The Ultimate ONT Study Package

Chris Bryant, CCIE #12933 **http://www.thebryantadvantage.com/**

Back To Index

Voice Over IP (VoIP)

Overview

An Overview Of Voice Networks

Even if you **have** been living in the proverbial cave, you just might have heard about a little something called "voice over IP", or VoIP. While the ONT exam will not require you to configure VoIP, you will need to know the basic concepts and the (many) devices that work together to make VoIP transmission possible.

VoIP is a major step forward over previous Voice services, as you'll see throughout this section. Remember how ISDN BRI uses a 64-kbps circuit for a single call? Those days are *gone.*

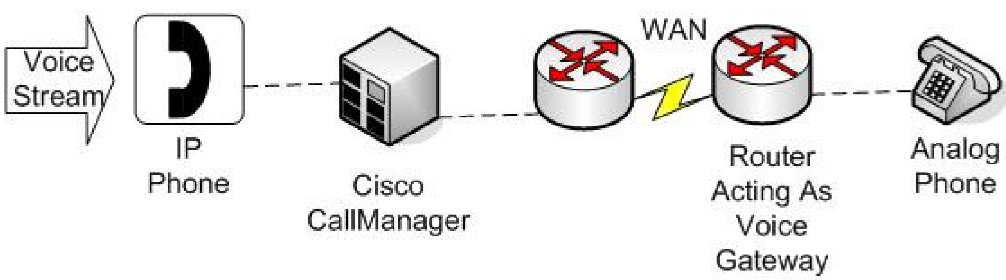
Once you've completed your CCNP, I strongly urge you to learn more about VoIP and to consider getting Voice certified. Cisco now offers a CCNA Voice certification that is an excellent starting point for those of you interested in the Cisco Certified Voice Professional (CCVP) certification, and the knowledge you acquire during your studies for those certifications will make you highly valuable in today's job market - and tomorrow's!

Even if you have no interest in the CCVP at present, I would definitely take a look at becoming CCNA Voice certified.

Now back to our CCNP studies!

This section is going to prepare you for ONT exam success as well as give you a foundation for future Voice studies. We'll take a look at the (many) different factors involved in a successful Voice deployment, and throw in a few new acronyms while we're at it!

Let's take a look at the overall steps in a typical VoIP deployment.



First and foremost are the phones themselves! That's obvious enough, but what might not be as obvious is that not all phones on our telephony network are going to be IP phones - some are going to be good ol' fashioned *analog phones*.

Those analog phones are going to require a *gateway* to run properly on our telephony network, and as you'll soon see, Cisco routers make great gateways. The gateway is the termination point for the local media, and also the point at which the digital-to-analog and analog-to-digital signal conversions take place.

The gateway may also be a *gatekeeper.* Cisco's website mentions three specific benefits regarding gatekeepers on their website:

- Gatekeepers allow VoIP networks to become more scalable, since changes can be made at the central location, the gatekeeper itself.

- Gatekeepers allow the use of a proxy to keep VoIP calls separate from data traffic and handle VoIP signaling as well.

- Gatekeepers allow the configuration of *Call Admission Control* (CAC) to limit the number of simultaneous IP calls on the network.

- Gatekeepers allow for greater management of bandwidth and the creation of dial plans.

- Gatekeepers perform the actual phone number-to-IP address conversion that VoIP calls require

We'll talk more about CAC later in this section, but note that CAC does *not* limit the bandwidth assigned to a specific call - it can be used to place a ceiling on the number of *overall* simultaneous calls.

Let's take a look at the different kinds of analog and digital interfaces. Some of the digital interfaces and their values will be familiar to you from your CCNA studies.

Analog Interface Types

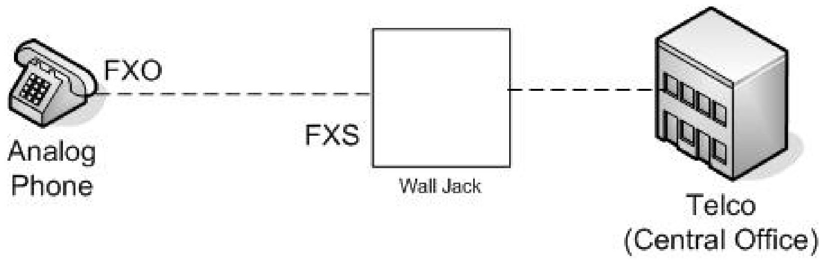
Cisco routers support three analog interface types:

- FXS, the *Foreign eXchange Service*
- FXO, the *Foreign eXchange Office*
- E&M, which stands for *Earth & Magneto* or *Ear & Mouth*, depending on who you talk to, so look for either or both on your exam and in real-world documentation.

