Telecommunication Standardization Sector AVC - 889 Intel-12

Original: English
(TSS)

Experts Group for Video Coding and Systems in ATM and Other Network Environments **January 16-19, 1996**

STUDY GROUP 15
CONTRIBUTION

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

Title: Comments for H.323 Clarification

Date: January 9, 1996

After a number of conversations with H.323 colleagues, there appears to be some further explanation needed of the following points.

Gatekeeper identifier

This octet string is used as a logical representation of the gatekeeper entity. It can (if optionally used) provide for location independence, and opaque hardware replacement.
There need not be any standardization of the format, as it is simply used as a string identifier between a gatekeeper and terminals. (This assumes that gatekeeper manufacturers have taken appropriate steps to provide for unique string identifiers)

The gatekeeper identifier may be used in the following manner. In the **GRQ** message a terminal may fill in the identifier to find the address of a particular gatekeeper. If that gatekeeper is moved to another device, no administrative overhead is taken on by any of the endpoints within its zone.
Any gatekeeper that receives a **GRQ** with its identifier knows that the terminal is asking specifically for *it* to be guardian. It may respond with a **GCF** or **GRJ** (all other gatekeepers should respond with a **GRJ** in this case). If the terminal receives only **GRJ** to this message, it should resend the **GRQ** with a null filled identifier (signifying "any gatekeeper that will have me"....)

The gatekeeper may also return its gatekeeper identifier in the **GCF** and/or the **GRJ** when responding to a **GRQ**. This allows a terminal to reuse this information in the aforementioned scenario.

Bind flag

This flag is passed in on the registration message (**RRQ**). It is assumed that in deployments with autobinding being utilized, there will be an aging process built into the binding. This will allow terminals to distribute themselves across multiple gatekeepers over time, with no administrative overhead. Assuming a binding has 'aged out' and needs to be renewed, the requesting terminal can effectively combine the **GRQ** process with the **RRQ** by setting this flag to true. If the terminal receives a **RRJ** after setting the bind flag, it shall assume that the gatekeeper is no longer its guardian.

Call Clearing (Disconnect PDU)

I believe that the current call clearing model in H.323 is overly complex and can be simplified; especially by dropping the release/release-complete messages (sec 7.5.2/3). In the case of no gatekeeper(s) present (note that the presence/absence of a gatekeeper should be taken in the context of a specific endpoint) the current text is sufficient and clear (7.5.1). The bandwidth should be explicitly returned if allocated by a gatekeeper. This could be done by sending a bandwidth request of zero, but I believe it is cleaner and more flexible to have a disconnect PDU.

If a terminal is bound to a gatekeeper and a call terminates for any reason (local end, remote end, or intermediate gatekeeper/mc) the terminal shall release all associated bandwidth by issuing a **DRQ** (disconnect request). A terminal receiving a DRQ from its guardian shall follow the normal disconnect procedure without delay.

Disconnect Request ::= SEQUENCE --(**DRQ**)
{
    ConferenceID INTEGER(1..4294967295),
    propExtension ToBeAdded
}