International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.361

Amendment 1

(06/2008)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Quality of service architecture for audiovisual and multimedia services

---

End-to-end quality of service (QoS) and service priority signalling in H.323 systems

Amendment 1: New Annex A "IntServ/RSVP support for H.323 systems", Annex B "DiffServ support for H.323 systems" and Annex C "Priority support for H.323 systems"

Recommendation ITU-T H.361 (2006) – Amendment 1

ITU-T

International Telecommunication Union

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	H.230–H.239
Communication procedures	H.240–H.259
Coding of moving video	H.260–H.279
Related systems aspects	H.280–H.299
Systems and terminal equipment for audiovisual services	H.300–H.349
Directory services architecture for audiovisual and multimedia services	H.350–H.359
Quality of service architecture for audiovisual and multimedia services	**H.360–H.369**
Supplementary services for multimedia	H.450–H.499
MOBILITY AND COLLABORATION PROCEDURES	
Overview of Mobility and Collaboration, definitions, protocols and procedures	H.500–H.509
Mobility for H-Series multimedia systems and services	H.510–H.519
Mobile multimedia collaboration applications and services	H.520–H.529
Security for mobile multimedia systems and services	H.530–H.539
Security for mobile multimedia collaboration applications and services	H.540–H.549
Mobility interworking procedures	H.550–H.559
Mobile multimedia collaboration inter-working procedures	H.560–H.569
BROADBAND AND TRIPLE-PLAY MULTIMEDIA SERVICES	
Broadband multimedia services over VDSL	H.610–H.619
Advanced multimedia services and applications	H.620–H.629
IPTV MULTIMEDIA SERVICES AND APPLICATIONS FOR IPTV	
General aspects	H.700–H.719
IPTV terminal devices	H.720–H.729

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T H.361

End-to-end quality of service (QoS) and service priority signalling in H.323 systems

Amendment 1

New Annex A "IntServ/RSVP support for H.323 systems", Annex B "DiffServ support for H.323 systems" and Annex C "Priority support for H.323 systems"

Summary

Amendment 1 to Recommendation ITU-T H.361 introduces three new annexes.

Annex A describes the procedures of H.323 quality of service (QoS) signalling when RSVP-based QoS signalling is used in the transport plane. Resource reservation protocol (RSVP) is the QoS signalling protocol used in the integrated services (IntServ) architecture. RSVP is a path-based QoS mechanism which is used to reserve resources for both individual flows and flow aggregates. RSVP can be used in a pure IntServ architecture or can be coupled with differentiated services architecture (DiffServ) to provide IntServ operation over DiffServ network. Annex A describes the procedures for H.323 QoS to allow the use of RSVP in the transport plane.

Annex B describes the procedures of H.323 QoS signalling under the differentiated services (DiffServ) architecture in the transport plane. DiffServ is a class-based QoS architecture which supports in-band signalling. The signalling occurs via a value defined in the differentiated services (DS) field of the IP header. This value is referred to as the differentiated services code point (DSCP). The packet forwarding treatment given to a packet in a network device is based on the DSCP value.

Annex C describes the QoS service priority support signalling used for H.323 systems. The service priority mechanism defines procedures and constructs within the signalling plane that are used to prioritize bearer traffic during periods of resource contention. This allows traffic of higher priority to receive preferred QoS treatment.

Source

Amendment 1 to Recommendation ITU-T H.361 (2006) was approved on 13 June 2008 by ITU-T Study Group 16 (2005-2008) under Recommendation ITU-T A.8 procedure.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

CONTENTS

Recommendation ITU-T H.361

End-to-end quality of service (QoS) and service priority signalling in H.323 systems

Amendment 1

New Annex A "IntServ/RSVP support for H.323 systems", Annex B "DiffServ support for H.323 systems" and Annex C "Priority support for H.323 systems"

Add the following normative references to clause 2.1:

– ITU-T Recommendation H.245 (2008), *Control protocol for multimedia communication*.

– ITU-T Recommendation H.323 (2006), *Packet-based multimedia communications systems.*

– ITU-T Recommendation H.460.4 (2002), *Call priority designation for H.323 calls*.

– ITU-T Recommendation H.460.11 (2004), *Delayed call establishment within H.323 systems.*

– ITU-T Recommendation H.460.14 (2004), *Support for Multi-Level Precedence and Preemption (MLPP) within H.323 systems*.

– IETF RFC 2210 (1997), *The Use of RSVP with IETF Integrated Services*.

– IETF RFC 3550 (2003), *RTP: A Transport Protocol for Real-Time Applications*.

Add the following informative reference to clause 2.2:

– IETF RFC 4594 (2006), *Configuration Guidelines for DiffServ Service Classes*.

Add the following abbreviations to clause 4:

PHB Per-Hop Behaviour

RAS Registration, Admission and Status

SLA Service Level Agreement

Insert new Annexes A, B and C as follows:

Annex A

IntServ/RSVP support for H.323 systems

(This annex forms an integral part of this Recommendation)

A.1 Summary

This annex describes the procedures of H.323 QoS signalling when RSVP-based QoS signalling is used in the transport plane. Resource reservation protocol (RSVP) in IETF RFC 2205 is the QoS signalling protocol used in the integrated services (IntServ) architecture. The use of RSVP in integrated services architecture is described in IETF RFC 2210. RSVP is a path-based QoS mechanism which is used to reserve resources for both individual flows and flow aggregates. RSVP can be used in a pure IntServ architecture or can be coupled with differentiated services architecture (DiffServ) to provide IntServ operation over DiffServ network described in IETF RFC 2998. This annex describes the procedures for H.323 QoS to allow the use of RSVP in the transport plane.

A.2 Background

RSVP is a QoS signalling protocol that enables applications to request reservation of network resources. These requests dictate the level of resources (e.g., bandwidth, buffer space) that must be reserved along with the transmission scheduling behaviour. The transmission scheduling behaviour must be installed in the network layer devices (e.g., routers) to provide the desired end-to-end QoS commitment for the data flow. The QoS can be provided on a per-flow basis according to requests from the end application. RSVP is described in IETF RFC 2205 and also summarized in Appendix II of ITU-T Rec. H.323. For higher scalability, RSVP has been extended to reserve resources for aggregation of flows. RSVP offers a "guaranteed" and a "controlled" service to the network.

