- Basic Voice over IP -

Voice over IP (VoIP)

Voice over IP (VoIP) is a digital form of transport for voice transmissions, replacing analog phone systems.

The benefits of VoIP are considerable:

- **Better use of bandwidth -** Traditional voice requires a dedicated 64-Kbps circuit for each voice call, while VoIP calls can use considerably less. Additionally, *no* bandwidth is consumed when no call is being made.
- **Single form of cabling –** Reduces implementation and maintenance costs by having a standardized and consolidated cabling and equipment infrastructure.
- **Cost savings from integration into the data network –** Toll charges for inter-office voice communication can be avoided by routing voice traffic across existing data lines.
- **Integration into devices beyond telephones**

Basic VoIP components can include:

- **Phones –** including both analog and IP phones.
- **Gateways –** allows a non-VoIP (analog) device to communicate with the VoIP network, or a VoIP device to communicate with an analog network.
- **Application Servers –** provides required applications to VoIP phones.
- **Gatekeepers –** maps phone numbers to IP addresses, and grants permission for call setup
- **Call Agents** – handles call routing and setup.

Digital Signal Processors (DSP's) are used by devices to perform analog-to-digital and digital-to-analog conversions. Both VoIP phones and gateways utilize DSP technology.

VoIP Packetization

Voice traffic must be **packetized** as it traverses the IP network. Sound is first captured using a microphone on the headset**.** A voice call requires a 4 kHz (4000 Hz) channel. To convert analog voice to a digital format, **samples** of the frequency and amplitude of the analog wave are made. Thus, **sampling** merely takes a snapshot of the signal at a given point in time.

The *amplitude height* of each snapshot is assigned a numeric value, through a process called **quantization.** This numeric value is then represented as a sequence of binary digits (usually 8) through a process called **encoding.**

The **Nyquist sampling theorem** dictates that the analog wave should be sampled at a rate of twice the channel's frequency range:

$$f_s = 2(\text{freq. range})$$

Thus, assuming a range of 4000 Hz, this requires a rate of 8000 samples per second. Remember that each sample is assigned an 8-bit value to represent the amplitude height at the time of sampling. Thus, a dedicated 64,000-bit channel (8-bits x 8000 samples per second) was traditionally required for a voice call (hence a DS0 being 64Kbps).

The process of encoding an analog signal into digital format is handled by a **codec** (coder-decoder). The codec usually provides a level of **compression.** The efficiency of the compression varies with the codec used; however, more compression generally degrades sound quality. Various codecs include:

- **G.711 –** uses 64 Kbps for a voice call
- **G.726 –** uses 32, 24, or 16 Kbps for a voice call
- **G.728** – uses 16 Kbps for a voice call
- **G.729 –** uses 8 Kbps for a voice call

Generally, the analog sound is chopped into *groups* of **10ms**, and then sampled and encoded. Each group (or often two groups, for a total of **20ms** of analog sound) is **encapsulated** within an IP packet. At the transport layer, **Real-Time Protocol (RTP)** is used instead of TCP. RTP operates on top of UDP.

When the voice packet arrives at a digital-to-analog gateway, the headers are stripped off, and the sound is reassembled as an analog stream.

Cisco VoIP Integration

Cisco devices operating as VoIP gateways can contain a variety of *analog* interfaces, including:

- **Foreign Exchange Station (FXS) interface** – connects to an analog device, providing the appropriate voltage and dial tone.
- **Foreign Exchange Office (FXO) interface** – connects to a PBX (Private Branch Exchange) or PSTN (Public Switched Telephone network).
- **E&M interface** – can also be used to connect to a PBX, or is used for PBX-to-PBX connections.

Additionally, Cisco gateways can connect to provider PBX's and networks using *digital* interfaces, including:

- **ISDN BRI and PRI**
- **T1/E1 CCS (Common Channel Signaling) –** employs a dedicated channel for signaling.
- **T1/E1 CAS (Channel Associated Signaling) –** a portion of each channel is utilized for signaling.

