Information technology - UPnP Device Architecture - Part 17-10: Quality of Service Device Control Protocol - Level 3 - Quality of Service Device Service

Contents

Page

1 Overview and Scope....5
1.1 Referenced Specifications....5
1.1.1 Normative References....5
1.1.2 Informative References....6
2 Service Modeling Definitions....6
2.1 ServiceType....6
2.2 State Variables....7
2.2.1 XML Fragments as UPnP Arguments....7
2.2.2 A_ARG_TYPE_TrafficDescriptor....8
2.2.3 A_ARG_TYPE_TrafficDescriptorsPerInterface....8
2.2.4 A_ARG_TYPE_TrafficHandle....10
2.2.5 A_ARG_TYPE_NumTrafficDescriptors....10
2.2.6 A_ARG_TYPE_QosDeviceCapabilities....10
2.2.7 A_ARG_TYPE_QosDeviceState....11
2.2.8 PathInformation....12
2.2.9 A_ARG_TYPE_QosDeviceInfo....14
2.2.10 A_ARG_TYPE_QosStateId....14
2.2.11 A_ARG_TYPE_NumRotameterObservations....14
2.2.12 A_ARG_TYPE_RotameterInformation....15
2.2.13 A_ARG_TYPE_ConfRotameterObservations....20
2.2.14 MostRecentStreamAction....21
2.2.15 A_ARG_TYPE_MaxPossibleRotameterObservations....22
2.2.16 A_ARG_TYPE_Resource....22
2.2.17 A_ARG_TYPE_AdmitTrafficQosExtendedResult....23
2.2.18 A_ARG_TYPE_ListOfAdmittedTraffic....26
2.2.19 A_ARG_TYPE_ PreferredQph....28
2.2.20 UnexpectedStreamChange....29
2.2.21 A_ARG_TYPE_PreemptingTrafficInfo....29
2.2.22 A_ARG_TYPE_ListOfMostRecentUnexpectedStreamChanges....30
2.2.23 A_ARG_TYPE_QosDeviceExtendedState....33
2.2.24 A_ARG_TYPE_Layer2Mapping....38
2.2.25 A_ARG_TYPE_AdmitTrafficQosSucceeded....38
2.2.26 A_ARG_TYPE_TrafficDescriptorsWanted....38
2.2.27 A_ARG_TYPE_SetPreferredQphResults....38
2.2.28 A_ARG_TYPE_ NumberOfUnexpectedStreamChangesRequested....39
2.2.29 A_ARG_TYPE_ NumberOfUnexpectedStreamChangesReported....39
2.2.30 A_ARG_TYPE_NewTrafficLeaseTime....39
2.2.31 A_ARG_TYPE_TrafficDescriptorContainer....39
2.2.32 A_ARG_TYPE_Layer2MappingContainer....41
2.2.33 A_ARG_TYPE_QosDeviceInfoContainer....41
2.3 Eventing and Moderation....43
2.3.1 Event Model....43
2.4 Actions....44
2.4.1 GetQosDeviceCapabilities()....45
2.4.2 GetQosState()....46

2.4.3 SetupTrafficQos() 47
2.4.4 ReleaseTrafficQos() 49
2.4.5 GetPathInformation 50
2.4.6 GetQosDeviceInfo() 51
2.4.7 ConfigureRotameterObservation() 52
2.4.8 GetRotameterInformation() 53
2.4.9 AdmitTrafficQos() 54
2.4.10 UpdateAdmittedQos() 62
2.4.11 ReleaseAdmittedQos() 65
2.4.12 GetExtendedQosState() 67
2.4.13 SetPreferredQph() 68
2.4.14 GetUnexpectedStreamChanges() 70
2.4.15 VerifyTrafficHandle() 71
2.4.16 UpdateTrafficLeaseTime() 71
2.4.17 SetL2Map() 72
2.4.18 Non-Standard Actions Implemented by a UPnP Vendor 73
2.4.19 Error Code Summary 73
2.4.20 Reason Code Summary 74
2.5 Theory of Operation (Informative) 75
2.5.1 Parameterized QoS 77
2.5.2 Prioritized QoS 80
2.5.3 Hybrid QoS 81
3 XML Service Descriptions 82
4 Test 88
Annex A (informative) Additional Examples for State Variables 89
A.1 Additional *PathInformation* Examples 89
A.1.1 Sample argument XML string – PC with two network interfaces that are both end point device and bridged 89
A.1.2 Sample argument XML string –Four port Ethernet Switch 89
A.1.3 Sample argument XML string – Wireless AP with one Ethernet Interface 90
A.1.4 Sample argument XML string – Bridge device between Wireless station and Ethernet 90
A.2 Additional A_ARG_TYPE_RotameterInformation Examples 91
A.2.1 Sample argument XML string – PC with two network interfaces that are both end point devices 91
A.2.2 Sample argument XML string – PC with two network interfaces that are both end point device with TrafficImportanceNumber reporting 94
A.2.3 Sample argument XML string –Four port Ethernet Switch 95
A.2.4 Sample argument XML string – Wireless AP with one Ethernet Interface 95
A.2.5 Sample argument XML string – Bridge device between Wireless station and Ethernet 96
Annex B (normative) Template for Requirements on the QosDevice Service implementation that are specific for the underlying Network Technologies 98
B.1 <Technology Name> 98
B.1.1 References 98
B.1.2 Priority Mapping 98
B.1.3 *QosSegmentId* formation 98
B.1.4 *Layer2StreamId* representation 99