The guaranteed service is for real-time applications that are unable to handle delay – it tries to deliver a practicable, constant stream of network capacity that is as close as possible to the end-to-end network delay.

The controlled-load service is a better than best-effort service; it tries to deliver end-to-end network capacity as close as possible to the condition of an unloaded network, but still provides the best-effort service. Controlled-load contracts agree that a flow will be handled within a certain range, but variance is anticipated. It is not expected to accept or use specific values for control parameters that include information about delay or loss.

In RSVP, traffic can be characterized by peak rate of flow (bytes per second), maximum datagram size/maximum burst size (bytes), token bucket rate/service rate/bandwidth (bytes per second), slack term/delay (milliseconds), variation in delay, and other parameters. It may be noted that packet losses (or bit errors) are not taken into account by RSVP specifications.

As described above, RSVP may be utilized in two ways. One is a pure IntServ approach where RSVP acts not only on the control plane providing admission control but is also used on the data plane providing the policing, queueing and scheduling of the flow. This was the original model of RSVP. However, as the per-flow state information with RSVP increases proportionally with the number of flows, it causes storage and processing overhead on the routers. To address this issue, the control plane and the data plane actions in RSVP were separated in the IntServ/DiffServ approach in IETF RFC 2298. RSVP acts on the control plane and allows class-based processing in the data plane. This has helped alleviate some of the scaling concerns.

A.3 Procedures for RSVP

A.3.1 Pre-call procedures

RSVP reservations can only be made by endpoints or network entities along the path of the media flow. In a gatekeeper-routed call signalling, media can be routed via the gatekeeper. In such a model, the gatekeeper can make RSVP reservations on behalf of the endpoint. Since it is common to route media directly between the endpoints, it is best for the endpoints to do the RSVP reservations itself. Endpoint-based reservations also enable resource reservation along the entire path of the media flow.

If the endpoint is capable of initiating RSVP and desires to do so, it selects **endpointControlled** in the **transportQoS** structure in the admission request (ARQ) message. If the gatekeeper is configured to perform the RSVP signalling on behalf of the endpoint, the gatekeeper rejects the selection and overwrites it with **gatekeeperControlled** when returning the **transportQoS** structure in the admission confirm (ACF) message. **GatekeeperControlled** RSVP is applicable only in scenarios where the media is routed through the gatekeeper. If the gatekeeper's policies require the endpoint to initiate RSVP, then the gatekeeper ensures that the **transportQoS** structure contains **endpointControlled** when returning the **transportQoS** in the ACF. If the endpoint indicates **noControl** or **gatekeeperControlled** and QoS control is required to be supported in the endpoint, then the gatekeeper rejects the request and returns the admission reject (ARJ) message and provides the appropriate error (qoSControlNotSupported) in the admission reject reason parameter. This indicates to the endpoint that the ARQ must be attempted with **endpointControlled** and include all relevant parameters in the **transportQoS**.

The endpoint may also negotiate the QoS selection during the registration process by including the **transportQoS** structure in the registration request (RRQ) message. In such a case, the selection applies to all calls made by the endpoint. Any selection made by the endpoint in an admission request (ARQ) overrides the selections made in an RRQ.

In the **transportQoS** structure, the endpoint may provide the necessary **qosType** and **qosClass** in the **qosDescriptor** structure. In the **qosType**, the endpoint sets the **qosType** to either "required" or "desired" depending upon the importance of QoS for the media flow. If the media flow is not to be initiated without securing the required QoS for the flow, then the endpoint selects the "required" **qosType**. If QoS is optional for the media flow or if the media flow is allowed to be initiated without securing the necessary QoS, then the endpoint selects "desired" QoS. The endpoint may provide the traffic characteristics to the gatekeeper in the **rsvpParameters** contained in the **qosCapability** structure. The endpoint may also indicate the differentiated services code point (DSCP) to be used for the media flow in the **dscp** parameter.

The purpose of providing the **qosDescriptor** structure in the **transportQoS** to the gatekeeper is to allow the gatekeeper to enforce policies and/or obtain QoS on behalf of the endpoint. The gatekeeper checks the information provided and ensures that the endpoint is permitted to make the selected choices as per configured policies. For example, the network administrator may have configured policies that disallow any call from initiating without the necessary QoS. In such a case, the endpoint will not be allowed to set **qosType** to "desired" QoS.

If the endpoint fails to provide sufficient information in the **transportQoS** structure that it is required to do so, then the gatekeeper can reject the request by responding with an ARJ and indicating the error in the admission reject reason parameter. For example, if endpoints fail to provide the traffic characteristics in the **rsvpParameter**, then the gatekeeper can reject the request. If a choice indicated by the endpoint is unacceptable to the gatekeeper, the gatekeeper does not reject the message but instead indicates its preferred choice by returning the **transportQoS** structure in an ACF message. For example, if the endpoint had selected "desired" QoS and the policies dictate that QoS is required, then the gatekeeper can indicate "required" in the

TransportQoS returned in the ACF. If the gatekeeper wishes to change the **dscp** value to be used for the flow, it can do so by indicating its preference in the returning **TransportQoS** structure.

All QoS decisions provided by the gatekeeper are binding on the endpoints. If the endpoint is unable to honour the request, it cannot proceed with call establishment.

A.3.2 Call setup procedures

It is necessary to synchronize the call signalling messages and the RSVP messages to allow the network reservation to be established before the callee is alerted. The procedures in ITU-T Rec. H.460.11 shall be followed to ensure that the called party is not alerted until the desired QoS-enabled media channels are established. This allows the call establishment to be discontinued if resources are unavailable and prevents ghost rings. This is a MUST in the case of a flow with a "required" **qosType**. In the case of "desired" **qosType**, the RSVP resource establishment may proceed in parallel. However, it is preferred that RSVP transactions be completed before media is transmitted such that the entire media flow can benefit from the established reservation.

If the caller requests QoS procedures but the callee is not able to support the required QoS procedures, then the callee rejects the OLC by issuing an OpenLogicalChannelReject citing "qoSControlNotSupported" as the cause.

