a	n/a
A_ARG_TYPE_QosDeviceState	R	String (XML fragment)	See §2.3.7	n/a	n/a
PathInformation	O	String (XML fragment)	See §2.3.8	n/a	n/a
A_ARG_TYPE_QosDeviceInfo	O	String (XML fragment)	See §2.3.9	n/a	n/a
A_ARG_TYPE_QosStateId	R	String		n/a	n/a
A_ARG_TYPE_NumRotameterObservations	O	ui4	See §2.3.10	1	n/a
A_ARG_TYPE_RotameterInformation	O	String (XML fragment)	See §2.3.11	n/a	n/a

Variable Name	Req. or Opt.[1]	Data Type	Allowed Value [2]	Default Value [2]	Eng. Units
A_ARG_TYPE_ConfRotameterObservations	O	String (XML fragment)	See §2.3.12	n/a	n/a
MostRecentStreamAction	O	String (XML fragment)	See §2.3.13	n/a	n/a
A_ARG_TYPE_MaxPossibleRotameterObservations	O	ui4	See §2.3.14	1	n/a

1. R = Required, O = Optional, X = Non-standard.

2. Values listed in this column are required. To specify standard optional values or to delegate assignment of values to the vendor, you must reference a specific instance of an appropriate table below.

2.3.2. A_ARG_TYPE_TrafficDescriptor

This is an escaped XML string, as specified in section 2.3.1.1, which contains QoS related information for a traffic stream. Refer to [QM] document, for details of this XML fragment using the namespace, `xmlns="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd"`.

2.3.3. A_ARG_TYPE_TrafficDescriptorsPerInterface

This is an escaped XML string, as specified in section 2.3.1.1, which contains the list of traffic descriptors that are associated with a network interface on a given QosDevice.

This argument is described by the schema identified by "`http://www.upnp.org/schemas/TrafficDescriptorsPerInterface.xsd`" and located at "`http://www.upnp.org/schemas/qos/TrafficDescriptorsPerInterface-v2.xsd`".

2.3.3.1. Description of fields in the TrafficDescriptorsPerInterface structure

The TrafficDescriptorsPerInterface is a complex structure that consists of one or more entries of 'TdInterfacePair'. TdInterfacePair lists one TrafficDescriptor, followed by the InterfaceId of the associated of the interface. Here are the details about these two parameters:

TrafficDescriptor: This field describes a TrafficDescriptor associated with an Interface. An Interface can have multiple associated TrafficDescriptor objects.

<u>InterfaceId</u>: This is a required field. The value is of type string and is a unique value for a given device and is used to identify the interface.

2.3.3.2. Sample argument XML string

```
<?xml version="1.0" encoding="UTF-8"?>
<TrafficDescriptorsPerInterface
xmlns="http://www.upnp.org/schemas/TrafficDescriptorsPerInterface.xsd"
xmlns:td="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.upnp.org/schemas/TrafficDescriptorsPerInterface.xsd
```

```
http://www.upnp.org/schemas/qos/TrafficDescriptorsPerInterface-v2.xsd">
	<TdInterfacePair>
		<TrafficDescriptor>
			<td:TrafficHandle>kiwin</td:TrafficHandle>
			<td:TrafficId>
				<td:SourceAddress>
					<td:Ipv4>192.168.1.50</td:Ipv4>
				</td:SourceAddress>
				<td:SourcePort>23</td:SourcePort>
				<td:DestinationAddress>
					<td:Ipv4>192.168.1.50</td:Ipv4>
				</td:DestinationAddress>
				<td:DestinationPort>23</td:DestinationPort>
				<td:IpProtocol>1</td:IpProtocol>
			</td:TrafficId>
			<td:AvailableOrderedTspecList>
				<td:Tspec>
					<td:TspecIndex>300</td:TspecIndex>
					<td:TrafficClass>AV</td:TrafficClass>
				</td:Tspec>
				<td:Tspec>
					<td:TspecIndex>2</td:TspecIndex>
					<td:TrafficClass>Audio</td:TrafficClass>
				</td:Tspec>
			</td:AvailableOrderedTspecList>
			<td:ActiveTspecIndex>300</td:ActiveTspecIndex>
			<td:TrafficImportanceNumber>5</td:TrafficImportanceNumber>
			<td:OptionalPolicyParams>
				<td:CpName>Amy's CP</td:CpName>
			</td:OptionalPolicyParams>
		</TrafficDescriptor>
		<InterfaceId>eth0</InterfaceId>
	</TdInterfacePair>
</TrafficDescriptorsPerInterface>
```

2.3.4. A_ARG_TYPE_TrafficHandle

A_ARG_TYPE_TrafficHandle is a string to identify a traffic stream. Refer to the [QM] document for more details.

2.3.5. A_ARG_TYPE_NumTrafficDescriptors

This is an integer argument. Refer to the [QM] document for more details.

2.3.6. A_ARG_TYPE_QosDeviceCapabilities

This is an escaped XML fragment, as specified in section 2.3.1.1, and contains information describing a device's QoS capabilities.

This argument is described by the schema identified by "`http://www.upnp.org/schemas/QosDeviceCapabilities.xsd`" and located at "`http://www.upnp.org/schemas/qos/QosDeviceCapabilities-v2.xsd`".

2.3.6.1. Description of fields in the QosDeviceCapabilities structure

Interface: This is a required field and defined as an XML element. This field describes a network interface on the QosDevice. An Interface definition is required for each interface supported by the device. This information is provided even if the physical interface is down at a given time.

MacAddress: This is a required field if a given interface has an associated MacAddress. Provides the MAC address of the Interface.

InterfaceId: This is a required field. The value is of type string and is a unique value for a given device and is used to identify the interface.

IanaTechnologyType: The IanaTechnologyType is an integer that indicates media interface type, such as 802.3 (value=6) or 802.11 (value=71). The allowed integer values for this parameter are specified in the IANA reference ifType-MIB <http://www.iana.org/assignments/ianaiftype-mib>.

AdmissionControlSupported: This is a required field. AdmissionControlSupported field indicates whether the interface on the device is capable of performing device level admission control. For this version of the specification the device must set this parameter to a value of "No".

PacketTaggingSupported: This is a required field. PacketTaggingSupported field indicates whether the device is capable of tagging layer2 priorities on the outgoing interface. This field can report only one of two values "Yes" or "No".

NativeQos: This is an optional field. Contains one of the 2 values (Prioritized, BestEffort).

MaxPhyRate: Indicates the maximum PHY rate of the interface and expressed as a value of type UnsignedInt. This parameter is optional and indicates (Units) phy rate measured in bits/sec.

ChannelInformation: Indicates the channel number of the IanaTechnologyType, if the technology supports channels. For example, 802.11 (value=71) supports multiple channels. Expressed as a value of type UnsignedInt.

2.3.6.2. Sample argument XML string

```
<?xml version="1.0" encoding="UTF-8"?>
<QosDeviceCapabilities
xmlns="http://www.upnp.org/schemas/QosDeviceCapabilities.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.upnp.org/schemas/QosDeviceCapabilities.xsd
       http://www.upnp.org/schemas/qos/QosDeviceCapabilities-v2.xsd">
  <Interface>
    <InterfaceId>eth0</InterfaceId>
    <MacAddress>0212abcdef11</MacAddress>
    <IanaTechnologyType>6</IanaTechnologyType>
    <AdmissionControlSupported>No</AdmissionControlSupported>
    <PacketTaggingSupported>Yes</PacketTaggingSupported>
    <NativeQos>Prioritized</NativeQos>
    <MaxPhyRate>100000000</MaxPhyRate>
  </Interface>
  <Interface>
    <InterfaceId>eth1</InterfaceId>
    <MacAddress>0212abcdef12</MacAddress>
    <IanaTechnologyType>71</IanaTechnologyType>
    <AdmissionControlSupported>No</AdmissionControlSupported>
    <PacketTaggingSupported>Yes</PacketTaggingSupported>
    <NativeQos>Prioritized</NativeQos>
    <MaxPhyRate>3000000</MaxPhyRate>
    <v2>
      <ChannelInformation>6</ChannelInformation>
    </v2>
  </Interface>
  <Interface>
    <InterfaceId>eth2</InterfaceId>
    <MacAddress>0212abcdef13</MacAddress>
    <IanaTechnologyType>6</IanaTechnologyType>
    <AdmissionControlSupported>No</AdmissionControlSupported>
    <PacketTaggingSupported>Yes</PacketTaggingSupported>
    <NativeQos>BestEffort</NativeQos>
    <MaxPhyRate>5000000</MaxPhyRate>
  </Interface>
  <Interface>
    <InterfaceId>example1</InterfaceId>
    <MacAddress>0212abcdefff</MacAddress>
    <IanaTechnologyType>12</IanaTechnologyType>
    <AdmissionControlSupported>No</AdmissionControlSupported>
    <PacketTaggingSupported>Yes</PacketTaggingSupported>
    <NativeQos>BestEffort</NativeQos>
    <MaxPhyRate>5000000</MaxPhyRate>
    <v2>
```

```
      <ChannelInformation>6</ChannelInformation>
    </v2>
  </Interface>
</QosDeviceCapabilities>
```

2.3.7. A_ARG_TYPE_QosDeviceState

A_ARG_TYPE_QosDeviceState is a structure that provides information about a device's current QoS state.

This argument is described by the schema identified by "`http://www.upnp.org/schemas/QosDeviceState.xsd`" and located at "`http://www.upnp.org/schemas/qos/QosDeviceState-v2.xsd`".

