QosManager:2 Erratum

Erratum Number:	***Next sequential erratum number***	**Effective Date:**	***July 31, 2008***
Document and Version:	***Document erratum applies to QosManager:2***		
Cross References:	***List other Erratum's or Documents that this change may apply to or have associated changes with***		

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.**

QosManager:2 Erratum

Entry	*1*
Section	***2.5 Theory of Operation***
Page	*31*

Background

Corrected text to cover the case where AdmissionControlSupported = 'Yes'

Current Text

2.5. Theory of Operation

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. The 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 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 QoS Management Entity decides that there are insufficient resource, it may try another Tspec listed in the TrafficDescriptor. If QoS Management 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 ~~for which AdmissionControlSupported equals 'No'~~. If **QD: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.

New Text

2.5. Theory of Operation

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. The 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 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 QoS Management Entity decides that there are insufficient resources, it may try another Tspec listed in the TrafficDescriptor. If QoS Management 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. If **QD: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.