There are three call signalling ways which allow the RSVP negotiation to occur before the callee is alerted. They are:

– Fast start procedures.

– Inclusion of the H.245 address in the setup message.

– H.245 tunnelling.

A.3.2.1 Fast start procedure

In the fast start procedure, the **openLogicalStructures** are provided by the caller in the initial setup message itself. This allows the callee to initiate RSVP messages without waiting for additional messages from the caller. Please refer to Figure A.1 below. In this figure, for the sake of simplicity, only one gatekeeper is shown. The call flow remains the same when multiple gatekeepers are involved.

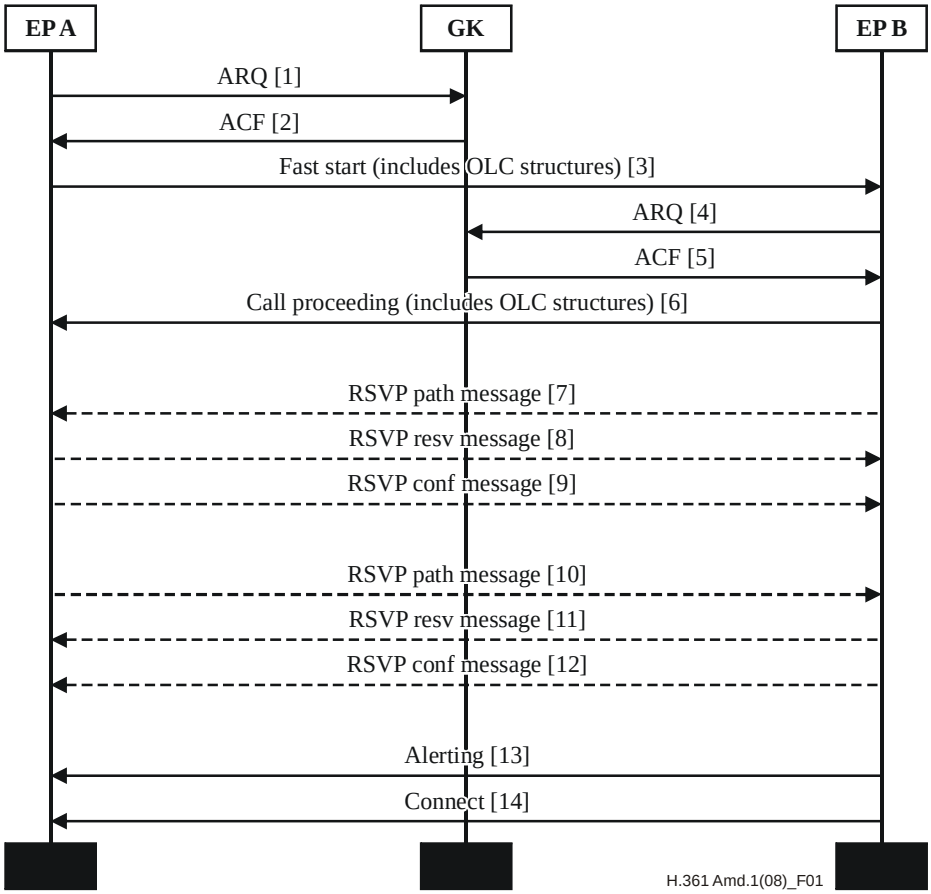

Figure A.1 – RSVP with fast start call establishment

After the endpoint has been admitted by the gatekeeper [1 and 2], the endpoint sends a setup message including a list of prioritized **qosCapability** parameters in the **OpenLogicalChannel** [OLC] structures [3]. Inclusion of the **qosCapability** structures indicates to the called endpoint that the caller wishes to establish QoS for the media flow. If the calling endpoint wishes to do RSVP, it includes the **rsvpParameters** in the **qosCapability** structure. In addition to the **rsvpParameters**, the endpoint may also include the **qosType** and the **qosClass**. In the **qosType**, the called endpoint indicates whether QoS is "required" or "desired" for the flow.

The called endpoint responds to the setup message by sending a call proceeding message with a fast start element [6]. In this message, the called endpoint includes a subset of OLC structures that it has selected. The OLC structures contain one or more **qosCapability** structures that are selected by the called endpoint. The **qosType** for the flow is decided as the strongest of the caller's and callee's preference. The OLC structures exchanged in the setup message and in the call proceeding message contain media port information which allows the endpoints to initiate RSVP.

Since RSVP is unidirectional, both the endpoints initiate RSVP. They send the RSVP **Path** message [7] and [10]. The message is sent to the same address and port as that of the media flow for a point-to-point flow. The traffic specification object in the RSVP message is derived from the **rsvpParameters** provided in the **qosCapability** structure. Information on how to derive the information needed for **rsvpParameters** is outside the scope of this Recommendation. The receiver of the flow returns an RSVP **Resv** message in response to the RSVP **Path** message [8] and [12]. The **ResvConf** message is an optional message which is only sent if requested in the RSVP **Resv** message.

The endpoints may optionally exchange flow control messages to prevent any media flow exchange until the reservation is established. The called endpoint considers the reservation process to be

complete when it receives the RSVP **Resv** message for the flow it is about to originate and a RSVP **ResvConf** message for the flow it is about to receive. If the endpoint has not requested an **ResvConf** message, then it could make use of a timer to decide when the reservation is established. If no RSVP **ResvError** message is received before the timer expires, the reservation may be considered as complete. The time value should be at least one round-trip and may be more to ensure that there is sufficient time to the RSVP **Resv** message to reach the other endpoint or for an RSVP **ResvError** message to be received back. Once the reservation is completed, the called endpoint alerts the user and transmits an alerting message followed by a connect message when the endpoint goes off-hook.

A.3.2.2 Inclusion of the H.245 address in the setup message

If the endpoint does not wish to do fast start, it can achieve synchronization of QoS signalling and call signalling by including the H.245 address in the setup message. Figure A.2 describes the call flow for this type of a call. The H.245 address allows the called endpoint to initiate a H.245 capability exchange while withholding the alerting. The H.245 capability exchange [7] includes a **qosDescriptor** structure. The **qosDescriptor** may include a **qosType** element. This indicates to the called endpoint that QoS exchange is necessary for the flow. The **rsvpParameters** may also be included with just the **qosMode**. The presence of **qosMode** and the **rsvpParameter** indicates to the called endpoint that the caller prefers RSVP-based QoS signalling. Since the capability exchange is not stream-specific, stream-level parameters are not included in the capability exchange. The call proceeding message is sent by the called endpoint to prevent any timer expiry in the caller side.