2.3.7.1. Description of fields in the A_ARG_TYPE_QosDeviceState *structure*

QosStateId: This is a required field. It must identify the QoS-related state of the QosDevice. In particular it must change after successful invocations of **SetupTrafficQos** or **ReleaseTrafficQos**. There may be other reasons a QosDevice changes QosStateId, but when the QosStateId is the same at two instants in time, all relevant Qos-state must be the same. Read theory of operation for more details as to how this parameter is used.

Interface: This is a required field and defines an interface. An Interface definition is required for each interface supported by the device.

InterfaceId: This is a required field. The value is of type string and unique for a device to identify the interface uniquely.

IpAddress: This is an optional field. This specifies the IP Address of the interface. This is optional for interfaces not configured with an IP Address. However the IP Address of configured interfaces must advertise this value.

InterfaceAvailability: This is a required field. The value of 0 indicates that the interface is not available. A value of 1 indicates the interface is available which may include being in power-save mode.

2.3.7.2. Sample argument XML string

```
<?xml version="1.0" encoding="UTF-8"?>
<QosDeviceState
 xmlns="http://www.upnp.org/schemas/QosDeviceState.xsd"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.upnp.org/schemas/QosDeviceState.xsd
 http://www.upnp.org/schemas/qos/QosDeviceState-v2.xsd">
 <QosStateId>MyStateId001</QosStateId>
 <Interface>
   <InterfaceId>eth0</InterfaceId>
   <IpAddress>
     <Ipv4>10.10.145.24</Ipv4>
   </IpAddress>
   <InterfaceAvailability>1</InterfaceAvailability>
 </Interface>
 <Interface>
   <InterfaceId>eth1</InterfaceId>
   <InterfaceAvailability>0</InterfaceAvailability>
 </Interface>
 <Interface>
   <InterfaceId>eth2</InterfaceId>
   <IpAddress>
     <Ipv4>10.10.144.23</Ipv4>
   </IpAddress>
   <InterfaceAvailability>1</InterfaceAvailability>
 </Interface>
</QosDeviceState>
```

2.3.8. PathInformation

PathInformation is a structure that provides MAC address information about devices reachable through each active interface.

This argument is described by the schema identified by "`http://www.upnp.org/schemas/PathInformation.xsd`" and located at "`http://www.upnp.org/schemas/qos/PathInformation-v2.xsd`".

2.3.8.1. Description of fields in PathInformation *structure*

LinkReachableMacs: This is a required field. A LinkReachableMacs definition is required for each available link supported by the device. For a device with physical media dedicated to an interface (such as Ethernet) there will be a LinkReachableMacs definition for each physical interface. For a device with a shared media (such as 802.11) there will be a LinkReachableMacs definition for each device pair where communication is supported by the device.

LinkId: This is a required field. Its value is of type string, it must be unique within the device. It identifies the layer-2 link.

MacAddress: This is a required field when available. Provides the MAC address of the interface for an end point device.

ReachableMac: Provides the MAC address(es) of end point devices that are reachable through the link, if any.

BridgedId: Identifies the links that are bridged together. All links that have the same BridgeID are interconnected within the device such that layer-2 frames are forwarded between them.

2.3.8.2. Sample argument XML string – PC with two network interfaces

This is an example of an end point network device with two network interfaces.

```
<?xml version="1.0" encoding="UTF-8"?>
<DeviceReachableMacs
  xmlns="http://www.upnp.org/schemas/PathInformation.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/PathInformation.xsd
  http://www.upnp.org/schemas/qos/PathInformation-v2.xsd">
        <LinkReachableMacs>
                <LinkId>eth0</LinkId>
                <MacAddress>112233aabb03</MacAddress>
        </LinkReachableMacs>
        <LinkReachableMacs>
                <LinkId>eth1</LinkId>
                <MacAddress>112233aabb02</MacAddress>
        </LinkReachableMacs>
</DeviceReachableMacs>
```

2.3.8.3. Sample argument XML string – PC with two network interfaces that are both end point device and bridged

Similar to the previous example this is an example of an end point network device with two network interfaces. However this device all forwards layer-2 frames between the two network interfaces.

```
<?xml version="1.0" encoding="UTF-8"?>
<DeviceReachableMacs
  xmlns="http://www.upnp.org/schemas/PathInformation.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/PathInformation.xsd
  http://www.upnp.org/schemas/qos/PathInformation-v2.xsd">
  <LinkReachableMacs>
      <LinkId>eth0</LinkId>
      <MacAddress>112233aabb03</MacAddress>
  </LinkReachableMacs>
  <LinkReachableMacs>
      <LinkId>eth1</LinkId>
      <MacAddress>112233aabb02</MacAddress>
  </LinkReachableMacs>
  <LinkReachableMacs>
      <LinkId>eth0</LinkId>
      <BridgeId>Bridge</BridgeId>
      <ReachableMac>112233aabb03</ReachableMac>
      <ReachableMac>112233aabb02</ReachableMac>
      <ReachableMac>112233aabb01</ReachableMac>
      <ReachableMac>112233aabb04</ReachableMac>
  </LinkReachableMacs>
  <LinkReachableMacs>
      <LinkId>eth1</LinkId>
      <BridgeId>Bridge0</BridgeId>
      <ReachableMac>112233aabb05</ReachableMac>
  </LinkReachableMacs>
</DeviceReachableMacs>
```

2.3.8.4. *Sample argument XML string –Four port Ethernet Switch*

This is an example of a layer-2 switching device that interconnects four physical Ethernet ports. The device supports layer-2 frame forwarding between all ports.

```
<?xml version="1.0" encoding="UTF-8"?>
<DeviceReachableMacs
  xmlns="http://www.upnp.org/schemas/PathInformation.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/PathInformation.xsd
  http://www.upnp.org/schemas/qos/PathInformation-v2.xsd">
  <LinkReachableMacs>
      <LinkId>eth0</LinkId>
      <BridgeId>Bridge0</BridgeId>
      <ReachableMac>112233aabb03</ReachableMac>
  </LinkReachableMacs>
  <LinkReachableMacs>
      <LinkId>eth1</LinkId>
      <BridgeId>Bridge0</BridgeId>
      <ReachableMac>112233aabb07</ReachableMac>
      <ReachableMac>112233aabb05</ReachableMac>
  </LinkReachableMacs>
  <LinkReachableMacs>
      <LinkId>eth2</LinkId>
      <BridgeId>Bridge0</BridgeId>
      <ReachableMac>112233aabb02</ReachableMac>
      <ReachableMac>112233aabb01</ReachableMac>
      <ReachableMac>112233aabb04</ReachableMac>
  </LinkReachableMacs>
  <LinkReachableMacs>
      <LinkId>eth3</LinkId>
      <BridgeId>Bridge0</BridgeId>
  </LinkReachableMacs>
</DeviceReachableMacs>
```

2.3.8.5. *Sample argument XML string – Wireless AP with one Ethernet Interface*

This is an example of a wireless access point with three associated wireless stations and a single Ethernet port. The device supports layer-2 frame forwarding between all links. This includes forwarding between wireless stations or to the Ethernet interface.

```
<?xml version="1.0" encoding="UTF-8"?>
<DeviceReachableMacs
  xmlns="http://www.upnp.org/schemas/PathInformation.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/PathInformation.xsd
  http://www.upnp.org/schemas/qos/PathInformation-v2.xsd">
  <LinkReachableMacs>
      <LinkId>WL0</LinkId>
      <BridgeId>Bridge0</BridgeId>
      <ReachableMac>112233aabb02</ReachableMac>
  </LinkReachableMacs>
  <LinkReachableMacs>
```

```
      <LinkId>WL1</LinkId>
      <BridgeId>Bridge0</BridgeId>
      <ReachableMac>112233aabb01</ReachableMac>
  </LinkReachableMacs>
  <LinkReachableMacs>
      <LinkId>WL2</LinkId>
      <BridgeId>Bridge0</BridgeId>
      <ReachableMac>112233aabb04</ReachableMac>
      <ReachableMac>112233aabb09</ReachableMac>
  </LinkReachableMacs>
  <LinkReachableMacs>
     <LinkId>eth0</LinkId>
     <BridgeId>Bridge0</BridgeId>
     <ReachableMac>112233aabb03</ReachableMac>
     <ReachableMac>112233aabb07</ReachableMac>
     <ReachableMac>112233aabb05</ReachableMac>
  </LinkReachableMacs>
</DeviceReachableMacs>
```

2.3.8.6. Sample argument XML string – Bridge device between Wireless station and Ethernet

This is an example of a bridging device with two interfaces on different network technologies. It does layer-2 forwarding of frames between wireless station interface and the wired Ethernet interface.

```
<?xml version="1.0" encoding="UTF-8"?>
<DeviceReachableMacs
  xmlns="http://www.upnp.org/schemas/PathInformation.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/PathInformation.xsd
  http://www.upnp.org/schemas/qos/PathInformation-v2.xsd">
  <LinkReachableMacs>
      <LinkId>WL0</LinkId>
      <BridgeId>Bridge0</BridgeId>
  </LinkReachableMacs>
  <LinkReachableMacs>
      <LinkId>eth0</LinkId>
      <BridgeId>Bridge0</BridgeId>
      <ReachableMac>112233aabb04</ReachableMac>
  </LinkReachableMacs>
</DeviceReachableMacs>
```

2.3.9. A_ARG_TYPE_QosDeviceInfo

A_ARG_TYPE_QosDeviceInfo is a structure that provides port numbers and protocol information associated with a traffic stream.

