Telecommunication Standardization Sector
Original: English

AVC - 1019

Experts Group for Video Coding and Systems in ATM and Other Network Environments

September 24-27, 1996

STUDY GROUP 15
CONTRIBUTION

Source: Jim Toga, Intel Corporation
email: jtoga@ibeam.intel.com
voice: +1 (503) 264-8816
fax: +1 (503) 264-3485

Title: Transport level QOS options for H.323

Date: September 16, 1996

Motivation

Section 8 in H.225.0 is titled Mechanisms for maintaining QOS. Although it does give options for utilizing H.245 signaling for QOS related media control and monitoring RTCP to assess the quality; it also provides for transport level supplied QOS:

> "Any transport QOS related signaling (e.g. a reservation request to a router) is done by the terminal as soon as possible, or by the gatekeeper on its behalf. The terminal may wish to make any reservations since the gatekeeper may not be logically near the terminal, or be able to make QOS related requests on behalf of the terminal. The means by which either the terminal or the gatekeeper make QOS or bandwidth reservations are beyond the scope of this recommendation."
> (Section 8.1)

Although the method by which the reservation is actually made is beyond the scope of H.323, the general method and coordination of transport QOS between H.323 entities should be specified - to prevent conflicting interoperability issues. Participants in a conference must be able to signal their intentions and responsibilities in a standard, protocol specified manner.

Almost all (all?) initial implementations of H.323 will be modeled after appendix D in H.225.0; namely, they will operate in an IP environment. For this reason it follows that an additional informative appendix (F?) should be added describing the signaling/usage/deployment of a native IP based QOS mechanism.

Introduction

The ability for H.323 implementations to take advantage of QOS services in a Non-Guaranteed Quality Service LAN must be signaled in a standardized manner. For this reason there are a few additional elements that should be added to H.245 (more specifically 225.0 parameters) to facilitate this. There are many conceivable gradations of service that may be provided by a transport layer service (or taken advantage of by the H.323 endpoint); the mechanisms proposed provide this flexibility.

RSVP is one such QOS service that can be provided in the unreliable environment of IP-based packet networks. RSVP enables endpoints to make reservations for a given set of QOS parameters. Although, once the reservation has been granted it is guaranteed[1], there are occasions in which the transport layer cannot provide the requested level of QOS. In the absence of the QOS reservation (RSVP), the media traffic resorts to the standard 'best effort', mechanism of delivery.

The IETF protocol, RSVP provides "a general facility for creating and maintaining distributed reservation state across a mesh of multicast or unicast delivery paths"[2]. Some of the salient points of RSVP are as follows:

- RSVP supports both uni/multicast environments
- RSVP is tied to specific streams (i.e. specific transport address pairs)
- RSVP adapts dynamically to changing group membership and routes
- RSVP is uni-directional
  (although this models H.323 media channels, it is problematic in the end-end signalling)
- RSVP is receiver oriented - the recipient of the media stream makes the reservation (scaleable)
- RSVP is soft-state based, provides for route changes, and ungraceful 'drop-offs'
-

RSVP supports several QOS models; controlled load, commited rate, and Guaranteed.

---

[1] Reservations must be periodically refreshed; there is no guarantee that a refresh request will be granted however, during the reservation period the QOS is guaranteed.
[2] Resource **ReSerVation Protocol (RSVP)** Version 1 Functional Specification

RSVP Background

In the following description, the high level usage of RSVP in a simple H.323 conference will be outlined.

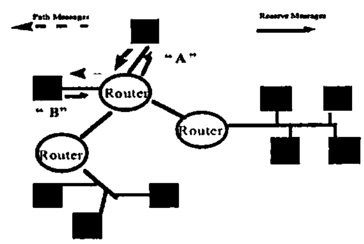

Figure 1

In figure 1 above endpoint A is sending a media stream to endpoint B. Endpoint A would cause *Path* messages to be sent out to B. These *Path* messages go through routers and leave 'state' on their way tracing towards B. Path mesages contain the complete source and destination addresses of the stream and a characterization of the traffic that the source will send. Endpoint B would use the information from the *Path* to make the *Reserve* request for the full length of the path. Reserve messages contain the actual reservation and will generally be the same as the traffic specification in the Path message. Note that B would send a related *Path* message to A; A could then make the *Reserve* request in the reverse direction.