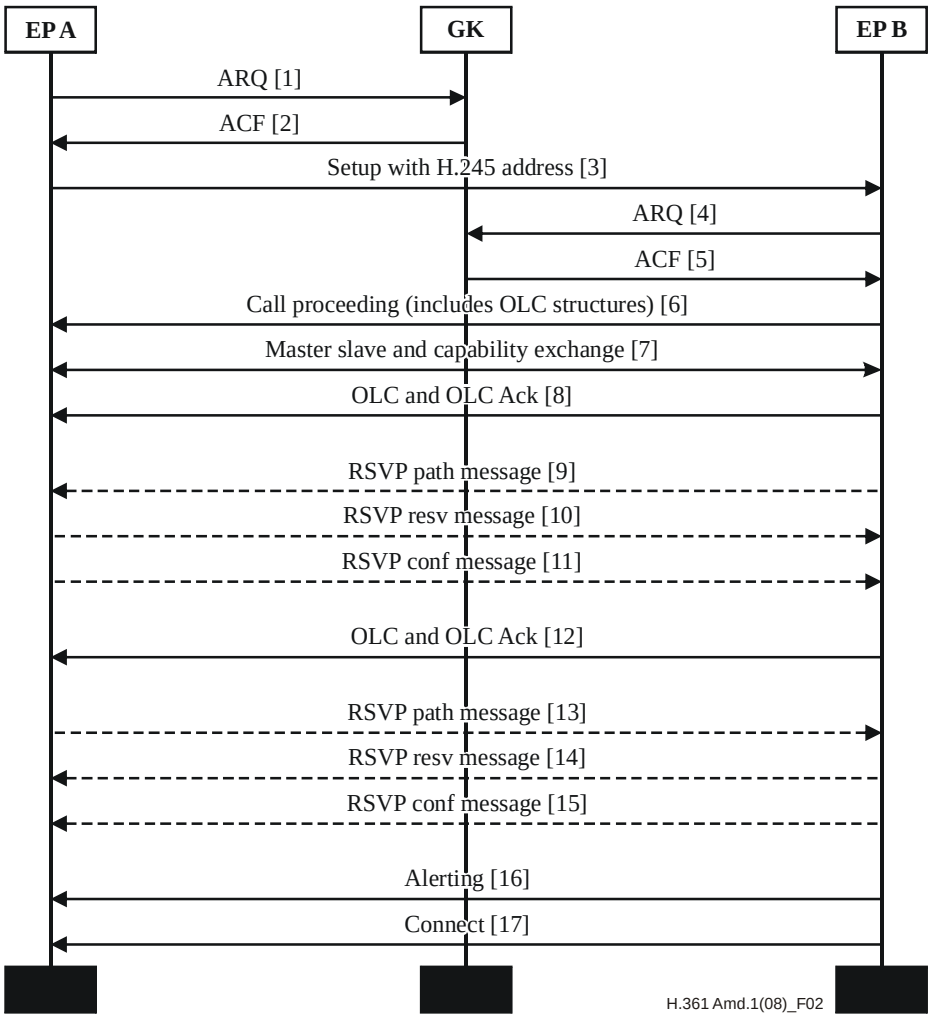

Figure A.2 – RSVP with H.245 address in setup message

Once the OLC exchange occurs [8] and [12], and the reservation is complete as described above, the called endpoint alerts the user and transmits the alerting message followed by the connect message.

A.3.2.3 RSVP without synchronization

As mentioned above, synchronization of RSVP and call signalling is necessary for flows that require a "required" **qosType**. If no synchronization is possible, such as when an endpoint cannot participate in fast start call setup or include H.245 address in setup or perform H.245 tunnelling, then the **qosType** cannot be "required". This is because there is no indication to the called endpoint to withhold the alerting in other mechanisms. It is only possible to support a "desired" **qosType**. The "desired" **qosType** means that QoS will be attempted but the call will continue even when the QoS fails to be secured. This is suitable for call establishment when alerting occurs before the QoS is secured. Figure A.3 shows such a call flow.

