Telephony Protocols

This chapter describes telephony protocols, including the Media Gateway Control Protocol (MGCP), the Session Initiation Protocol (SIP), H.323 (H.225, H.245, RAS), the Skinny Client Control Protocol (SCCP), and TDM/analog signaling protocols.

Skinny Client Control Protocol

The Skinny Client Control Protocol (SCCP) is a Cisco proprietary master/slave protocol that CallManager and other call agents can use to communicate with devices and endpoints such as Cisco IP phones. One advantage of SCCP is that it is flexible and allows features to be easily added.

SCCP uses TCP ports 2000 (SCCP) and 2443 (Secure SCCP [SCCPS]).

Call agent, device, and endpoint types that can communicate using SCCP include CallManager, CallManager Express, Cisco Unity (voice mail ports), routers running SRST, VG200 (DSP-farm), VG224, VG248, ATA186, ATA188, Cisco IP phones, Cisco IP Communicator, Tandberg and Sony video endpoints, Cisco 3800 gateways, Cisco 2800 gateways, Cisco 3700 gateways (DSP farm), Cisco 3640/3660 gateways (DSP farm), Cisco 2600/2600XM gateways (DSP farm), Cisco 1751/1760 (conferencing/transcoding), and WS-X6608-T1/E1.

CallManager uses SCCP to control media resources such as transcoding resources, media termination point (MTP) resources, music-on-hold (MoH) resources, and annunciator resources. These media resources types are discussed in Chapter 1.

RTP and cRTP

The Real Time Transport Protocol (RTP, RFC 3550) is used to transport voice and video media packets across a network. RTP uses UDP as its transport protocol.

RTP uses UDP rather than TCP as its transport layer protocol mainly because TCP retransmission is not suitable (it would take too long), TCP congestion control/slow start mechanisms can lead to an inability to ensure sufficient packet rates, and TCP headers are larger than UDP headers. Finally, it is worth mentioning that RTP/UDP is preferred over TCP for voice media transport because TCP does not provide required encoding and timestamp information.

RTP uses sequence numbers and timestamps to detect packet loss and ensure correct playout timing. RTP timestamps for each session begin with a random number, so time information provided by RTP is relative. (Absolute time, such as that provided by Network Time Protocol [NTP], is not required to run RTP.)

The Real Time Control Protocol (RTCP) provides out-of-band reporting of QoS statistics for RTP flows, including information relating to packet loss, jitter, and round-trip time. Although NTP is not required to run RTP, RTCP can make use of NTP.

The IP/UDP/RTP headers that encapsulate voice media packets total 40 bytes, and this overhead is significant when you consider that the voice media payload may, for example, be 20 bytes (G.729). So, IP/UDP/RTP headers make up a large proportion of total voice media packet size, and on slow speed links it can be advantageous to compress these headers to save bandwidth.

Compression of the IP/UDP/RTP headers can be achieved using Compressed RTP (cRTP). According to RFC 2508 and 3545, the 40 bytes of IP/UDP/RTP headers can typically be compressed down to 2 to 4 bytes. The compressed header is 2 bytes when no UDP checksums are sent (the checksum is zero), and 4 bytes when (nonzero) checksums are sent.

The following sample calculation illustrates bandwidth savings that can be realized when using cRTP:

Using the following formula:

Bandwidth = (Layer 2 header + Layer 3 header + Layer 4 header + voice payload in bytes) * 8 bits * packets per second

So, assuming the G.711 codec (default 160 byte payload), with MLP (1 + 6 bytes overhead), at 50 pps:

Without cRTP:

Bandwidth = ([1 + 6] + 20 + 8 + 12 + 160) = 207 bytes * 8 = 1656 * 50 pps = 82,800 bps = 82.8 kbps

With cRTP:

Bandwidth = ([1 + 6] + 2 + 160) = 169 bytes * 8 = 1352 bits * 50 pps = 67,600 bps = 67.6 kbps

Bandwidth savings:

82,800 – 67,600 = 15,200 bps

(15,200 / 82,800) * 100 = 18.36 percent savings

Note that when calculating voice bandwidth, it is important to take into account the following factors:

Layer 2 headers/overhead:

MLP = 6 bytes

Frame Relay/FRF.12 = 6 bytes

ATM = 5 bytes

MLP over Frame Relay = 14 bytes

Ethernet = 18 bytes, including CRC

Also, add 1 byte for the end-of-frame flag for MLP and Frame Relay frames.

Layer 3 and 4 headers:

IP header = 20 bytes

UDP header = 8 bytes

RTP header = 12 bytes

cRTP (IP/UDP/RTP) header = 2 (or 4) bytes

Sample sizes:

G.711 uses a sample size of 80 bytes/10ms and has a default voice payload size of 160 bytes/20ms (default payload includes 2 samples)

G.729 uses a sample size of 10 bytes/10ms and has a default voice payload size of 20 bytes/20ms (default payload includes 2 samples)

Media Gateway Control Protocol

Media Gateway Control Protocol (MGCP) is defined in RFC 3435 (which obsoletes RFC 2705), and it specifies an application programming interface and text-based master/slave protocol used by a media gateway controller (MGC) or Call Agent to control media gateways (MG). MGCP is based on two other (now obsolete) protocols, the Simple Gateway Control Protocol (SGCP) and Internet Protocol Device Control (IPDC).

MGCP MGCs and MGs can be described as follows:

■ **MGC/Call Agent:** This element possesses call control intelligence and controls MGs. An MGC/Call Agent could, for example, be a Cisco CallManager.

■ **MG:** This device provides translation between data packets and audio signals received over VoIP networks and other networks, such as the PSTN. A media gateway could, for example, be an IOS router with analog or digital voice ports.

Media gateways can be classified depending on the connectivity they provide. For example, a media gateway that terminates trunks connecting to the telephone network can be referred to as a *trunking gateway*, and a media gateway that provides analog connections to phones can be referred to as a *residential gateway*.

MGCP specifies a connection model involving endpoints and connections:

■ **Endpoint:** MGCP media gateways contain endpoints, which are sources/destinations for data. Endpoints can be physical, such as interfaces terminating trunks connecting to a PSTN or interfaces terminating POTS connections to PBXs, key systems, or telephones. Endpoints can also be virtual endpoints, such as audio content sourced from a server.

■ **Connection:** This is an association between endpoints for the purpose of transmitting data, and can be either point-to-point or multipoint in nature.

MGCs/Call Agents and MGs use several commands and responses (or verbs) to communicate with each other:

■ **EndpointConfiguration (EPCF):** A Call Agent sends this message to a gateway to specify signal encoding that will be received by an endpoint. This message could, for example, be used to specify whether audio calls will be encoded using mu-law or a-law.

■ **CreateConnection (CRCX):** This command creates a connection between two endpoints. The connection is created based on parameters included with the command, such as codec, allowable bandwidth, use of echo cancellation, silence suppression, gain control, and so on.

■ **ModifyConnection (MDCX):** This is used to modify the parameters associated with a connection that was previously created.