This argument is described by the schema identified by "`http://www.upnp.org/schemas/QosDeviceInfo.xsd`" and located at "`http://www.upnp.org/schemas/qos/QosDeviceInfo-v2.xsd`".

2.3.9.1. *Description of fields in* **A_ARG_TYPE_QosDeviceInfo** *structure*

TrafficHandle: This is a required field that identifies the Traffic Descriptor for which QoS device information is being returned.

SourcePort: This value represents the source port that is going to be used for a traffic stream.

DestinationPort: This value represents the destination port that is going to be used for a traffic stream.

Protocol: This field represents the IANA assigned protocol number of the traffic stream.

2.3.9.2. Sample argument XML string

```
<?xml version="1.0" encoding="UTF-8"?>
<QosDeviceInfo
  xmlns="http://www.upnp.org/schemas/QosDeviceInfo.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/QosDeviceInfo.xsd
  http://www.upnp.org/schemas/qos/QosDeviceInfo-v2.xsd">
        <TrafficHandle>abcxyz</TrafficHandle>
        <SourcePort>1001</SourcePort>
        <DestinationPort>2003</DestinationPort>
        <IpProtocol>6</IpProtocol>
</QosDeviceInfo>
```

2.3.10.A_ARG_TYPE_NumRotameterObservations

This is a positive integer argument. This state variable indicates the number of Rotameter Observations per MAC address that a requesting Control Point is interested in receiving. If the QosDevice has this number of observations available, it must return the most recent (in time) observations indicated by the number A_ARG_TYPE_NumRotameterObservations.

2.3.11.A_ARG_TYPE_RotameterInformation

A_ARG_TYPE_RotameterInformation is a structure that provides MAC address and Rotameter information about devices reachable through each active interface.

This argument is described by the schema identified by “`http://www.upnp.org/schemas/RotameterInformation.xsd`” and located at “`http://www.upnp.org/schemas/qos/RotameterInformation-v2.xsd`”.

2.3.11.1.Description of fields in **RotameterInformation** *structure*

LinkReachableMacs: This is a required field. A LinkReachableMacs definition is required for each available link supported by the device.

LinkId: This is a required field. Its value is of type string, it must be unique within the device. It identifies the layer-2 link.

MacAddress: This is a required field when available. Provides the MAC address of the interface for an end point device.

BridgedId: Identifies the links that are bridged together. All links that have the same BridgeID are interconnected within the device such that layer-2 frames are forwarded between them.

RotameterObservation A Sequence of elements describing a Rotameter Observation

RotameterIndex An index that is incremented and is unique per observation on the reporting device. This can serve to correlate overlapping history snapshots to determine where they overlap.

ROPeriod Duration of the Observation period. ROPeriod shall be less than or equal to the MonitoringResolutionPeriod. Units: seconds (See Figure 2-1)

ReportingDateTime Time of Completion of Observation Period. Wall clock time formatted per section 5.6 of RFC 3339. Potentially non-synchronized with other devices on the network.

MonitorResolutionPeriod How often a Rotameter observation is initiated. Units: seconds (See Figure 2-1)

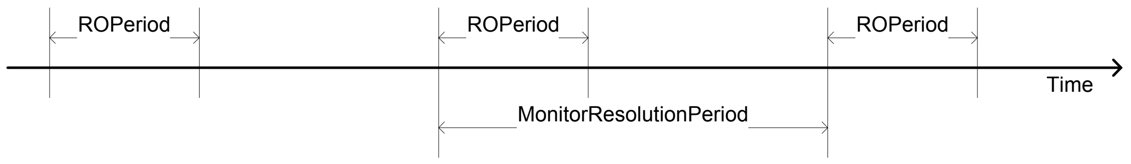

Figure 2-1 Relationship between ROPeriod and MonitorResolutionPeriod

If MonitorResolutionPeriod is sufficiently larger than ROPeriod, there is an increased likelihood that an ROPeriod would not capture a bursty traffic condition, i.e. ROBits would not account for these bursts. In Figure 2-1 above, one could visualize this traffic burst occurring between ROPeriods. Examples of such situations are media server filling the receive buffer in a digital media renderer or a large file transfer/download.

ROAddr Address of the device of the report. If it is the same address as the reporting device interface, then the Rotameter Observation is for all traffic to/from that reporting device. If the address is different than the reporting device, then the Rotameter Observation is for all traffic between the address and the reporting device.

ROBits is total number of bits in the Observation period (ROPeriod).

ROBits0 (Optional) Number of bits interpreted as TrafficImportanceNumber 0 in the Observation period

ROBits1 (Optional) Number of bits interpreted as TrafficImportanceNumber 1 in the Observation period

ROBits2 (Optional) Number of bits interpreted as TrafficImportanceNumber 2 in the Observation period

ROBits3 (Optional) Number of bits interpreted as TrafficImportanceNumber 3 in the Observation period

ROBits4 (Optional) Number of bits interpreted as TrafficImportanceNumber 4 in the Observation period

ROBits5 (Optional) Number of bits interpreted as TrafficImportanceNumber 5 in the Observation period

ROBits6 (Optional) Number of bits interpreted as TrafficImportanceNumber 6 in the Observation period

ROBits7 (Optional) Number of bits interpreted as TrafficImportanceNumber 7 in the Observation period

If the QosDevice employs separate priority queues for different traffic types, and is capable of managing separate traffic counters (total number of bits in and out of the queue) for each of these priority queues (per attached device), it would be valuable to do so. For example, if a WLAN AP has four priority queues (background, best-effort, video, and voice) and is capable of managing separate counters for each of these queues, each of ROBits1, ROBits0, ROBits5, and ROBits7 respectively should be implemented for each attached device. If the QosDevice is unable to manage separate counters for each priority queue (per attached device), implementing a single counter (ROBits) per attached device is a reasonable compromise.

To further this example, ROPeriod and MonitorResolutionPeriod (described below) are both configured at 1 second and there is only a single device attached to the WLAN AP. If managing counters per-priority-queue is possible for the AP, and the attached device sent two bursts of traffic; one at 1 Mbps for 1 second with no priority (i.e. best-effort), then another at 6 Mbps for 1 second with video priority, the counters for two requested observations (for the attached device) would contain:

Observation #1: ROBits (1000000), and ROBits0 (1000000). If only a single counter per-device is possible for this AP, the single counter would contain: ROBits (1000000).

Observation #2: ROBits (6000000), and ROBits5 (6000000). If only a single counter per-device is possible for this AP, the single counter would contain: ROBits (6000000).

If only a single observation was requested, the most recent would be returned, i.e. Observation #2 above. If an observation was requested after 1.5 seconds, i.e. between observation periods, the most recent complete observation would be returned (#1 above).

2.3.11.2.Sample argument XML string – PC with two network interfaces

This is an example of an end point network device with two network interfaces that are not currently making Rotameter Observations.

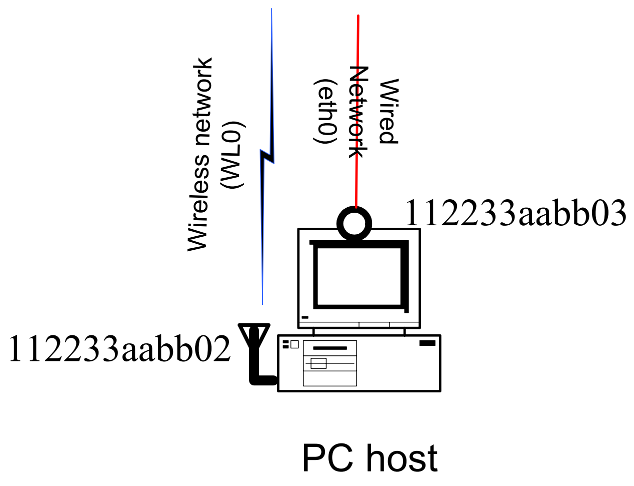

Figure 2-2 Example Network Rotameter Observation on a PC with two interfaces

```
<?xml version="1.0" encoding="UTF-8"?>
<RotameterInformation
  xmlns="http://www.upnp.org/schemas/RotameterInformation.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/RotameterInformation.xsd
  http://www.upnp.org/schemas/qos/RotameterInformation-v2.xsd">
  <LinkReachableMacs>
      <LinkId>eth0</LinkId>
      <MacAddress>112233aabb03</MacAddress>
  </LinkReachableMacs>
  <LinkReachableMacs>
      <LinkId>WL0</LinkId>
      <MacAddress>112233aabb02</MacAddress>
  </LinkReachableMacs>
</RotameterInformation>
```

2.3.11.3.Sample argument XML string – PC with two network interfaces that are both end point device

Similar to the previous example this is an example of an end point network device with two network interfaces. In this example the interfaces are actively connected and actively making Rotameter Observations.