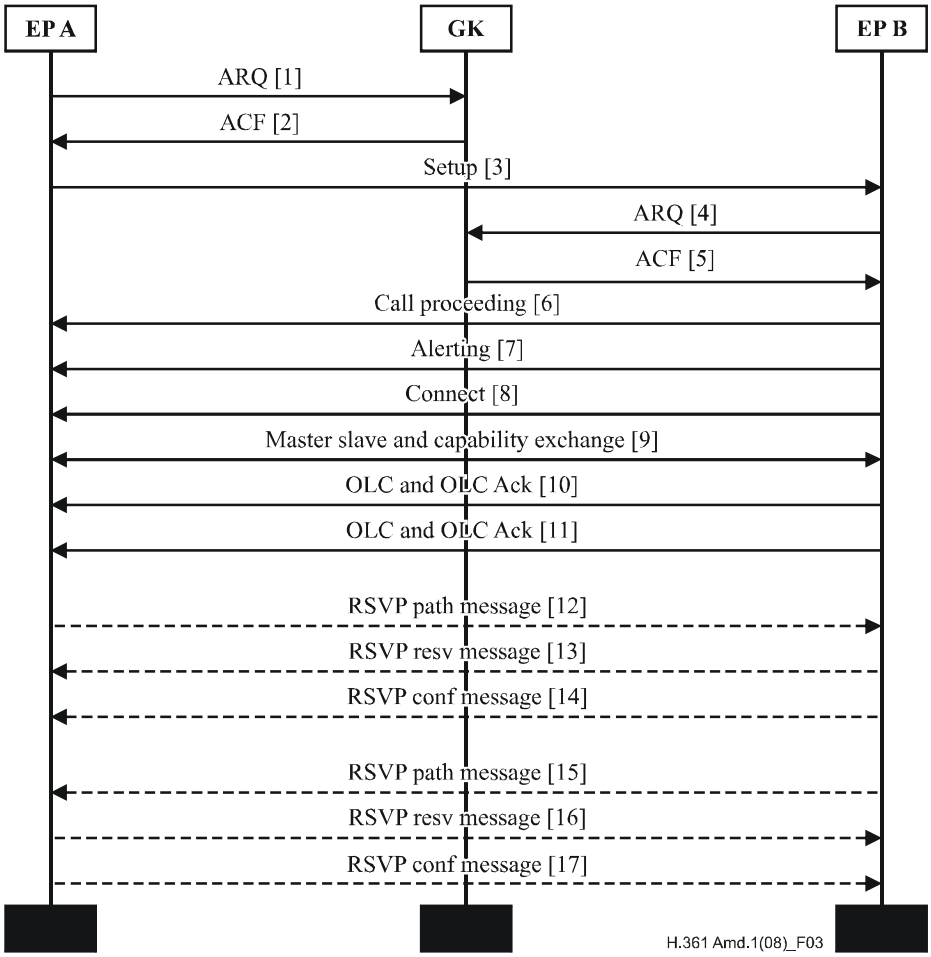

Figure A.3 – No synchronization with RSVP

In the above figure, RSVP can only be initiated after the capability exchange and the OLC exchange. This is because the media port is only available after the OLC exchange. Since the called endpoint is unaware that the caller requires RSVP, it alerts the callee and responds with alerting and connect messages. In this scenario, if RSVP requests are unsuccessful then there is no impact to the call at all. The call proceeds with no QoS.

A.3.2.4 Procedures for gatekeeper-routed call signalling

If the gatekeeper routes call signalling as well as the control channel, then the calling endpoint is not aware of the destination endpoint's address information before the alerting phase. Therefore, the endpoint cannot make an end-to-end RSVP reservation with synchronization. In such a case, the gatekeeper may be configured to make the RSVP reservation. The gatekeeper may direct the reservation to the called endpoint or to the called endpoint's gatekeeper, depending upon whether the endpoints use the same gatekeeper or not. There may also be a series of reservations from the caller's endpoint to the callee's endpoint such as caller's endpoint to the caller's gatekeeper, caller's gatekeeper to callee's gatekeeper and then callee's gatekeeper to the callee's endpoint.

If the gatekeeper routed just the call signalling and did not route the control channel, then the presence of the H.245 control channel address included in the setup message could signal to the called endpoint to withhold the alert to the user to facilitate the RSVP process. The endpoints can then initiate and complete the RSVP process as outlined above. The connection establishment completes after the reservation process is complete.

A.3.2.5 Releasing a call

If RSVP is involved, the call release procedures must include releasing the reservations as well. If no RSVP refreshes are received for a certain time period then the RSVP state will be automatically deleted. However, the exchange of explicit teardown messages is recommended so that the network resources are freed up and made available as quickly as possible. The following steps are for releasing a call:

– Close all the open channels after exchanging the **CloseLogicalChannel** messages.

– Close the H.245 control channel if one had been opened.

– Finally, exchange the **ReleaseComplete** messages and release the call.

In addition to the above, the RSVP process in the endpoint must be discontinued. This involves the following steps:

– If the endpoint is a receiver, then stop transmitting the **Resv** refreshes and transmit a **ResvTear** message for the reservation originally made.

– If the endpoint is a sender, then stop transmitting the **Path** message and transmit a **PathTear** message to delete the path state in the network.

A.3.2.6 RSVP error handling

In this clause, the procedures that need to be invoked on account of an RSVP error have been considered. There are two levels of decisions that are to be made. They are as follows:

– The first decision pertains to each individual channel. The decision required here is what to do with a media channel for which the reservation has failed. This decision can be derived from the QoS modes in the derived set. For more information on derived QoS mode, please refer to the main body of this Recommendation. Table A.1 below gives some examples of the derived sets and the actions that are required in each case. This decision could cause the media channel to remain unestablished.

Table A.1 – Failure actions

DerivedQoSType set	Actions to be taken
Required	In this derived set, best effort is not an option. Hence, if a reservation fails, then the media channel is not established.
Desired	In this set, the media channel may proceed with best effort (BE). Therefore, if a reservation fails for media channel, the media channel may be opened and media may be transmitted as BE.

– The second decision involves the action to be done when some channels remain unopened due to reservation failures. A simple decision would be to fail and release the call. However, to allow greater flexibility, an endpoint could decide to allow the call even if some of the channels failed. For example, a videoconferencing call may be allowed to continue as an audio-only call if the video channel was not established due to the reservation failure while the audio stream had a successful reservation. The decision to allow the call even if some of the channels are closed is guided by administrative policies. However, it is also highly recommended that messages are provided to the endpoints to explain the change in the call. Given below are some examples of call handling when some channels are closed.

Example 1: Call release with reservation failure

In this example, the call consists of two channels: one audio channel and one video channel. It is assumed that the endpoints are configured to fail the call if any media channel is closed.

