Real-time Transfer Protocol and Real-time Transfer Control Protocol

Emulate Hundreds of Thousands of RTP Streams controllable with RTCP with a Single Testing Product

The Real-time Transfer Protocol (RTP) and the RTP Control Protocol (RTCP) date back to January 1996 when the Audio-Video Transport Working Group of the Internet Engineering Task Force (IETC) published Request for Comment (RFC) 1889. During the mid-1990's, rapid adoption of the Internet facilitated the need for a standardized method for audio and video transmission over packet-switched networks. The IETF updated the original specification with RFC 3550.

RTP provides end-to-end network functionality designed specifically for real-time applications. Real-time applications have distinct characteristics that necessitated their own protocol. These applications are generally far more sensitive to network delay than to packet loss. A single packet lost contains only a tiny fraction of the stream which can be mitigated by various error concealment algorithms built-in to the payload types. While RTP provides the streaming data transfer mechanism, RTCP is primarily used for gathering statistics for quality of service (QoS) measurements. Both protocols generally use the User Datagram Protocol (UDP) for their transport layer protocol, with the port numbers being negotiated by some other establishment protocol, such as SIP, RTSP or H.323. Generally RTP uses an even-numbered port with RTCP making use of the next highest odd-numbered port.

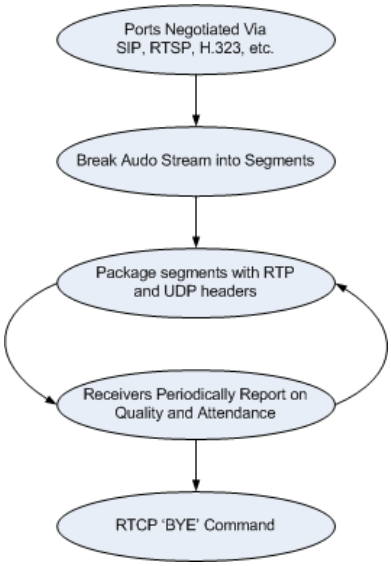
The contents of an RTP packet are extremely important to the application for media stream reconstruction. While UDP is not a reliable transport, it does provide unicast and multicast capabilities, as well as packet checksums to verify data integrity. Within the RTP header, two distinct fields are very important to the receiver of the RTP stream. A sequence number is used so the receiver is able to reconstruct the order of the packets sent by the originator. In addition to the sequence number, a type field is included which indicates the encoding type of the actual RTP payload. This allows the receiver to properly decode the incoming media stream.

Figure 1 illustrates a typical RTP and RTCP use model for a simple audio conference. Utilizing some 3rd party mechanism such as SIP, RTSP or H.323, a pair of UDP ports is negotiated for use with RTP and RTCP. The audio stream is broken into small fragments that contain singular time slices. Each of these slices is packaged with an RTP header and then into a UDP packet. Because the RTP sender may choose to change audio encodings based on RTCP feedback, the RTP header contains not only a sequence number for stream reconstruction, but also an indication of the audio/video encoding for the particular packet. Periodically, each receiver will send information back to the sender via RTCP indicating the quality of the audio being received as well as whether or not the receiver is still actually receiving. When a receiver wishes to stop receiving, it will send an RTCP 'BYE' command.

BreakingPoint Testing Tools Emulate the RTP and RTCP Protocols:

- BreakingPoint is able to generate hundreds of thousands of RTP/RTCP streams using a single, easy to use test product
- Utilizing blended applications and security, users can fully test their network infrastructure to ensure RTP/RTCP quality of service conditions are met under full network load
- Utilizing Application Manager allows users to create RTP/RTCP flows using the Session Initiation Protocol, Real Time Streaming Protocol, and H.323 to fully qualify RTP/RTCP stream establishment under both normal and abnormal protocol adherence

Figure 1 - Typical RTP and RTCP Use in an Audio Confrence



Following port establishment, audio streams are broken up into smaller segments which are encapsulated into a UDP packet with appropriate RTP headers. Periodic feedback from the receivers is sent to the sender until the RTCP 'BYE' command is issued.

© 2005 - 2010. BreakingPoint Systems, Inc. All rights reserved.
The BreakingPoint logo is a trademark of BreakingPoint Systems, Inc.
All other trademarks are the property of their respective owners.

BreakingPoint™
Find it before they do.™