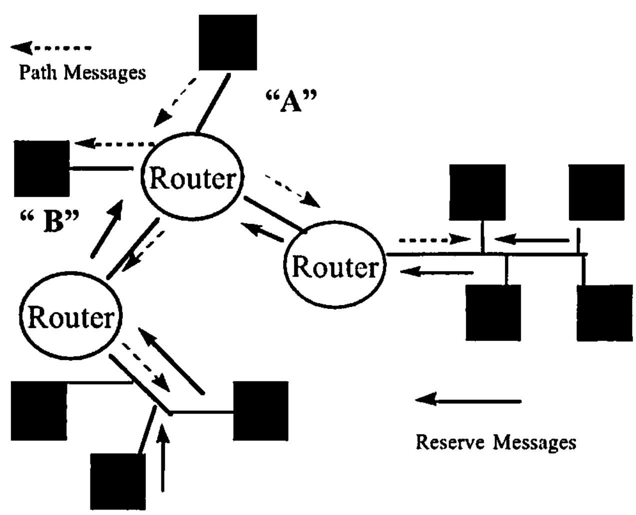

Figure 2

In figure 2 above, a multipoint conference is shown. The *Path* messages are utilized in the same manner as the simpler point to point case. It should be noted that the *Reserve* requests are aggregated by the routers and keep redundant requests from occurring upstream.

Since the *Path* messages need to contain the complete destination/ source addresses and a traffic specification, *Path* messages can be sent after the H.245 capabilities negotiation and

OpenLogicalChannel for the particular media stream. The reservation is released after the CloseLogicalChannel using the RSVP *Teardown* messages.

In order to provide an end to end QOS service in a typical LAN environment, RSVP will have a policy component such as that supplied by SBM (Subnet Bandwidth Management) [3]. RSVP is the reservation and guarantee portion on point-to-point links. SBM is the policy and admission control for shared media (e.g. Ethernet). This policy component can be used to permission both at the overall call level, and for incremental stream level permissioning. After the policy for high level call admission, and individual stream admission has been approved , the resources can be reserved via RSVP. In order for H.323 to work in this environment (or any externally supplied QOS environment), both gatekeeper-endpoint and endpoint-endpoint coordination must be signaled. SBM signaling is controlled completely from within RSVP messages.

It should be noted at this point that the Gatekeeper is, and can continue to be the permissioning agent for H.323 call signaling. The bandwidth parameter that is contained in the **ARQ** is but one piece of information that the Gatekeeper may utilize when permissioning the call. It has become clear that the bandwidth indicator as seen by the Gateway is at best, an approximation. In addition, on a shared access medium such as Ethernet, the bandwidth in the **ARQ** essentially provides a QOS to all *other* applications *except* for H.323 applications. That is, the intent is to ensure that H.323 traffic does not adversely affect other traffic.

QOS Support for H.323

There are essentially two issues that need to be dealt with in order for H.323 entities to support varied QOS services on any transport. The first, is that each endpoint must be able to signal to its Gatekeeper that a transport level QOS mechanism is being utilized/requested for this particular call. Any permissioning mechanism should be contained within one logical entity. At minimum if it is a shared function, permissioning must be coordinated. In other words if there is a bandwidth ceiling set for H.323 conferencing, it should not be debited twice (Gatekeeper and a transport QOS such as SBM).

The second point, for H.323/QOS support is that both (or all, in a multipoint case) endpoints must be able to signal the other that they should attempt QOS associated with media streams with each other. The media streams of the same type (e.g. video or audio) in opposite directions will be considered logically associated by system users. (For example; users of telephone expect to send and hear the same level of speech quality) Additionally, some QOS services (RSVP being one) require a coordinated effort on behalf of the source and the destination to achieve the desired QOS (*Path/Reserve* messages)

RSVP can be utilized under H.323 to provide a higher level of quality for the conference without any additions to the protocol. However, with a few minor additional fields, a generic transport QOS service can be embraced, in a clean interoperable manner.

[3] See IETF draft titled "draft-yavatkar-sbm-ethernet-00.txt"

Endpoint to Gatekeeper Signalling

(H.225.0)

```
RegistrationRequest            ::=SEQUENCE --(RRQ)
{
        requestSeqNum  RequestSeqNum,
        protocolIdentifier ProtocolIdentifier,
        nonStandardData NonStandardParameter OPTIONAL,
        discoveryComplete      BOOLEAN,
        callSignalAddress      SEQUENCE OF TransportAddress,
        rasAddress             SEQUENCE OF TransportAddress,
        terminalType           EndpointType,
        terminalAlias          SEQUENCE OF AliasAddress OPTIONAL,
        gatekeeperIdentifier   GatekeeperIdentifier  OPTIONAL,
        ...
        transportQOS                   TransportQOS OPTIONAL[4]
}


        TransportQOS                          ::=CHOICE
        {
            endpointcontrolled                NULL,
            ...
        }
```

endpointcontrolled - indicates that the source of the ARQ is under control of a transport level QOS service for bandwidth permissioning. Specifically, the Gatekeeper should ignore the bandWidth field for admission control.

```
                RegistrationRejectReason                ::=CHOICE
        {
                discoveryRequired              NULL,  -- registration permission has aged
                invalidRevision                NULL,
                invalidCallSignalAddress NULL,
                invalidRASAddress              NULL,  -- supplied address is invalid
                duplicateAlias                 SEQUENCE OF AliasAddress,      -- alias
        registered to another endpoint
                invalidTerminalType            NULL,
                undefinedReason          NULL,
                transportNotSupported          NULL,  -- one or more of the transports
                ...
                transportQOSNotSupported       NULL,  -- Endpoint QOS not supported

        }
```

[4] Note that the absence of this field indicates that the 'normal' bandwidth permissioning should occur as described in the H.323 protocol (Revision 1).