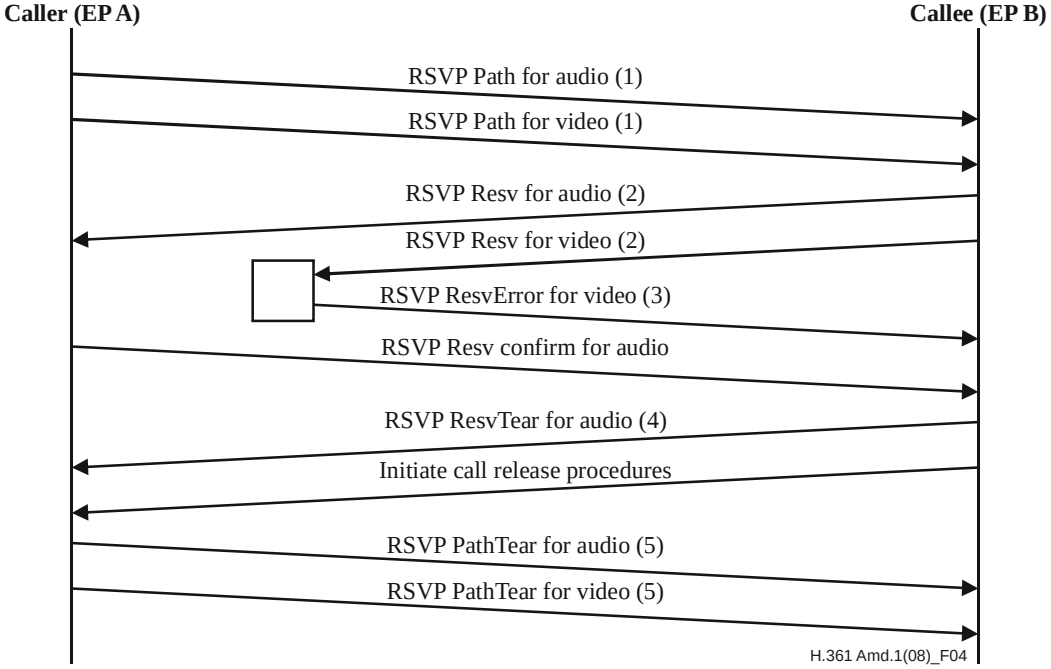

Figure A.4 – Call failure due to reservation failure

1) EP A sends a **Path** message for its audio and video channels.
2) EP B responds with a **Resv** message for both the channels.
3) The reservation for video is denied in one of the network devices along the path and the device sends a **ResvError** message back to EP B.

4) Since the reservation for the video channel fails, EP B decides to close the video channel (best effort is not an option in the derived set). On the failure of the video channel, EP B initiates the release process to release the call. It also sends a **ResvTear** message for the audio channel to tear down the reservation that has already been made.

5) As part of the release process, EP A receives a **ReleaseComplete** message from EP B. This message indicates the reason for call failure, which in this case is "nobandwidth". EP A informs the user of the call failure along with the reason for it. It also sends **PathTear** messages to tear down the path state that has been established.

Example 2: Modified call with reservation failure

Given below is an example where the reservation for the audio channel succeeds, whereas the reservation for the video channel fails. In this case, the call is modified and continues as an audio-only call.

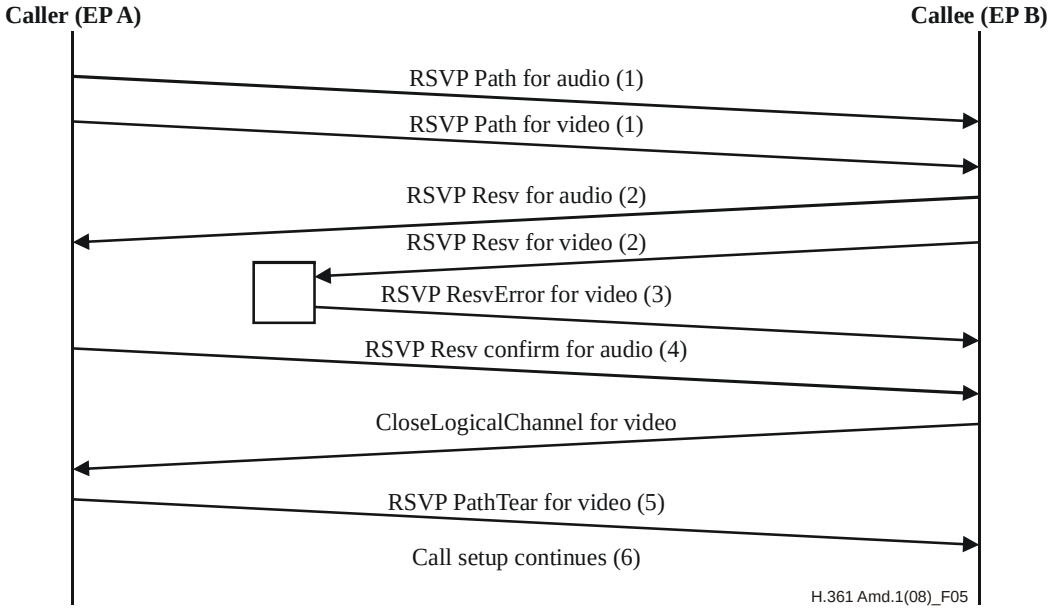

Figure A.5 – Call continues with reservation failure

1) EP A sends a **Path** message for its audio and video channels.

2) EP B responds with a **Resv** Message for both the channels.

3) The reservation for video is denied in one of the network devices along the path and the device sends a **ResvError** message back to EP B.

4) EP B receives a **ResvConf** message for the reservation for the audio channel. Since the reservation for the video channel fails, EP B closes the video channel. In this case, EP B decides to continue the call as an audio-only call as the audio channel has a successful reservation.

5) EP A receives the reason for the failure of the video channel from the **CloseLogicalChannel** message and communicates it to the end user. The video channel is closed and EP A sends a **PathTear** message to delete the path state for that channel in the network.

6) The call setup continues and establishes an audio-only call.

If a channel is closed due to reservation failure, then the reason provided in the **RequestChannelClose** message is set to "reservation failure" and the error code received from the network (in the RSVP message) is included in the newly introduced parameter "networkErrorCode".

A.3.2.7 Retrying reservations

As explained in the above clauses, a call can continue without reservation if the following two conditions are met:

– If best effort QoS treatment is implicitly allowed for the channel for which the reservation has failed.

– If the call is configured to continue even after the failure of one or more channels.

There are two reservation retry options available. The first one is to allow the reservation for the channel to be retried at regular intervals. If successful, the endpoint should provide an indicator to the end user that the channel now has reserved resources. The retry interval is implementation-dependent and guided by administrative and network policies.

The **Resv** message for retrying the reservation cannot be sent in the absence of a path state, which is created by a **Path** message. Hence, it is necessary that the **Path** message be continuously refreshed even after a reservation failure if retry is to be attempted. At the end of each retry interval, the receiver endpoint will respond to the **Path** message by sending a **Resv** message to retry the reservation. Once the endpoint receives a **ResvConf** message to confirm that the reservation has been made, it can inform the user that the channel now has reserved resources.