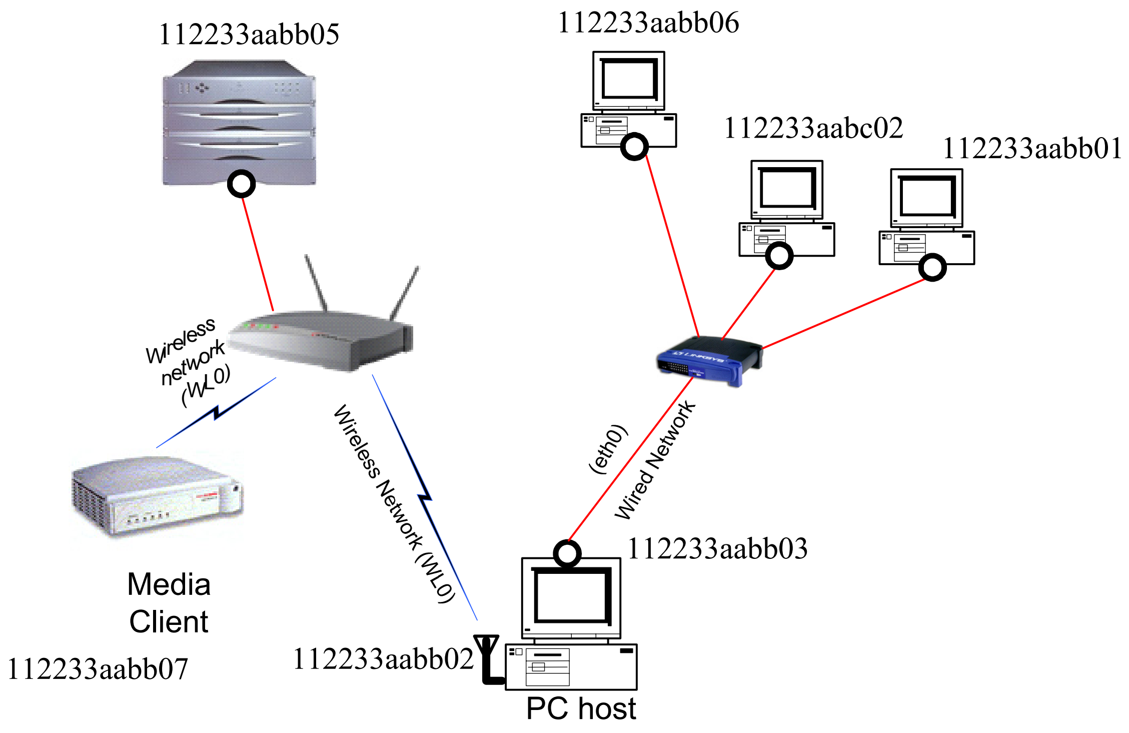

Figure 2-3 Example of a PC connected to an active network

```
<?xml version="1.0" encoding="UTF-8"?>
<RotameterInformation
  xmlns="http://www.upnp.org/schemas/RotameterInformation.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/RotameterInformation.xsd
  http://www.upnp.org/schemas/qos/RotameterInformation-v2.xsd">
  <LinkReachableMacs>
      <LinkId>eth0</LinkId>
      <MacAddress>112233aabb03</MacAddress>
      <RotameterObservation>
              <RotameterIndex>10000001</RotameterIndex>
              <ROAddr>112233aabb03</ROAddr>
              <ROBits>1000000</ROBits>
              <ROPeriod>1</ROPeriod>
              <ReportingDateTime>2004-11-26T15:03:23-08:00</ReportingDateTime>
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
      <RotameterObservation>
              <RotameterIndex>10000002</RotameterIndex>
              <ROAddr>112233aabb03</ROAddr>
              <ROBits>1000000</ROBits>
              <ROPeriod>1</ROPeriod>
              <ReportingDateTime>2004-11-26T15:04:23-08:00</ReportingDateTime>
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
      <RotameterObservation>
              <RotameterIndex>10000006</RotameterIndex>
              <ROAddr>112233aabb06</ROAddr>
              <ROBits>500000</ROBits>
              <ROPeriod>1</ROPeriod>
              <ReportingDateTime>2004-11-26T15:03:23-08:00</ReportingDateTime>
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
      <RotameterObservation>
              <RotameterIndex>10000007</RotameterIndex>
              <ROAddr>112233aabb06</ROAddr>
              <ROBits>500000</ROBits>
```

```
                <ROPeriod>1</ROPeriod>
                <ReportingDateTime>2004-11-26T15:04:23-08:00</ReportingDateTime>
                <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
        </RotameterObservation>
        <RotameterObservation>
                <RotameterIndex>10000011</RotameterIndex>
                <ROAddr>112233aabc02</ROAddr>
                <ROBits>500000</ROBits>
                <ROPeriod>1</ROPeriod>
                <ReportingDateTime>2004-11-26T15:03:43-08:00</ReportingDateTime>
                <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
        </RotameterObservation>
        <RotameterObservation>
                <RotameterIndex>10000012</RotameterIndex>
                <ROAddr>112233aabc02</ROAddr>
                <ROBits>500000</ROBits>
                <ROPeriod>1</ROPeriod>
                <ReportingDateTime>2004-11-26T15:04:43-08:00</ReportingDateTime>
                <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
        </RotameterObservation>
        <RotameterObservation>
                <RotameterIndex>10000013</RotameterIndex>
                <ROAddr>112233aabc02</ROAddr>
                <ROBits>500000</ROBits>
                <ROPeriod>1</ROPeriod>
                <ReportingDateTime>2004-11-26T15:05:43-08:00</ReportingDateTime>
                <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
        </RotameterObservation>
    </LinkReachableMacs>
    <LinkReachableMacs>
        <LinkId>WL0</LinkId>
        <MacAddress>112233aabb02</MacAddress>
        <RotameterObservation>
                <RotameterIndex>10000021</RotameterIndex>
                <ROAddr>112233aabb02</ROAddr>
                <ROBits>1000000</ROBits>
                <ROPeriod>1</ROPeriod>
                <ReportingDateTime>2004-11-26T15:03:23-08:00</ReportingDateTime>
                <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
        </RotameterObservation>
        <RotameterObservation>
                <RotameterIndex>10000022</RotameterIndex>
                <ROAddr>112233aabb02</ROAddr>
                <ROBits>1000000</ROBits>
                <ROPeriod>1</ROPeriod>
                <ReportingDateTime>2004-11-26T15:04:23-08:00</ReportingDateTime>
                <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
        </RotameterObservation>
        <RotameterObservation>
                <RotameterIndex>10000013</RotameterIndex>
                <ROAddr>112233aabb05</ROAddr>
                <ROBits>380000</ROBits>
                <ROPeriod>1</ROPeriod>
                <ReportingDateTime>2004-11-26T15:03:24-08:00</ReportingDateTime>
                <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
        </RotameterObservation>
        <RotameterObservation>
                <RotameterIndex>10000031</RotameterIndex>
                <ROAddr>112233aabb07</ROAddr>
                <ROBits>500000</ROBits>
                <ROPeriod>1</ROPeriod>
                <ReportingDateTime>2004-11-26T15:03:43-08:00</ReportingDateTime>
                <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
        </RotameterObservation>
        <RotameterObservation>
                <RotameterIndex>10000032</RotameterIndex>
                <ROAddr>112233aabb07</ROAddr>
                <ROBits>500000</ROBits>
                <ROPeriod>1</ROPeriod>
                <ReportingDateTime>2004-11-26T15:04:43-08:00</ReportingDateTime>
                <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
        </RotameterObservation>
        <RotameterObservation>
                <RotameterIndex>10000033</RotameterIndex>
                <ROAddr>112233aabb07</ROAddr>
```

```
              <ROBits>500000</ROBits>
              <ROPeriod>1</ROPeriod>
              <ReportingDateTime>2004-11-26T15:05:43-08:00</ReportingDateTime>
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
  </LinkReachableMacs>
</RotameterInformation>
```

2.3.11.4.Sample argument XML string – PC with two network interfaces that are both end point device with TrafficImportanceNumber reporting

Similar to the previous example this is an example of an end point network device with two actively connected network interfaces. In this example, one interface reports bits per interpreted TrafficImportanceNumber.

```
<?xml version="1.0" encoding="UTF-8"?>
<RotameterInformation
  xmlns="http://www.upnp.org/schemas/RotameterInformation.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/RotameterInformation.xsd
  http://www.upnp.org/schemas/qos/RotameterInformation-v2.xsd">
  <LinkReachableMacs>
      <LinkId>eth0</LinkId>
      <MacAddress>112233aabb03</MacAddress>
      <RotameterObservation>
              <RotameterIndex>10000001</RotameterIndex>
              <ROAddr>112233aabb03</ROAddr>
              <ROBits>1000000</ROBits>
              <ROPeriod>1</ROPeriod>
              <ReportingDateTime>2004-11-26T15:03:23-08:00</ReportingDateTime>
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
      <RotameterObservation>
              <RotameterIndex>10000002</RotameterIndex>
              <ROAddr>112233aabb03</ROAddr>
              <ROBits>1000000</ROBits>
              <ROPeriod>1</ROPeriod>
              <ReportingDateTime>2004-11-26T15:04:23-08:00</ReportingDateTime>
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
      <RotameterObservation>
              <RotameterIndex>10000006</RotameterIndex>
              <ROAddr>112233aabb06</ROAddr>
              <ROBits>500000</ROBits>
              <ROPeriod>1</ROPeriod>
              <ReportingDateTime>2004-11-26T15:03:23-08:00</ReportingDateTime>
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
      <RotameterObservation>
              <RotameterIndex>10000007</RotameterIndex>
              <ROAddr>112233aabb06</ROAddr>
              <ROBits>500000</ROBits>
              <ROPeriod>1</ROPeriod>
              <ReportingDateTime>2004-11-26T15:04:23-08:00</ReportingDateTime>
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
      <RotameterObservation>
              <RotameterIndex>10000011</RotameterIndex>
              <ROAddr>112233aabc02</ROAddr>
              <ROBits>500000</ROBits>
              <ROPeriod>1</ROPeriod>
              <ReportingDateTime>2004-11-26T15:03:43-08:00</ReportingDateTime>
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
      <RotameterObservation>
              <RotameterIndex>10000012</RotameterIndex>
              <ROAddr>112233aabc02</ROAddr>
              <ROBits>500000</ROBits>
              <ROPeriod>1</ROPeriod>
              <ReportingDateTime>2004-11-26T15:04:43-08:00</ReportingDateTime>
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
      <RotameterObservation>
```

```
                <RotameterIndex>10000013</RotameterIndex>
                <ROAddr>112233aabc02</ROAddr>
                <ROBits>500000</ROBits>
                <ROPeriod>1</ROPeriod>
                <ReportingDateTime>2004-11-26T15:05:43-08:00</ReportingDateTime>
                <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
        </RotameterObservation>
    </LinkReachableMacs>
    <LinkReachableMacs>
        <LinkId>WL0</LinkId>
        <MacAddress>112233aabb02</MacAddress>
        <RotameterObservation>
                <RotameterIndex>10000011</RotameterIndex>
                <ROAddr>112233aabb02</ROAddr>
                <ROBits0>780000</ROBits0>
                <ROBits5>200000</ROBits5>
                <ROBits7> 20000</ROBits7>
                <ROPeriod>1</ROPeriod>
                <ReportingDateTime>2004-11-26T15:03:23-08:00</ReportingDateTime>
                <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
        </RotameterObservation>
        <RotameterObservation>
                <RotameterIndex>10000013</RotameterIndex>
                <ROAddr>112233aabb05</ROAddr>
                <ROBits0>380000</ROBits0>
                <ROBits5>200000</ROBits5>
                <ROPeriod>1</ROPeriod>
                <ReportingDateTime>2004-11-26T15:03:24-08:00</ReportingDateTime>
                <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
        </RotameterObservation>
        <RotameterObservation>
                <RotameterIndex>10000015</RotameterIndex>
                <ROAddr>112233aabb07</ROAddr>
                <ROBits0>400000</ROBits0>
                <ROBits7> 20000</ROBits7>
                <ROPeriod>1</ROPeriod>
                <ReportingDateTime>2004-11-26T15:03:25-08:00</ReportingDateTime>
                <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
        </RotameterObservation>
      </LinkReachableMacs>
</RotameterInformation>
```

