Welcome to H.323, the fifth module in the Polycom Fundamentals series. This module is approximately 9 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 H.323 and how it works.

As a quick recap, H.323 is an umbrella standard which comes under the ITU. As an umbrella standard, H.323 takes care of other related standards such as call control and audio, video and data processing.
The H.323 umbrella has under it four main areas which we will look at here. Video and audio we have already covered; that's the first two done.

The third area is data, which in this context means content sharing – the data sent across the network which allows us to share a computer desktop or a DVD.

The fourth is control, which is really how you make a call. Control governs the call speed, the protocols used, among other things. This is by far the most complex of these areas, so after a quick exploration of the data aspect, we'll move onto a deeper look at control.

There are many protocols associated with H.323, however we shall only look at a couple here to provide an overview.
H.239 is an ITU protocol which sits under the H.323 umbrella. H.239 defines rules and messages for establishing an additional video/graphics channel, often to transmit a PC graphics presentation or video from a document camera, while still transmitting the video of the presenter.

When sharing content in a multipoint conference, H.239 also defines procedures to guarantee that only one endpoint in the conference sends the additional video channel which is then distributed to all conference participants.

Prior to the H.239 standard being available, Polycom wrote a proprietary standard called People+Content. These two standards are not the same, though some Polycom products still support People+Content.
So now let’s move onto the concept of control. When we want to have a videoconference, what exactly has to happen in order for this to occur? Well, the first thing that we need is to set up the call, and this is where H.225 comes in. H.225 provides a number of services which enable a call to be set up, connected and ended gracefully.

The first aspect of H.225 we shall cover is framing. When I speak, you can interpret when one word ends and another one begins by my use of spaces and punctuation. Framing is how an endpoint understands how the incoming data should be interpreted.

To explain a little further, data moving through a network has to be split into small sections to be quickly and efficiently routed around the network. The actual data is packaged up inside a frame; the frame enables other information to be added to it, such as the source and destination address. The frame is also able to mark the start of where the data begins, and can also provide more sophisticated information such as the function of the message sent. In this way, the endpoints can understand the incoming data and interpret it correctly.
To make a call, we must first complete call setup. The call setup aspect of H.225 is performed using a protocol called Q.931 which was originally created for H.320, and here’s how it looks.

Endpoint 1 sends a SETUP request to endpoint 2 using TCP port 1720. Endpoint 2 replies with a CALL PROCEEDING message which tells endpoint 1 that endpoint 2 is processing the request, and sends an ALERTING message back so a ringtone is heard and the end-user knows the call is trying to connect. When endpoint 2 answers, the final CONNECT message goes through and the call goes ahead.

When the call is over, endpoint 1 sends a DISCONNECT message to endpoint 2. Endpoint 2 then sends a RELEASE to let endpoint 1 know that the channels are closing down, and endpoint 1 sends a RELEASE COMPLETE to finalise the end of the call.
RAS is an abbreviation for registration, admission and status, and is also a function undertaken by H.225 when a gatekeeper is in use. RAS, as the name suggests, is how the gatekeeper manages the endpoint registrations and allows or denies access to the network, though we still use H.225 to set up a call in the same way once the gatekeeper has done its stuff.

RAS messages are usually sent using TCP port 1719 and are one of three types, a request, a confirmation or a rejection. From there what looks like a long list of possible messages becomes quite a bit shorter. The following are commonly seen messages:

RRQ, which is a registration request. An endpoint sends this to a gatekeeper to ask if it can register. The gatekeeper will send back either an RCF to confirm, or an RRJ, to reject. When you make a call using an endpoint which is registered to a gatekeeper, your endpoint will send an ARQ, which is an admission request – this asks your gatekeeper to make a call. It will try and find the endpoint you are dialling and reply with either an ACF or an ARJ. When you hang up the call, your endpoint sends a DRQ, which is a disengage request. Your gatekeeper will then send a DCF or a DRJ to confirm or deny the request. If the gatekeeper sends the DRQ, then all endpoints must return a DCF and end the call.

There are more, but for now this is enough for you to get the idea.
So, whether you are using a gatekeeper or not, once your call is connected another protocol, H.245, takes over and handles our capabilities exchange (which we will look at next), and opens ports on either end of the call. If these UDP ports are not opened up for the media then there will be no audio and video in our conference call.

Once the call is in progress and media is flowing, H.245 remains part of the call and handles our control and indication functions. For example, an endpoint can request a picture refresh, where the whole picture is re-sent, and in a multipoint call H.245 is also used to request chairperson rights, amongst other functions.

H.245 is also the message protocol required to close the logical channels at the end of the call. The H.245 End Session command needs to be sent before the endpoints will stop sending H.245. Only once this is done will the Q.931 disconnect message be used to finish the call tear down.
The capabilities exchange is the part of the call setup where the endpoints determine what audio, video and data capabilities they have in common. This will decide the video and audio codecs to be used and which bandwidth both will connect at. For example, if I try to call you at 512kbps, my endpoint might tell your endpoint that it wants to connect using H.264 with G.722.1c. If your endpoint can do this, the call will go ahead, but let’s say it has been bandwidth restricted to 384kbps. In that case, your endpoint will tell mine that it can only do 384kbps, mind will confirm it can do that too, and the call can start based on those variables.

From there, once this has all been decided, the endpoints will decide which is the ‘master’ and which is the ‘slave’ during the call. This is necessary to ensure that in the event of a change being made, one endpoint is able to overrule the other so a decision can occur.

Once this is done, the call setup is complete, and media flow can begin.
This module provides a simple overview of the H.323 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