The second retry option is to restart the closed channel at certain time intervals. It could open the logical channel by exchanging the OLC messages containing the QoS parameters. By exchanging flow control command, the endpoints can disallow media to be transmitted in that channel until the reservation is successful. The RSVP message is exchanged requesting reservations for the newly created channel. If the reservation is successful, then the user is notified and the media is allowed to flow through the newly opened channel.

Thus, this feature enables the call to acquire reserved resources and improved quality during the call even if the reservation had failed originally.

A.3.2.8 Modifying a call

The following are some of the ways in which a call may be modified causing the bandwidth used in the call to change. If the bandwidth used is changed then the RSVP reservation must also be changed:

– **Change in the codec used**: The endpoints may decide to use a different codec from the one selected at call establishment time. This may require the existing logical channel to be closed and a new channel for the newly selected codec to be opened. Hence, the RSVP reservation must also be stopped for the old channel and re-reserved for the new channel.

– **Change in the bit rate used**: The endpoint may decide to either increase or decrease the bit rate used in the call. When an endpoint changes the bandwidth used, it should make a bandwidth request (BRQ) to the gatekeeper. The gatekeeper can either accept or reject the request (BCF/BRJ). This causes the endpoints to close the existing logical channel and open a new one with the new bit rates. If the media channel is closed and re-opened then the RSVP reservation must also be torn down and requested again. If the physical channel remains the same with no changes to ports or addresses then the reservation can just be modified. The advantage here is that if the modified RSVP is rejected then the older reservation still remains. In such a case, the endpoint may decide not to change to the new bit rate.

– **Opening and closing of media channels**: At any time, the endpoints may decide to either add new channels or release some existing channels without closing the call. If a new channel is added during the call, the RSVP reservation process must be followed. If the reservation is rejected, then the derived set should be used to make a decision about the channel/call. This is similar to the process followed during the call establishment. If a channel is closed then the reservation release process should be followed.

Annex B

DiffServ support for H.323 systems

(This annex forms an integral part of this Recommendation)

B.1 Summary

This annex describes the procedures of H.323 QoS signalling under the differentiated services (DiffServ) architecture in the transport plane. DiffServ is a class-based QoS architecture which supports in-band signalling. The signalling occurs via a value defined in the type of service (ToS) byte (also called differentiated services (DS) field defined in IETF RFC 2474 of the IP header). This value is referred to as the differentiated services code point (DSCP). The packet forwarding treatment given to a packet in a network device is based on the DSCP value and is known as the per-hop behaviour (PHB).

B.2 Background

The IETF's DiffServ is an in-band signalling mechanism that is relatively simple to implement. From the development point of view, the mechanism calls for the appropriate DSCP value marking in the IP header of the packet. It is a very scalable solution and can be deployed in network clouds where there is very high traffic load. The disadvantages of this mechanism are that there is no explicit admission control, no flow-based or session-based QoS treatment and no feedback to the user when adequate QoS cannot be granted.

DiffServ requires that some capacity in the network be set aside for particular classes of traffic. In this mechanism, a set of primitives are applied to the traffic. They are: classification, policing, shaping and marking. Classification is done based on the DSCP values contained in the IP header of the packet. The DSCP value is a 6-bit value and therefore can range from 0 to 63. Some of the values within this range are defined by IETF standards and their associated per-hop behaviour is outlined as well. There are some values within that range that are left for experimental purposes. DiffServ policing primitives will police the traffic based on the given profile. If traffic within a class exceeds the given profile, then there are either dropped or shaped. Shaping primitives causes the traffic to be delayed and forwarded rather than dropped when the profile max has been reached. Shaping also helps in smoothing the flow of packets within a class. Finally, the marking primitives will mark or remark the packets based on the given DSCP value.

Using these above mechanisms and other QoS tools, the DiffServ method can provide a variety of services such as premium service for applications requiring low-delay and low jitter service, assured service for application requiring better reliability than best-effort service, and others.

The increased scalability in the DiffServ method is due to the following:

– Classification using just the DSCP values.

– Limited state information as state is maintained per class and not per flow.

– All primitives are not required in every hop. One example could be that all primitives are employed at the edge while just the classification primitive is employed at the core routers.

The main disadvantage of using this exclusively for QoS is that there is no protection of traffic within a class. For example, if too many packets arrive at the router for admission within the same class, all the excess packets will most likely be dropped. This causes quality degradation across all traffic flows. Instead, if combined with an explicit admission control, only a smaller set of flows would have been affected while the other flows would be provided with a guaranteed quality. The other major disadvantage of the DiffServ approach is that there is no feedback to the application.

B.3 QoS mechanisms in a DiffServ network

This clause discusses the possible ways in which a H.323 QoS may be deployed in a DiffServ network.

B.3.1 Differentiated services only

In this method of QoS, the appropriate DSCP value is signalled in the media packets. If the network is not congested in the class in which the media packet is admitted, the flow will receive the right QoS as indicated by its DSCP value. The problem occurs only when the class is over-committed. In such a case, the network device has to drop or delay some packets such that the traffic is within the limits configured for that class. Since there is no feedback to the application when a packet is over the limit of that class, the H.323 application cannot take any corrective actions or provide proper feedback to the user. Therefore, it is generally unsuitable as a sole mechanism for providing QoS in a H.323 system when operating in networks that may experience congestion.

The **transportQoS** structure contains a **dscp** parameter. The endpoint may provide the DSCP value that it intends to use to the gatekeeper during the RAS exchange. The gatekeeper may accept the value provided by the endpoint by returning the same value in its response. The gatekeepers can also force a different value to be used by changing the DSCP value in its response. The endpoint is required to use the value provided by the gatekeeper. If no value is provided, then the endpoint can use the DSCP value of its choice. The endpoint signals EndpointControl with localQoS with this mechanism.

The endpoint may also provide the DSCP value in the OLC exchange during the call-setup procedure. This value is provided in the **qosCapability** parameter. The purpose of providing this value to the called party is to provide the type of per-hop behaviour that the caller is intending to request for that channel. The callee may use this information to select the channel. The callee may also use the same DSCP value for its direction of the flow.

As there is no messaging from the network on QoS issues with this DiffServ-only system, the H.323 QoS system may add enhancements to improve the QoS solution. Some possible enhancements are described below.