2.3.11.5.Sample argument XML string –Four port Ethernet Switch

This is an example of a layer-2 switching device that interconnects four physical Ethernet ports. The device supports layer-2 frame forwarding between all ports.

```
<?xml version="1.0" encoding="UTF-8"?>
<RotameterInformation
  xmlns="http://www.upnp.org/schemas/RotameterInformation.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/RotameterInformation.xsd
  http://www.upnp.org/schemas/qos/RotameterInformation-v2.xsd">
    <LinkReachableMacs>
        <LinkId>eth0</LinkId>
        <BridgeId>Bridge0</BridgeId>
        <RotameterObservation>
                <RotameterIndex>10000001</RotameterIndex>
                <ROAddr>112233aabb03</ROAddr>
                <ROBits>1000000</ROBits>
                <ROPeriod>1</ROPeriod>
                <ReportingDateTime>2004-11-26T15:03:23-08:00</ReportingDateTime>
                <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
        </RotameterObservation>
    </LinkReachableMacs>
    <LinkReachableMacs>
        <LinkId>eth1</LinkId>
        <BridgeId>Bridge0</BridgeId>
        <RotameterObservation>
                <RotameterIndex>10000004</RotameterIndex>
                <ROAddr>112233aabb06</ROAddr>
                <ROBits>1000000</ROBits>
                <ROPeriod>1</ROPeriod>
                <ReportingDateTime>2004-11-26T15:04:23-08:00</ReportingDateTime>
```

```
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
      <RotameterObservation>
              <RotameterIndex>10000007</RotameterIndex>
              <ROAddr>112233aabb01</ROAddr>
              <ROBits>1000000</ROBits>
              <ROPeriod>1</ROPeriod>
              <ReportingDateTime>2004-11-26T15:04:43-08:00</ReportingDateTime>
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
  </LinkReachableMacs>
  <LinkReachableMacs>
      <LinkId>eth2</LinkId>
      <BridgeId>Bridge0</BridgeId>
      <RotameterObservation>
              <RotameterIndex>10000017</RotameterIndex>
              <ROAddr>112233aabc02</ROAddr>
              <ROBits>2300000</ROBits>
              <ROPeriod>1</ROPeriod>
              <ReportingDateTime>2004-11-26T15:04:33-08:00</ReportingDateTime>
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
  </LinkReachableMacs>
  <LinkReachableMacs>
      <LinkId>eth3</LinkId>
      <BridgeId>Bridge0</BridgeId>
  </LinkReachableMacs>
</RotameterInformation>
```

2.3.11.6.Sample argument XML string – Wireless AP with one Ethernet Interface

This is an example of a wireless access point with three associated wireless stations and a single Ethernet port. The device supports layer-2 frame forwarding between all links. This includes forwarding (bridging) between wireless stations and to the Ethernet interface.

```
<?xml version="1.0" encoding="UTF-8"?>
<RotameterInformation
  xmlns="http://www.upnp.org/schemas/RotameterInformation.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/RotameterInformation.xsd
  http://www.upnp.org/schemas/qos/RotameterInformation-v2.xsd">
  <LinkReachableMacs>
      <LinkId>WL0</LinkId>
      <BridgeId>Bridge0</BridgeId>
      <RotameterObservation>
              <RotameterIndex>10001001</RotameterIndex>
              <ROAddr>112233aabb02</ROAddr>
              <ROBits>2000000</ROBits>
              <ROPeriod>1</ROPeriod>
              <ReportingDateTime>2004-11-26T15:04:43-08:00</ReportingDateTime>
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
      <RotameterObservation>
              <RotameterIndex>10001002</RotameterIndex>
              <ROAddr>112233aabb07</ROAddr>
              <ROBits>2300000</ROBits>
              <ROPeriod>1</ROPeriod>
              <ReportingDateTime>2004-11-26T15:04:33-08:00</ReportingDateTime>
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
  </LinkReachableMacs>
  <LinkReachableMacs>
      <LinkId>eth0</LinkId>
      <BridgeId>Bridge0</BridgeId>
      <RotameterObservation>
              <RotameterIndex>10001004</RotameterIndex>
              <ROAddr>112233aabb05</ROAddr>
              <ROBits>5800000</ROBits>
              <ROPeriod>1</ROPeriod>
              <ReportingDateTime>2004-11-26T15:03:34-08:00</ReportingDateTime>
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
      <RotameterObservation>
              <RotameterIndex>10001005</RotameterIndex>
```

```
                <ROAddr>112233aabb07</ROAddr>
                <ROBits>3700000</ROBits>
                <ROPeriod>1</ROPeriod>
                <ReportingDateTime>2004-11-26T15:04:34-08:00</ReportingDateTime>
                <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
        </RotameterObservation>
        <RotameterObservation>
                <RotameterIndex>10001006</RotameterIndex>
                <ROAddr>112233aabb05</ROAddr>
                <ROBits>6200000</ROBits>
                <ROPeriod>1</ROPeriod>
                <ReportingDateTime>2004-11-26T15:04:31-08:00</ReportingDateTime>
                <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
        </RotameterObservation>
    </LinkReachableMacs>
</RotameterInformation>
```

2.3.11.7.Sample argument XML string – Bridge device between Wireless station and Ethernet

This is an example of a bridging device with two interfaces on different network technologies. It does layer-2 forwarding of frames between wireless station interface and the wired Ethernet interface.

```
<?xml version="1.0" encoding="UTF-8"?>
<RotameterInformation
  xmlns="http://www.upnp.org/schemas/RotameterInformation.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/RotameterInformation.xsd
  http://www.upnp.org/schemas/qos/RotameterInformation-v2.xsd">
  <LinkReachableMacs>
      <LinkId>WL0</LinkId>
      <BridgeId>Bridge0</BridgeId>
      <RotameterObservation>
              <RotameterIndex>10001004</RotameterIndex>
              <ROAddr>112233aabb02</ROAddr>
              <ROBits>5800000</ROBits>
              <ROPeriod>1</ROPeriod>
              <ReportingDateTime>2004-11-26T15:03:34-08:00</ReportingDateTime>
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
  </LinkReachableMacs>
  <LinkReachableMacs>
      <LinkId>eth0</LinkId>
      <BridgeId>Bridge0</BridgeId>
      <RotameterObservation>
              <RotameterIndex>10001005</RotameterIndex>
              <ROAddr>112233aabb03</ROAddr>
              <ROBits>3700000</ROBits>
              <ROPeriod>1</ROPeriod>
              <ReportingDateTime>2004-11-26T15:04:34-08:00</ReportingDateTime>
              <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
      </RotameterObservation>
    </LinkReachableMacs>
</RotameterInformation>
```

2.3.12.A_ARG_TYPE_ConfRotameterObservations

A_ARG_TYPE_ConfRotameterObservations is a structure that configures how Rotameter observations are made.

This argument is described by the schema identified by "`http://www.upnp.org/schemas/ConfRotameterObservations.xsd`" and located at "`http://www.upnp.org/schemas/qos/ConfRotameterObservations-v2.xsd`".

2.3.12.1.Description of fields in ConfRotameterObservations structure

ROPeriod Duration of the Observation period. ROPeriod shall be less than or equal to the MonitoringResolutionPeriod. Units: seconds.

MonitorResolutionPeriod How often a Rotameter observation is initiated. Units: seconds.

Please see section 2.3.11.2 for a description of how these fields relate.