VoIP Signaling Protocols

VoIP protocols are responsible for the three key *stages* of a voice call:

- Call setup
- Call maintenance
- Call teardown

The most common VoIP protocols are as follows:

- H.323 – an ITU standard
- Session Initiation Protocol (SIP) – an IETF standard
- Media Gateway Control Protocol (MGCP) – an IETF standard
- Skinny Client Control Protocol (SCCP) – Cisco proprietary

Cisco VoIP CallManager Configuration – Voice Trunk Ports

Traditionally, Cisco IP phones contain a switch with two interfaces. The first interface connects the IP phone to the wall jack. The second interface connects the user's workstation to the IP phone. This allows a single cable to handle the user's voice and data needs.

To keep the voice/data traffic segregated, the IP phone forms a **trunk link** with the remote switch. Data traffic is tagged as a different VLAN than the voice traffic. Configuration on the remote switch (or the call-manager functioning as the switch) is simple:

VoIP-Switch(config)# *interface FastEthernet0/1/4*
VoIP-Switch(config-if)# *switchport access vlan 50*
VoIP-Switch(config-if)# *switchport trunk native vlan 50*
VoIP-Switch(config-if)# *switchport mode trunk*
VoIP-Switch(config-if)# *switchport voice vlan 60*

In the above example, data traffic will be tagged as *VLAN 50*, while voice traffic will be tagged as *VLAN 60*.

Cisco VoIP CallManager Configuration – Dial Peers

Dial Peers provide call-routing, and serve two key functions:

- **VoIP dial-peers** - used to connect Cisco call-managers, gateways, or gatekeepers to other such VoIP devices. For example, two call-managers at separate branches would point to each other using VoIP dial-peer commands.
- **POTS dial-peers** – used to connect Cisco VoIP devices to an analog device or network. A dial-string is mapped to a *local* analog port on the VoIP gateway or call-manager.

Thus, the function of a Dial Peer is to match an *incoming* call with a *destination pattern,* which points to either a remote device or local interface.

To configure a VoIP dial-peer:

CallManager(config)# *dial-peer voice 1 voip*
CallManager(config-dial-peer)# *session procotol sipv2*
CallManager(config-dial-peer)# *session target ipv4:10.1.5.50*
CallManager(config-dial-peer)# *destination-pattern 15865551212*
CallManager(config-dial-peer)# *codec g711ulaw*

The above configuration maps a *sip* connection to a remote VoIP peer at address *10.1.5.50* for phone number *15865551212*. The *g711* codec is being employed.

To configure a POTS dial-peer:

CallManager(config)# *dial-peer voice 2 pots*
CallManager(config-dial-peer)# *destination-pattern 1212*
CallManager(config-dial-peer)# *port 0/2/0*

The above configuration maps an extension or phone number of *1212* to the analog voice port *0/2/0.*

(Reference: http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/vvfax_c/int_c/dpeer_c/dp_ovrvw.htm)

Cisco VoIP CallManager Configuration – Telephony Service

Telephony-Service configuration provides a wide variety of global configuration options for a Cisco CallManger:

> **CallManager(config)#** *telephony-service*
> **CallManager(config-telephony-service)#**

The configuration files for *specific* models of IP phones are stored in flash, with a *.bin* extension. To load these configuration files:

> **CallManager(config-telephony-service)#** *load 7914 S00105000200*
> **CallManager(config-telephony-service)#** *load 7920 7920.4.0-02-00*
> **CallManager(config-telephony-service)#** *load 7960-7940 P00308000400*

To specify the maximum number of phones that can register with the Call-Manager (dependent on the hardware/software platform):

> **CallManager(config-telephony-service)#** *max-ephones 12*

To specify the maximum number of directory numbers (DNs) the Call-Manager will support (also dependent on the hardware/software platform):

> **CallManager(config-telephony-service)#** *max-dn 48*

To specify the IP address of the Call-Manager on the voice VLAN:

> **CallManager(config-telephony-service)#** *ip source-address 10.5.5.1 port 2000*

To specify the extension for voicemail:

> **CallManager(config-telephony-service)#** *voicemail 2000*

To specify the audio file for music-on-hold:

> **CallManager(config-telephony-service)#** *moh music-on-hold.au*

To configure a username and password for the Call Manager's web interface, and to enable configuration of DN's through that interface:

> **CallManager(config-telephony-service)#** *web admin system name AARON password CISCO*
> **CallManager(config-telephony-service)#** *dn-webedit*

To access the web-interface, use the following URL:

http://*IPADDRESS*/telephony_service.html

Cisco VoIP CallManager Configuration – Telephony Service (continued)

To allow the transferring of calls to an outside line, using a specific dial-pattern (such as dialing 9 first, and then a seven-digit number):

CallManager(config-telephony-service)# *transfer-pattern 9 . . . .*

To configure the auto-attendant for night-service:

CallManager(config-telephony-service)# *night-service code *11*
CallManager(config-telephony-service)# *night-service day Mon 18:00 06:00*
CallManager(config-telephony-service)# *night-service day Tue 18:00 06:00*
CallManager(config-telephony-service)# *night-service day Wed 18:00 06:00*
CallManager(config-telephony-service)# *night-service day Thu 18:00 06:00*
CallManager(config-telephony-service)# *night-service day Fri 18:00 06:00*

To configure a directory of extensions:

CallManager(config-telephony-service)# *directory entry 1 3000 name Aaron*
CallManager(config-telephony-service)# *directory entry 1 3001 name Petey*
CallManager(config-telephony-service)# *directory entry 1 3002 name Team Awesome*
CallManager(config-telephony-service)# *directory entry 1 3003 name Team Tiger*
CallManager(config-telephony-service)# *directory entry 1 3003 name Jack Nicholson*
CallManager(config-telephony-service)# *directory entry 1 3004 name Nick Cage*

To define the URL's for ephones:

CallManager(config-telephony-service)# *url directories http://10.5.5.1/localdirectory*
CallManager(config-telephony-service)# *url services http://10.5.5.1/menu.php*
CallManager(config-telephony-service)# *url authentication http://10.5.5.1/auth.php*

Some configuration changes require a reset of the phone(s). To reset all phones connected to the Call Manager:

CallManager(config-telephony-service)# *reset all*

(Reference: http://www.cisco.com/en/US/tech/tk1077/technologies_configuration_example09186a00800ffdcc.shtml)

Cisco VoIP CallManager Configuration – DN's and Extensions

Directory Numbers (DNs) are assigned to phones for identification and to allow call-routing. Call Managers usually support a finite number of DNs, depending on the hardware/software/licensing platform.

Extensions are then *mapped* to these DNs. The Call Manager identifies the phone using a DN, but users *call* a phone using the extension number. To configure a DN:

CallManager(config)# *ephone-dn 5 dual-line*
CallManager(config-ephone-dn)# *number 3001*
CallManager(config-ephone-dn)# *description Call at your own risk*
CallManager(config-ephone-dn)# *name Petey*

In the above example, *dn 5* is configured as a *dual-line,* which allows for call transfer, conferencing, and call waiting. DNs/extensions that serve solely as a voicemail box can leave off the *dual-line* parameter.

An extension number of *3001* has been assigned to this DN.

(Reference: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122limit/122z/122zj15/cme30cr/icr30am.htm#wp1008071)

Cisco VoIP CallManager Configuration – Phones

To configure the actual VoIP phone:

CallManager(config)# *ephone 7*
CallManager(config-ephone)# *mac-address 0011.2233.4455*
CallManager(config-ephone)# *type 7920*
CallManager(config-ephone)# *button 1:5*
CallManager(config-ephone)# *pin 12345*
CallManager(config-ephone)# *speed-dial 1 3002 label "Team Awesome"*

In the above example, *ephone 7* has been identified as having a *mac-address* of *0011.2233.4455*, and a model number of *7920*.

The *button* command maps the first button (or extension on the phone – Cisco phones support multiple extensions) with *DN* number *5* (per our previous configuration, this maps to extension *3001*).

Finally, the phone's pin number for voice mail has been set to *12345,* and a *speed-dial* entry has been added for extension *3002*.

(Feference: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122limit/122z/122zj15/cme30cr/icr30am.htm#wp1014674)