B.3.2 Reactive enhancements

One possible mechanism is to monitor and correct. Using RTCP defined in IETF RFC 3550, or other similar mechanisms, the endpoints can monitor the QoS attributes of the flow such as delay, drops, jitter, etc. If the attributes show the QoS falling behind acceptable limits, then corrective actions may be taken by the endpoint. The corrective actions include removing the flow, re-routing the flow using other IP or non-IP routes, or even possibly altering the characteristics of the flow by reducing the average rate or codec to reduce the bandwidth consumption.

These enhancements are not without problems. Some of the problems are:

– Such reactive mechanisms work well if all applications are well-behaved and take corrective actions to reduce congestion. However, badly behaved endpoints that do not take corrective actions, benefit from the actions of the others and continue to cause congestion.

– There is always a risk of over-correction and under-utilization of the available bandwidth. When congestion occurs and packets are dropped or delayed, it affects a wide set of flows traversing that link. If applications for all such flows take corrective actions, then there is likely to be an over-correction and that may cause that class to be under-subscribed.

B.3.3 Proactive enhancements

The H.323 QoS system may perform admission control to gain admittance and then use DiffServ QoS to obtain the appropriate QoS for its flows. One form of admission control is RSVP which is discussed in Annex A. Annex A mentions the possibility of using RSVP for gaining admittance and using DiffServ for the media flow. This is described in IETF RFC 2998. Other admission control mechanisms other than RSVP may also be used.

The gatekeeper can directly interact with the network devices and perform admission control. This is described as option 1 in ITU-T Rec. H.360. This is an off-path admission control mechanism. Once the network devices have admitted the flows, the flows are provided with the necessary QoS. The flows may be identified by the flow-information such as source address, destination address and UDP port information. Alternatively, only the DSCP value of the flow is required for the network to provide the right QoS for the flow. The use of DSCP value is advantageous since no per-flow classifications have to be performed in the network devices, but the disadvantage is that it only works for networks with well-behaved endpoints.

With this off-path admission control mechanism, the endpoint signals "**gatekeeperControlled**" during the RAS exchange. The gatekeeper uses the other relevant information such as **qosClass** and **genericTrafficDescriptor** to interact with the network device and perform admission control. Once the admission is granted, the gatekeeper provides the DSCP to be used for the flow in its response.

B.3.4 Suggested DSCP values

IETF RFC 4594 suggests the DSCP values to be used for the various flows.

Annex C

Priority support for H.323 systems

(This annex forms an integral part of this Recommendation)

C.1 Summary

The main body of this Recommendation defines the use of service priority for prioritizing media streams in order to achieve QoS within an H.323 system. When there is contention for media resources, the elements and procedures defined in this annex are used to achieve the desired QoS results using the service priority element. For example, critical communications can be prioritized above ordinary traffic to ensure quality and reliability. Service providers may use service priority as a means to discriminate between varying levels of service based on SLAs.

Other ITU-T Recommendations such as ITU-T Recs H.460.4 and H.460.14 specify call signalling priority. Service priority as defined by this annex is only concerned with the priority of the media stream. It can be used in the presence or absence of call priority signalling mechanisms.

C.2 Scope

The use of Service Priority for media QoS is described within this annex independently of other call priority mechanisms.

C.3 Service priority

The main body of this Recommendation introduces service priority as one of the elements of the **qosCapability** structure. This clause discusses the **servicePriority** structure. **servicePriority** contains two elements: **servicePrioritySignalled** and **servicePriorityValue**. **ServicePrioritySignalled** indicates whether explicit priority definitions are being provided.

This annex defines three new elements within **servicePriority**. They are the **serviceClass**, **serviceSubclass** and **servicePriorityValue**.

C.3.1 Service class

A service class identifies a specific service provider or service type. The service class can be used to determine the high level policy that will be applied to the media stream. For example, a policy could be applied to one service class that releases bandwidth to be used by other resources of the same service class during resource contention. Another service class might simply mark the traffic for different priority queueing. A class may contain multiple subclasses.

C.3.2 Service subclass

Service subclass defines the context for the service priority value. It may be desirable for a service provider to segment media traffic, allowing prioritization to occur among traffic within that segment or subclass. Priority values can only be compared to other priorities within the same subclass and have no significance to media streams outside of that subclass. When contention for resources occurs, only resources within the same service subclass are evaluated to determine which request will be granted the resource.

When no service class is specified, the service class default of 0 shall be used when signalling the service priority parameter.

C.3.3 Priority values

The priority value is an integer value that describes 256 levels of relative priority for the media flow within a service subclass. An ascending integer order designates a higher priority relative to other priority values. Higher priority values are given preferential treatment with respect to resources within their service subclass. If a contention for resources exists during the admission control process, the resource should be allocated to the request with the higher resource priority.

The priority value may be used with integrated service architecture and RSVP in the transport domain. The priority value is signalled within the RSVP message for prioritization by the RSVP process within a network node. The priority value may also be used in differentiated service architecture in the transport domain. The priority value could help determine the differentiated service code point (DSCP) marking of packets. The packets of a flow are marked in a manner that correlates the importance or criticality of the media stream.

C.4 Resource contention

When there is a contention for a resource, the **servicePriority** parameter is examined to determine which resource request is granted. The **serviceSubclass** is inspected to determine the set of resources that are to be evaluated for use. Within a given **serviceSubclass**, the request made with the highest **servicePriorityValue** is granted. For example, consider a scenario in which **serviceSubclass** 1 and **serviceSubclass** 2 have been defined and resources are associated with each. **serviceSubclass** 1 has no more available resources and a priority call is attempted within **serviceSubclass** 1. The resources in **serviceSubclass** 2 remain unaffected by the new call attempt while the resources in **serviceSubclass** 1 are evaluated to determine if it is possible to grant the resource request. If the **servicePriority** for the new call attempt is higher than other priority values in **serviceSubclass** 1, the appropriate service behaviour is invoked to resolve the resource contention. The actual behaviour with regard to granting resources must be made within the service logic for a given service class. For example, a service may release resources to provide a resource to a higher priority call. Other services may queue resources for use based on the priority of a call. The definition of any such behaviour is beyond the scope of this annex. The **servicePriority** will simply convey the information between service entities, allowing the services to take the appropriate action.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2009