2.3.12.2.Sample argument XML string

```
<?xml version="1.0" encoding="UTF-8"?>
<ConfRotameterObservations
  xmlns="http://www.upnp.org/schemas/ConfRotameterObservations.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/ConfRotameterObservations.xsd
  http://www.upnp.org/schemas/qos/ConfRotameterObservations-v2.xsd">
        <ROPeriod>1</ROPeriod>
        <MonitorResolutionPeriod>60</MonitorResolutionPeriod>
</ConfRotameterObservations>
```

2.3.13.MostRecentStreamAction

MostRecentStreamAction is a structure that lists counters for QosDevice traffic stream actions, i.e. SetupTrafficQos and ReleaseTrafficQos. When the respective action is successfully invoked on the QosDevice, the counter is incremented. This state variable, when implemented, must be evented (to subscribing devices) to identify when QoS is setup for or removed from a traffic stream. This behavior can be meaningful for diagnostic purposes, e.g. identifying which source device started or stopped a QoS-enabled traffic stream that may be contending with an ongoing stream. Further queries may be done to gain relevant information about the stream, such as querying GetQosState or the GetRotameterInformation, or examine the TrafficDescriptor that identifies traffic stream and policy information. This information could be displayed to an end user interested in diagnosing a streaming problem.

This argument is described by the schema identified by
"`http://www.upnp.org/schemas/MostRecentStreamAction.xsd`" and located at
"`http://www.upnp.org/schemas/qos/MostRecentStreamAction-v2.xsd`".

2.3.13.1.Description of fields in the TrafficStreamUpdate structure

SetupTrafficQos: A non-negative integer value representing the number of successful invocations of SetupTrafficQos on the QosDevice.

ReleaseTrafficQos: A non-negative integer value representing the number of successful invocations of RemoveTrafficQos on the QosDevice.

2.3.13.2. Sample Argument XML String

```
<?xml version="1.0" encoding="UTF-8"?>
<MostRecentStreamAction
    xmlns="http://www.upnp.org/schemas/MostRecentStreamAction.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.upnp.org/schemas/MostRecentStreamAction.xsd
    http://www.upnp.org/schemas/qos/MostRecentStreamAction-v2.xsd" >
  <SetupTrafficQos>2</SetupTrafficQos>
  <ReleaseTrafficQos>1</ReleaseTrafficQos>
</MostRecentStreamAction>
```

2.3.14.A_ARG_TYPE_ MaxPossibleRotameterObservations

A_ARG_TYPE_MaxPossibleRotameterObservations is the maximum number of observations that the device is capable of providing.

2.3.14.1.Description of MaxPossibleRotameterObservations structure

The MaxPossibleRotameterObservation is ui4 variable and represents the maximum number of observations the device is capable of observing.

2.3.15.Relationships between State Variables

2.4. Eventing and Moderation

Table 2-2: Event Moderation

Variable Name	Evente d	Moderated Event	Max Event Rate[1]	Logical Combination	Min Delta per Event[2]
PathInformation	Yes	Yes	2	NA	NA
MostRecentStreamAction	Yes	Yes	2	NA	NA

[1] Determined by N, where Rate = (Event)/(N secs).
[2] (N) * (allowedValueRange Step).

2.4.1. Event Model

PathInformation: The state variable PathInformation is optional, but must be evented when implemented.

When there is a change in PathInformation, the QosDevice will issue an event and send the updated PathInformation variable in the body of the event. This event is moderated to avoid flooding the network with repeated events.

MostRecentStreamAction: The MostRecentStreamAction state variable is optional, but must be evented when implemented

Any time a **SetupTrafficQos** or **ReleaseTrafficQos** action is invoked successfully, the QoS device will issue an event and send the updated MostRecentStreamAction variable in the body of the event. This event is moderated to avoid flooding the network with repeated events.

2.5. Actions

Immediately following this table is detailed information about these actions, including short descriptions of the actions, the effects of the actions on state variables, and error codes defined by the actions.

Table 2-3: Actions

Name	Req. or Opt. [1]
GetQosDeviceCapabilities	R
GetQosState	R
SetupTrafficQos	R
ReleaseTrafficQos	R
GetPathInformation	O
GetQosDeviceInfo	O
GetRotameterInformation	O
ConfigureRotameterObservation	O

[1] R = Required, O = Optional, X = Non-standard.

2.5.1. GetQosDeviceCapabilities

This action returns the static QoS capabilities of the QosDevice.

2.5.1.1. Service requirements

None.

2.5.1.2. Control Point requirements when calling the action

None.

2.5.1.3. Arguments

Table 2-4: Arguments for GetQosDeviceCapabilities

Argument	Direction	relatedStateVariable
QosDeviceCapabilities	Out	A_ARG_TYPE_QosDeviceCapabilities

2.5.1.4. Dependency on State (if any)

None, these are static capabilities.

2.5.1.5. Effect on State (if any)

None.

2.5.1.6. Errors

Refer to UPnP Device architecture for common error codes.

Table 2-5: Error Codes for GetQosDeviceCapabilities

errorCode	errorDescription	Description

2.5.2. GetQosState

This action returns the instantaneous QoS state of the device. The device must list the TrafficDescriptor(s) in the ListOfTrafficDescriptors argument that were registered in the device by QoS Management Entities.

2.5.2.1. Service requirements

None.

2.5.2.2. Control Point requirements when calling the action

None.

2.5.2.3. Arguments

Table 2-6: Arguments for GetQosState

Argument	Direction	relatedStateVariable
QosDeviceState	Out	A_ARG_TYPE_QosDeviceState
NumberOfTrafficDescriptors	Out	A_ARG_TYPE_NumTrafficDescriptors
ListOfTrafficDescriptors	Out	A_ARG_TYPE_TrafficDescriptorsPerInterface

2.5.2.4. Dependency on State (if any)

This action does not have any dependency on the state of QosDevice service.

2.5.2.5. Effect on State (if any)

This action does not have any effect on the state of QosDevice service.

2.5.2.6. Errors

Table 2-7: Error Codes for GetQosState

errorCode	errorDescription	Description

2.5.3. SetupTrafficQos

SetupTrafficQoS interface indicates to the device to setup QoS for the Traffic described by A_ARG_TYPE_TrafficDescriptor.

Please refer to the [QM] document Appendix A 'Traffic Descriptor Matrix' for information about all of the fields of the TrafficDescriptor and how they are used.

Typically, the QoS Management Entity calls this action only once per traffic handle registration. If the QoS Management Entity intends to update QoS associated with the traffic (e.g. the lease time of the traffic), then it has to go over the complete traffic setup process again after it has released the QoS.

2.5.3.1. Service requirements

If there is no TrafficDescriptor registered in the QosDevice with the same A_ARG_TYPE_TrafficHandle, then this TrafficDescriptor will be registered in the QosDevice after the successful execution of this action.

If the device already has the TrafficDescriptor (identified by the TrafficHandle) registered, then the QosDevice must return an error 702.

If the QosDevice does not receive a TrafficDescriptor with a TrafficImportanceNumber, the QosDevice must return error 711. A TrafficImportanceNumber with NULL value is improper (being that an integer cannot be NULL), and thus the QosDevice must return error 716.

If the QosDevice does not receive a TrafficDescriptor with ActiveTspecIndex, it must return error 711.

If the QosDevice does not receive a TrafficDescriptor with a TrafficHandle, or TrafficHandle has a NULL value, it must return error 700.

In the TrafficDescriptor to the QosDevice, the Tspec for which TrafficPolicy is provided is indicated by the ActiveTspecIndex. ActiveTspecIndex must be one of the TspecIndex values in the AvailableOrderedTspecList. If not, QosDevice must return the error 720.

2.5.3.2. Control Point requirements when calling the action

A Control Point (QoS Management Entity) must supply the TrafficImportanceNumber in TrafficDescriptor to QosDevice when calling the **SetupTrafficQos** action.

A Control Point (QoS Management Entity) must supply the ActiveTspecIndex in TrafficDescriptor to QosDevice when calling the **SetupTrafficQos** action.

A Control Point (QoS Management Entity) must supply the TrafficHandle in TrafficDescriptor to QosDevice when calling the **SetupTrafficQos** action.

A Control Point (QoS Management Entity) must supply an ActiveTspecIndex that is one of the TspecIndex values in the AvailableOrderedTspecList in TrafficDescriptor to QosDevice when calling the **SetupTrafficQos** action.

2.5.3.3. Arguments

Table 2-8: Arguments for SetupTrafficQos

Argument	Direction	relatedStateVariable
SetupTrafficDescriptor	In	A_ARG_TYPE_TrafficDescriptor
QosStateId	In	A_ARG_TYPE_QosStateId

2.5.3.4. Dependency on State (if any)

QosStateId is provided as an input to this action. In case the current QosStateId of the device is different than the one specified by the Control Point (QoS Management Entity), the action returns the error 760. Otherwise, the QosDevice sets up QoS for the traffic stream.

2.5.3.5. Effect on State (if any)

Upon successful completion of this action, the QosDevice sets up QoS for the traffic specified in the action request. Please refer to 'Theory of Operation' section for more details.

The QosDevice service must not modify any of the the TrafficPolicy elements (such as TrafficImportanceNumber, UserImportanceNumber etc.) assigned by the QoS Management Entity in the TrafficDescriptor structure. The QosDevice must not modify any of the OptionalPolicyParams specified in the TrafficDescriptor by the QoS Management Entity.

Upon successful completion of SetupTrafficQos, source devices implementing the QosDevice service must prioritize the traffic, associated with the TrafficId, according to the TrafficImportanceNumber (hence PacketTaggingSupported="Yes") on their output interfaces. Intermediate devices implementing the QosDevice service with PacketTaggingSupported="Yes" must prioritize the traffic associated with the TrafficId according to the TrafficImportanceNumber on their output interfaces irrespective of incoming traffic priority.