B.1.5 Mapping of UPnP-QoS Parameters to *<technology>* Parameters 99
B.1.6 Blocking traffic stream identification 100
B.1.7 Responsibility for QoS Setup 100
B.1.8 Mapping of *<technology>* Returned Parameters to *ProtoTspec* Parameters 101
B.1.9 Mapping of *<technology>* Returned Parameters to *AdmitTrafficQosExtendedResult and AllocatedResources* Parameters 102

Figure 2-1 — Relationship between ROPeriod and MonitorResolutionPeriod 16
Figure 2-2 — PC with Two Network Interfaces 18
Figure 2-3 — Example of a PC connected to an active network 19
Figure 2-4 — Relationship between End-to-End Delay and QoS Segment Delay 57
Figure 2-5 — Relationship between QoS Segment Delay And MaxCommittedDelay. 58
Figure 2-6 — Components of *MaxCommittedDelay* 59
Figure 2-7 — Containers and How They Nest 78
Figure A.1 — Example of a PC connected to an active network 91

Table 2-1 — State Variables 7
Table 2-2 — Reason Codes For AdmissionStatusNet 24
Table 2-3 — Reason Codes For AdmissionStatusDev 25
Table 2-4 — Containers In Which A Parameter Can Appear 34
Table 2-5 — Reason Codes For *A_ARG_TYPE_SetPreferredQphResults* 39
Table 2-6 — Event Moderation 43
Table 2-7 — Actions 45
Table 2-8 — Arguments for *GetQosDeviceCapabilities()* 45
Table 2-9 — Error Codes for *GetQosDeviceCapabilities()* 46
Table 2-10 — Arguments for *GetQosState()* 46
Table 2-11 — Error Codes for *GetQosState()* 47
Table 2-12 — Arguments for *SetupTrafficQos()* 47
Table 2-13 — Error Codes for *SetupTrafficQos()* 49
Table 2-14 — Arguments for *ReleaseTrafficQos()* 49
Table 2-15 — Error Codes for *ReleaseTrafficQos()* 50
Table 2-16 — Arguments for *GetPathInformation()* 50
Table 2-17 — Error Codes for Get*PathInformation* 50
Table 2-18 — Arguments for *GetQosDeviceInfo()* 51
Table 2-19 — Error Codes for *GetQosDeviceInfo()* 51
Table 2-20 — Arguments for *ConfigureRotameterObservation()* 52
Table 2-21 — Error Codes for *ConfigureRotameterObservation()* 53
Table 2-22 — Arguments for *GetRotameterInformation()* 53
Table 2-23 — Error Codes for *GetRotameterInformation()* 54
Table 2-24 — Arguments for *AdmitTrafficQos()* 54
Table 2-25 — Error Codes for *AdmitTrafficQos()* 61
Table 2-26 — Reason Codes for *AdmitTrafficQos()* 61
Table 2-27 — Arguments for *UpdateAdmittedQos()* 62

Table 2-28 — Error Codes for *UpdateAdmittedQos()* 65
Table 2-29 — Reason Codes for *UpdateAdmittedQos()* 65
Table 2-30 — Arguments for *ReleaseAdmittedQos()* 65
Table 2-31 — Error Codes for *ReleaseAdmittedQos()* 67
Table 2-32 — Arguments for *GetExtendedQosState()* 67
Table 2-33 — Error Codes for *GetExtendedQosState()* 68
Table 2-34 — Arguments for *SetPreferredQph()* 68
Table 2-35 — *SetPreferredQphResults* for *SetPreferredQph()* 69
Table 2-36 — Arguments for *GetUnexpectedStreamChanges()* 70
Table 2-37 — Error Codes for *GetUnexpectedStreamChanges()* 70
Table 2-38 — Arguments for *VerifyTrafficHandle()* 71
Table 2-39 — Error Codes for *VerifyTrafficHandle()* 71
Table 2-40 — Arguments for *UpdateTrafficLeaseTime()* 72
Table 2-41 — Error Codes for *UpdateTrafficLeaseTime()* 72
Table 2-42 — Arguments for *SetL2Map()* 73
Table 2-43 — Error Codes for *SetL2Map()* 73
Table 2-44 — Error Code Summary 73
Table 2-45 — Common Reason Codes 75
Table 2-46 — Actions in Version 3 and Version 2 76
Table 2-47 — State Variables in Version 3 and Version 2 77
Table B.1 — Priority Mapping 98
Table B.2 — Traffic Specification Parameters 100
Table B.3 — ProtoTspec Parameters 102
Table B.4 — AllocatedResources Parameters 103