Fundamentals Series
SIP
Welcome to SIP, the sixth module in the Polycom Fundamentals series. This module is approximately 8 minutes long.
In order to understand how videoconferencing works it’s important to understand the underlying technologies at work behind the scenes. In this short module we will talk more about Session Initiation Protocol and how it works.

As a quick recap, SIP is a different beast than the ITU umbrella standards as it is an IETF signaling protocol specifically designed for IP networks. SIP was designed to work with other protocols already in existence, and is a text based protocol (which we shall see a little later on). Because of this it can interact with other IP standards to perform tasks, and this is part of why it doesn’t have to be an umbrella standard to control the process of video and audio sessions on the network.
SIP was designed in 1996 as a way for multimedia devices to connect to each other on an IP network. Unlike an umbrella standard which defines how devices have to handle the multimedia data once a session is connected, SIP is primarily involved in the signaling portion of the communication session.

So, for videoconferencing, SIP is involved in setting up and tearing down the calls and exchanging capabilities data but it doesn’t define what those capabilities have to be. It does, however encompasses a registration process to allow alias dialing, which we will look at later also.

To carry media, SIP uses two protocols we have not discussed yet. These protocols are Real-time Transport Protocol, or RTP, and RTP Control Protocol, or RTCP, and it is these we shall look at first.
Real-time Transport Protocol (RTP) is responsible for the transport of the audio and video in our SIP videoconference. RTP is also used extensively in other communication and entertainment systems that involve streaming media, such as telephony and television services.

RTP is used in conjunction with RTP Control Protocol (RTCP). While RTP has the responsibility of carrying the media, RTCP is used to monitor transmission statistics and other control information.
The protocol used by SIP to perform the capability exchange during call setup is called Session Description Protocol. SDP allows the endpoints to negotiate parameters for the call including video and audio codecs, port usage and line rate. Because SIP isn’t an umbrella standard the endpoints are free to use whatever video and audio codecs are available, but of course they have to match each other or the call will fail.

SDP is the mechanism for them to determine which video and audio they have in common; most manufacturers use the same codecs for SIP and H.323, which makes compatibility easier between the formats when making a SIP to H.323 call using a gateway.

Once the endpoints determine they can interact, SDP steps back until it’s time to make a change or to end the call.
Like many IP based protocols, SIP uses basic text messages to initiate processes and do the work. There are several messages that SIP uses but here are the most common messages and some possible responses. You will notice that the messages are quite easy to decipher, and are described as requests and responses.

Endpoint 1 sends an INVITE request to endpoint 2 using TCP or UDP port 5060. Endpoint 2 replies with a Trying response which tells endpoint 1 that endpoint 2 is processing the request, and sends a Ringing message back so a ringtone is heard and the end-user knows the call is trying to connect. When endpoint 2 answers, an OK message goes through and the call goes ahead once endpoint 1 sends an ACK message to confirm the session.

When the call is over, endpoint 2 sends a BYE request to endpoint 1. Endpoint 1 then sends an OK message to let endpoint 2 finalise the end of the call.

The numbers associated with the responses are fairly simple too – a code starting with 1 means that the response is provisional. A code starting with 2 means that the response is successful.
SIP and H.323 are similar in that they both work on the IP network, and because of that some of the same processes used for H.323 calling can be applied to SIP.

The idea of having an alias for a device is very prevalent in the IP world, and in order to achieve that we need a server designated to handle that registration information. When using H.323, this device is the gatekeeper. When using SIP, this device is known as the SIP registrar; it registers devices and looks up the IP addresses from the alias just like our gatekeeper does.

However, unlike H.323, when we’re using SIP there is only one alias available to us. This is called the SIP URI (or Uniform Resource Identifier). The URI is an IETF standard that defines how to create an alias for TCP/IP networks. The URI is familiar to most people today, as it takes on the form of an IP address, that is, alias@domain.domain suffix.
A proxy server passes on SIP messages and makes requests on the behalf of SIP clients, for example SIP endpoints.

This is primarily to allow the correct routing of messages across the network. Proxy servers are also in charge of enforcing any policies, such as bandwidth or access restrictions. So it can also perform some of the functions of an H.323 gatekeeper.

SIP also allows for a device called a Redirect Server that forwards messages intended for one URI to another URI or set of URIs. This is useful when you need to forward messages outside of one domain into another; the Redirect Server can make sure your requests get forwarded to the correct registrar to find the destination address to complete your call connection.
SIP can support a content channel just like H.323 or H.320. By using the SDP messages to tell the far side to do so and using a protocol called Binary Floor Control Protocol (or BFCP) to handle who is presenting, a content channel can be opened and closed during a SIP call by doing a re-invite with a new SDP containing two video channels. Again SIP, SDP and BFCP don’t dictate what media formats are used in the call, they just handle the messages and process of opening a second channel and providing control of it during the call.
This module provides a simple overview of the SIP process which enables a call to be made. You should be able to see how the information we have covered in the previous module or two comes together here, enabling us to understand the flow behind a successful audio or video call.
Thank You