2.5.3.6. Errors

Table 2-9: Error Codes for SetupTrafficQos

errorCode	errorDescription	Description
700	Traffic Handle missing or empty	Traffic Handle must be filled in as input to this action.

errorCode	errorDescription	Description
702	Traffic Handle already registered	A Control Point (QoS Management Entity) is not allowed to setup or modify QoS using **SetupTrafficQos** if QoS has already been setup for that handle.
710	Incomplete TrafficId	All TrafficId fields (SourceAddress, DestinationAddress, SourcePort, DestinationPort and Protocol) must be present.
711	Insufficient information	The input information is not complete.
716	An input parameter (e.g. TrafficDescriptor) does not validate against the XML schema	One of the XML-based input arguments does not follow the schema
720	ActiveTspecIndex is not a TspecIndex	
751	Device not on path	
760	QosStateId does not match	Please refer to the 'Theory of Operation' section.
761	QosDevice cannot setup this stream	QoS Setup failed, e.g device does not support prioritized QoS

2.5.4. ReleaseTrafficQos

The **ReleaseTrafficQos** indicates that the traffic stream is no longer managed by UPnP QoS at this device. **ReleaseTrafficQoS** provides an indication to the device to release the QoS for the traffic identified by A_ARG_TYPE_TrafficHandle. After this call, traffic handle is no longer registered at the device to provide QoS.

2.5.4.1. Service requirements

The QosDevice must return an error code 703 if the input TrafficHandle is not a valid. An input TrafficHandle is valid only if it is part of one and only one of the TrafficDescriptors stored in that device.

2.5.4.2. Control Point requirements when calling the action

Control Point must supply a valid traffic handle to revoke the QoS of the traffic stream.

2.5.4.3. Arguments

Table 2-10: Arguments for ReleaseTrafficQos

Argument	Direction	relatedStateVariable
ReleaseTrafficHandle	In	A_ARG_TYPE_TrafficHandle

2.5.4.4. Dependency on State (if any)

The TrafficHandle provided has to be valid.

2.5.4.5. Effect on State (if any)

After this call, traffic handle is no longer registered at the device to provide QoS. The device must release all its QoS resources allocated to that traffic.

2.5.4.6. Errors

Table 2-11: Error Codes for ReleaseTrafficQos

errorCode	errorDescription	Description
703	Traffic Handle unknown to this device	

2.5.5. GetPathInformation

This is an optional action. When supported, this action call returns the 'PathInformation' structure for that QosDevice service providing information about the reachable MACs. This information may be used by the Control Point (QoS Management Entity) for path determination.

2.5.5.1. Service requirements

None.

2.5.5.2. Control Point requirements when calling the action

None.

2.5.5.3. Arguments

Table 2-12: Arguments for GetPathInformation

Argument	Direction	relatedStateVariable
PathInformation	Out	PathInformation

2.5.5.4. Dependency on State (if any)

None.

2.5.5.5. Effect on State (if any)

None.

2.5.5.6. Errors

Table 2-13: Error Codes for GetPathInformation

errorCode	errorDescription	Description

2.5.6. GetQosDeviceInfo

This is an optional action. When supported, this action call returns the 'QosDeviceInfo' structure for that QosDevice service providing information about the port number and protocol information associated with the provided TrafficDescriptor. The device may be able to determine this information based on the following:

- Available elements of the TrafficId
- AvTransportUri and AvTransportInstanceId if specified
- MediaServerConnectionId and MediaRendererConnectionId if specified

QosDeviceInfo returned as part of this action may be used by the Control Point (QoS Management Entity) to complete the traffic identifier structure.

2.5.6.1. Service requirements

If the QosDevice receives information which is insufficient to determine the port numbers and proptocol the the QosDevice will return error 712.

2.5.6.2. Control Point requirements when calling the action

Control Point (QoS Management Entity) should supply all available information related to the UPnP AV scenario such as MediaServerConnectionId, MediaRendererConnectionId, AvTransportUri or AvTransportInstanceId.

2.5.6.3. Arguments

Table 2-14: Arguments for GetQosDeviceInfo

Argument	Direction	relatedStateVariable
TrafficDescriptor	In	A_ARG_TYPE_TrafficDescriptor
QosDeviceInfo	Out	A_ARG_TYPE_QosDeviceInfo

2.5.6.4. Dependency on State (if any)

None.

2.5.6.5. Effect on State (if any)

None.

2.5.6.6. Errors

Table 2-15: Error Codes for GetQosDeviceInfo

errorCode	errorDescription	Description
712	Incomplete information to determine protocol and port numbers	Incomplete information. For example, in case of the UPnP AV scenario, MediaServerConnectionId, MediaRendererConnectionId, AvTransportUri or AvTransportInstanceId is required but not provided.

2.5.7. GetRotameterInformation

This is an optional action. When supported, this action call returns the 'RotameterInformation' structure for that QosDevice service providing information about the reachable MACs and Rotameter information. This information may be used directly by any Control Point to observe traffic flow. The QosDevice provides the most recent (in time) observations indicated by the number A_ARG_TYPE_NumRotameterObservations.

2.5.7.1. Service requirements

A QosDevice must be first configured using the action **ConfigureRotameterObservation** before accessing the RotameterObservation using the **GetRotameterInformation** action. If this sequence is not followed then the QosDevice will return the error 735.

2.5.7.2. Control Point requirements when calling the action

A Control Point must first configure the QosDevice by calling the action **ConfigureRotameterObservation** before accessing the RotameterObservation.

2.5.7.3. Arguments

Table 2-16: Arguments for GetRotameterInformation

Argument	Direction	relatedStateVariable
RequestedNumRotameterObservations	In	A_ARG_TYPE_NumRotameterObservations
RotameterObservation	Out	A_ARG_TYPE_RotameterInformation

2.5.7.4. Dependency on State (if any)

RequestedNumRotameterObservations is provided as an input to this action to indicate the number of Rotameter Observations per MAC address requested by a Control Point. The response to GetRotameterInformation provides the most recent (in time) observations indicated by the number RequestedNumRotameterObservations. In case the RequestedNumRotameterObservations is more than the Rotameter service is capable of providing, the action must return error code 732. In case the RequestedNumRotameterObservations is more than the Rotameter service currently has at this time, the action must return error code 733. Otherwise, the QosDevice returns RotameterObservation.

2.5.7.5. Effect on State (if any)

None

2.5.7.6. Errors

Table 2-17: Error Codes for GetRotameterInformation

errorCode	errorDescription	Description
732	Requested too many observations	RequestedNumRotameterObservations is more than device capabilities
735	ConfigureRotameterObservation has not been invoked	ConfigureRotameterObservation has not been invoked before calling GetRotameterObservation

errorCode	errorDescription	Description
733	No valid observation	Unable to provide an observation at this time.

2.5.8. ConfigureRotameterObservation

2.5.8.1. Arguments

Table 2-18: Arguments for ConfigureRotameterObservation

Argument	Direction	relatedStateVariable
RequestedConfRotameterObservations	In	A_ARG_TYPE_ConfRotameterObservations
MaxPossibleRotameterObservations	Out	A_ARG_TYPE_MaxPossibleRotameterObservations

2.5.8.2. Dependency on State (if any)

RequestedConfRotameterObservations is provided as an input to this action. In case the RequestedConfRotameterObservations is more than the capabilities of the device, the action returns an error (Error Code 730 or 731, described below). If ROPeriod is greater than MonitorResolutionPeriod, the action returns error code 734. If no error is returned, observations must begin immediately, i.e. the device must not wait until a Control Point requests a Rotameter observation.

Because the Rotameter service is purposed at providing diagnostic value (i.e. which device is sending or receiving the most traffic on the network), and most applications do not have steady traffic characteristics, it is recommended that ROPeriod and MonitorResolutionPeriod are the same value with sufficiently small granularity (1 second for example).

Upon successful configuration of the QosDevice Rotameter function, the maximum number of observations the device is capable of must be returned, i.e. MaxPossibleRotameterObservations. This allows a Control Point to know the maximum number of observations to request when calling GetRotameterInformation.

2.5.8.3. Service requirements

None.

2.5.8.4. Control Point requirements when calling the action

None.

2.5.8.5. Effect on State (if any)

A Control Point (QoS Management Entity) calling action **GetRotameterInformation** should ensure that a QosDevice has been configured with **ConfigureRotameterObservation**.

2.5.8.6. Errors

Table 2-19: Error Codes for ConfigureRotameterObservation

errorCode	errorDescription	Description
730	ROPeriod incapable	ROPeriod is outside the capabilities of the device.
731	MonitorResolution Period incapable	MonitorResolutionPeriod is outside the capabilities of the device.
734	Invalid Arguments	ROPeriod > MonitorResolutionPeriod

2.5.9. Non-Standard Actions Implemented by a UPnP Vendor

To facilitate certification, non-standard actions implemented by UPnP vendors should be included in this service template. The UPnP Device Architecture lists naming requirements for non-standard actions (see the section on Description).

2.5.10.Relationships Between Actions

2.5.11.Common Error Codes

The following table lists error codes common to actions for this service type. If an action results in multiple errors, the most specific error must be returned. These common error codes are defined in the UPnP™ Device Architecture [DEVICE] and other Technical Committee documents.