(H.245)

```
H2250Capability                               ::=SEQUENCE
{
    maximumAudioDelayJitter                   INTEGER(0..1023),         -- units in
milliseconds
    receiveMultipointCapability               MultipointCapability,
    transmitMultipointCapability              MultipointCapability,
    receiveAndTransmitMultipointCapability    MultipointCapability,
    mcCapability                              SEQUENCE
    {
        centralizedConferenceMC               BOOLEAN,
        decentralizedConferenceMC             BOOLEAN,
        ...
    },
    rtcpVideoControlCapability                BOOLEAN,                  -- FIR and
NACK
    mediaPacketizationCapability              MediaPacketizationCapability,
    ...
    transportCapability                       TransportCapability OPTIONAL
}
QOSCapability                                 ::=SEQUENCE
{
        rsvpParams                            RSVPParams OPTIONAL,
           ...
}

TransportCapability                           ::=SEQUENCE
{
    qOSCapabilities                           SEQUENCE OF QOSCapability OPTIONAL,
    securityCapabilities                      SEQUENCE OF SecurityCapability
OPTIONAL,
    ...
}

H2250LogicalChannelParameters                 ::=SEQUENCE
{
    nonStandard                               SEQUENCE OF NonStandardParameter
OPTIONAL,
    sessionID                                 INTEGER(0..255),
    associatedSessionID                       INTEGER(1..255) OPTIONAL,
    mediaChannel                              TransportAddress OPTIONAL,
    mediaGuaranteedDelivery                   BOOLEAN OPTIONAL,
    mediaControlChannel                       TransportAddress OPTIONAL,  -- reverse
RTCP channel
    mediaControlGuaranteedDelivery            BOOLEAN OPTIONAL,
    silenceSuppression                        BOOLEAN OPTIONAL,
    destination                               TerminalLabel OPTIONAL,

    dynamicRTPPayloadType                     INTEGER(96..127) OPTIONAL,
    mediaPacketization                        CHOICE
    {
        h261aVideoPacketization               NULL,
        ...
    } OPTIONAL,
    ...
    transportCapability                       TransportCapability,
}



MiscellaneousIndication                       ::=SEQUENCE
{
    logicalChannelNumber                      LogicalChannelNumber,
```

```
    type                                CHOICE
    {
        logicalChannelActive            NULL,                   -- same as
H.230 AIA and VIA
        logicalChannelInactive          NULL,                   -- same as
H.230 AIM and VIS

        multipointConference            NULL,
        cancelMultipointConference      NULL,

        multipointZeroComm              NULL,                   -- same as
H.230 MIZ
        cancelMultipointZeroComm        NULL,                   -- same as
H.230 cancel MIZ

        multipointSecondaryStatus       NULL,                   -- same as
H.230 MIS
        cancelMultipointSecondaryStatus NULL,                   -- same as
H.230 cancel MIS

        videoIndicateReadyToActivate    NULL,                   -- same as
H.230 VIR

        videoTemporalSpatialTradeOff    INTEGER (0..31),        -- indicates
current trade-off

        ...,

        videoNotDecodedMBs              SEQUENCE
        {
          firstMB                       INTEGER (1..6336),
          numberOfMBs                   INTEGER (1..6336),
          temporalReference             INTEGER (0..255),
          ...
        }
        transportCapability             TransportCapability OPTIONAL

    },
    ...
}
```

Example: RSVP Support in Windows/IP Environment

This section will outline an example implementation of RSVP in a Microsoft Windows™ environment. This is meant to demonstrate the practicality of deployable technology in the current timeframe.

The RSVP/SBM code can be in place 'below' the WinSock 2 (WS2) interface. After opening a WS2 socket, the client will have to issue a standard control API call to WS2 to get the QOS on that particular socket. If the lower level stack (below WS2) detects that there is an SBM Server which will set policy for this endpoint - it will provide this notification to the H.323 application. The H.323 application will have to change one field, the transportQOS value in the ARQ (or RRQ) sent to its Gatekeeper, to indicate that it is 'under the control of' an SBM Server.

Gatekeepers will have to interpret the transportQOS field (if present in the ARQ) to see if it indicates whether the client is under the control of SBM or not. Depending on this field it will need to make its policy decision on whether or not to allow the H.323 call to complete. This will allow Gatekeepers to flexibly interoperate in all cases (H.323 endpoints with/without QOS client code, or in the presence/absence of an active QOS services). This also allows Gatekeepers to have not direct interactions with QOS services (if they do not want to).

The logical grouping of Gatekeeper and QOS controls might occur in the following manner. If we use the picture below as an example, then the following explanation should outline the interactions.