Table 2-20: Common Error Codes

errorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnPDevice Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
700	Traffic Handle missing or empty	Traffic Handle must be filled in as input to this action.
702	Traffic Handle already registered	A Control Point (QoS Management Entity) is not allowed to setup or modify QoS using SetupTrafficQos if QoS has already been setup for that handle.
703	Traffic Handle unknown to this device	
710	Incomplete TrafficId	All TrafficId fields (Source Ip, Destination IP, Source Port, Destination Port and Protocol) must be present.
711	Insufficient information	The input information is not complete.
712	Incomplete information to determine protocol and port numbers	Incomplete information. For example, in case of the UPnP AV scenario, MediaServerConnectionId, MediaRendererConnectionId, AvTransportUri or AvTransportInstanceId is required but not provided.

errorCode	errorDescription	Description
716	An input parameter (e.g. TrafficDescriptor) does not validate against the XML schema	One of the XML-based input arguments does not follow the schema
720	ActiveTspecIndex is not a TspecIndex	
730	ROPeriod incapable	ROPeriod is outside the capabilities of the device.
731	MonitorResolution Period incapable	MonitorResolutionPeriod is outside the capabilities of the device.
732	Requested too many observations	RequestedNumRotameterObservations is more than device capabilities.
733	No valid observation	Unable to provide an observation at this time.
734	Invalid arguments	ROPeriod cannot be greater than MonitorResolutionPeriod
751	Device not on path	
760	QosStateId does not match	Please refer to the 'Theory of Operation' section.
735	ConfigureRotameterObservation has not been invoked	ConfigureRotameterObservation has not been invoked before calling GetRotameterObservation
761	QosDevice cannot setup this stream	QoS Setup failed, e.g. device does not support prioritized QoS
800-899	TBD	(Specified by UPnP™ vendor.)

2.6. Theory of Operation

The QosDevice Service provides an action for Control Points to query the QoS state of the device, execute actions on the device and register for events the device generates. Typically the QoS Management Entity interacts with the QosDevice Service.

A UPnP QosDevice will expose its static QoS capabilities through the **GetQosDeviceCapabilities** action. The static capabilities include type of native QoS support, such as "Prioritized" or "BestEffort". Other capabilities include maximum PHY rate. Although device level admission control is not currently supported, there is an element 'AdmissionControlSupported' which is expected to always return the value of 'No'.

The run time QoS state of the device may be very different from what was advertised through the **GetQosDeviceCapabilities** and this state is exposed through the **GetQosState** action. The **GetQosState** action provides information about the currently active traffic streams on the device using TrafficDescriptor structures.

The **SetupTrafficQos** action provides a mechanism for the QoS Management Entity to request the device to provide network resources for a traffic stream identified by the TrafficDescriptor.

Race conditions may occur when different QoS Management Entities use the actions **GetQosState** and **SetupTrafficQos**. To identify such conditions, the QosDevice provides a unique identification of its state named QosStateId in reply to the action **GetQosState**. The QoS Management Entity provides QosStateId as input argument to **SetupTrafficQos**. In case the QosStateId sent by the QoS Management Entity does not match the most recent QosStateId handed out by the device, the device responds to **SetupTrafficQoS** with error code 760.

The QoS Management Entity can indicate to the QosDevice to release resources in two ways. The QoS Management Entity may either call the **ReleaseTrafficQoS** action or simply not renew the lease time associated with the traffic stream.

The QosDevice provides layer 2 reachability information to the QoS Management Entity through the **GetPathInformation** action. This information includes the link identifier, MAC address, bridging information and reachable MAC addresses for each link on the QosDevice. This is useful to help a QoS Management Entity determine the topology of the network and the path of the traffic streams.

A QosDevice optionally collects information about network flows (Rotameter) from its interfaces. The QosDevice can be queried for the information using the action **GetRotameterInformation**. The collection of data may be controlled using the action **ConfigureRotameterObservation**.

A QoS Management Entity may provide a lease time as part of a TrafficDescriptor as input to **SetupTrafficQos**. When the lease time expires, the QosDevice releases the QoS resources allocated to that TrafficDescriptor. The state of the QosDevice must be the same as if it were a ReleaseTrafficQos action. In case a QoS Management Entity does not specify the lease time, the QoS resources remain allocated until they are released by a QoS Management Entity.

The persistence of any QosDevice information across reboots is not required.

3. XML Service Descriptions

```
<?xml version="1.0" encoding="utf-8"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">

   <specVersion>
      <major>1</major>
      <minor>0</minor>
   </specVersion>

<actionList>
   <action>
      <name>GetPathInformation</name>
      <argumentList>
         <argument>
            <name>PathInformation</name>
            <direction>out</direction>
            <relatedStateVariable>PathInformation</relatedStateVariable>
         </argument>
      </argumentList>
   </action>
   <action>
      <name>GetRotameterInformation</name>
      <argumentList>
         <argument>
            <name>RequestedNumRotameterObservations</name>
            <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_NumRotameterObservations</relatedStateVariable>
         </argument>
         <argument>
            <name>RotameterObservation</name>
            <direction>out</direction>
            <relatedStateVariable>A_ARG_TYPE_RotameterInformation</relatedStateVariable>
         </argument>
      </argumentList>
   </action>
   <action>
      <name>ConfigureRotameterObservation</name>
      <argumentList>
         <argument>
            <name>RequestedConfRotameterObservations</name>
            <direction>in</direction>
<relatedStateVariable>A_ARG_TYPE_ConfRotameterObservations</relatedStateVariable>
         </argument>
         <argument>
            <name>MaxPossibleRotameterObservations</name>
            <direction>out</direction>
 <relatedStateVariable>A_ARG_TYPE_MaxPossibleRotameterObservations</relatedStateVariable>
         </argument>
      </argumentList>
   </action>
   <action>
      <name>GetQosDeviceCapabilities</name>
      <argumentList>
         <argument>
            <name>QosDeviceCapabilities</name>
            <direction>out</direction>
            <relatedStateVariable>A_ARG_TYPE_QosDeviceCapabilities</relatedStateVariable>
         </argument>
      </argumentList>
   </action>
   <action>
      <name>GetQosDeviceInfo</name>
      <argumentList>
         <argument>
            <name>SetupTrafficDescriptor</name>
            <direction>in</direction>
            <relatedStateVariable>A_ARG_TYPE_TrafficDescriptor</relatedStateVariable>
         </argument>
         <argument>
            <name>QosDeviceInfo</name>
            <direction>out</direction>
            <relatedStateVariable>A_ARG_TYPE_QosDeviceInfo</relatedStateVariable>
         </argument>
      </argumentList>
   </action>
   <action>
      <name>GetQosState</name>
      <argumentList>
         <argument>
            <name>QosDeviceState</name>
            <direction>out</direction>
            <relatedStateVariable>A_ARG_TYPE_QosDeviceState</relatedStateVariable>
         </argument>
         <argument>
            <name>NumberOfTrafficDescriptors</name>
            <direction>out</direction>
            <relatedStateVariable>A_ARG_TYPE_NumTrafficDescriptors</relatedStateVariable>
         </argument>
         <argument>
            <name>ListOfTrafficDescriptors</name>
            <direction>out</direction>

<relatedStateVariable>A_ARG_TYPE_TrafficDescriptorsPerInterface</relatedStateVariable>
         </argument>
      </argumentList>
   </action>
   <action>
      <name>ReleaseTrafficQos</name>
      <argumentList>
         <argument>
            <name>ReleaseTrafficHandle</name>
            <direction>in</direction>
            <relatedStateVariable>A_ARG_TYPE_TrafficHandle</relatedStateVariable>
         </argument>
      </argumentList>
   </action>
   <action>
      <name>SetupTrafficQos</name>
      <argumentList>
         <argument>
            <name>SetupTrafficDescriptor</name>
            <direction>in</direction>
            <relatedStateVariable>A_ARG_TYPE_TrafficDescriptor</relatedStateVariable>
         </argument>
         <argument>
            <name>QosStateId</name>
            <direction>in</direction>
            <relatedStateVariable>A_ARG_TYPE_QosStateId</relatedStateVariable>
         </argument>
      </argumentList>
   </action>
</actionList>

<serviceStateTable>
   <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_TrafficHandle</name>
      <dataType>string</dataType>
   </stateVariable>
   <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_QosStateId</name>
      <dataType>string</dataType>
   </stateVariable>
   <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_QosDeviceState</name>
      <dataType>string</dataType>
   </stateVariable>
   <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_NumTrafficDescriptors</name>
      <dataType>ui4</dataType>
   </stateVariable>
   <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_QosDeviceInfo</name>
      <dataType>string</dataType>
   </stateVariable>
   <stateVariable sendEvents="yes">
      <name>PathInformation</name>
      <dataType>string</dataType>
   </stateVariable>
   <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_QosDeviceCapabilities</name>
      <dataType>string</dataType>
   </stateVariable>
   <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_TrafficDescriptorsPerInterface</name>
      <dataType>string</dataType>
   </stateVariable>
   <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_TrafficDescriptor</name>
      <dataType>string</dataType>
   </stateVariable>
   <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_NumRotameterObservations</name>
      <dataType>ui4</dataType>
      <defaultValue>1</defaultValue>
   </stateVariable>
   <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_RotameterInformation</name>
      <dataType>string</dataType>
   </stateVariable>
   <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_ConfRotameterObservations</name>
      <dataType>string</dataType>
   </stateVariable>
   <stateVariable sendEvents="yes">
      <name>MostRecentStreamAction</name>
      <dataType>string</dataType>
   </stateVariable>
   <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_MaxPossibleRotameterObservations</name>
      <dataType>ui4</dataType>
   </stateVariable>
</serviceStateTable>
</scpd>
```

4. Test

No semantic tests have been specified for this service.