It's vital to keep FXS and FXO straight, and someone gave me a good way to remember the major difference between the two. If I remembered who it was, I'd credit him/her, but here's one way to remember which interface type does what:

- The FX**O** port points to the **O**ffice
- The FX**S** port points to the **S**ubscriber (customer)

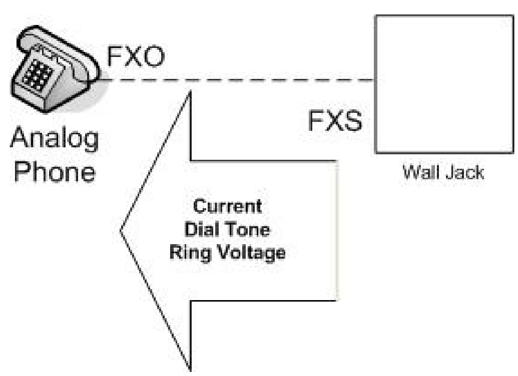
If we have a phone plugged straight into a wall jack, here's where the ports are:



The FXS port will be connected directly to an analog phone or a fax machine, and it supplies vital information to that device, including voltage and dial tone.

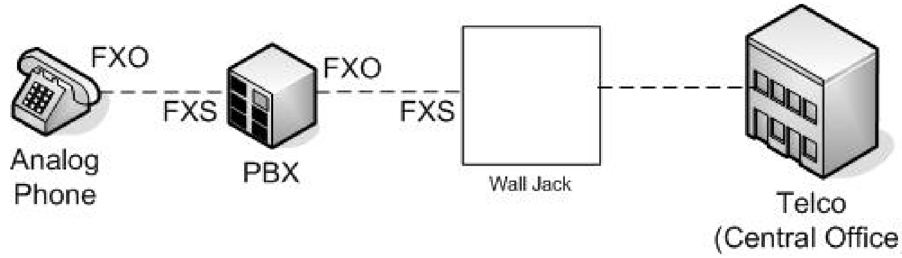
In your CCNA studies, you learned that we directly connect routers with a DTE/DCE cable, and the DCE end must supply clockrate to the DTE end. I mention that because here, the Foreign eXchange Subscriber (FXS) interface must supply *three* things to the Foreign eXchange Office (FXO) end of the connection:

- line current
- dial tone
- ring voltage



By the way, the phone shown in these examples could be a fax machine or a modem - any device that needs a dial tone to operate.

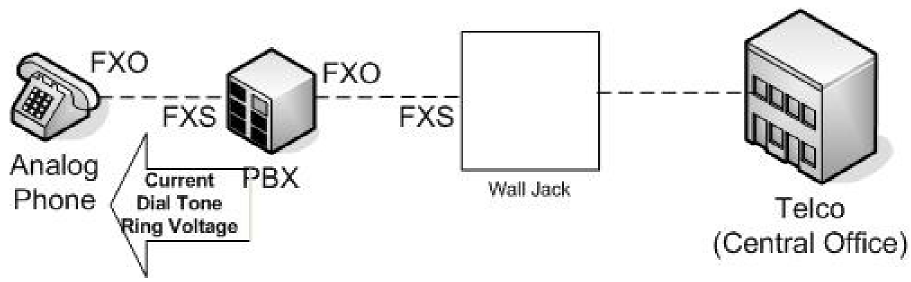
As we add devices to that basic setup, things can get a little more complicated. *Private Branch eXchanges* (PBX) are basically private enterprise phone systems. PBXs can save a company quite a bit of cash, since calls from one enterprise end user to another are handled by the local lines rather than external phone lines. When a PBX is introduced into our little network, we've got extra FXO and FXS ports to be aware of.



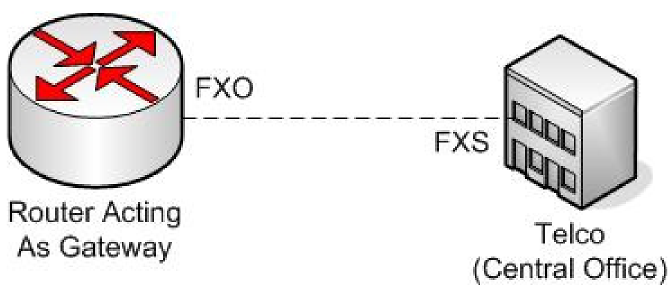
We do have more FXO and FXS ports, but the rules remain the same:

- FXO ports point toward the **O**ffice
- FXS ports point toward the **S**ubscriber (customer)

With that topology, it's the *PBX* that will supply the current, ring voltage, and dial tone.



When a Cisco router serves as a gateway, the router will have both FXS and FXO interfaces. The FXO interface will connect a gateway to a Central Office.



Digital Interface Types And Number Of Channels

Before we look at the digital interfaces supported by Cisco routers, let's review an important term from your CCNA ISDN studies. *Signaling* is used by telephone devices for several important tasks:

- Setup and teardown of calls
- Tracking of connect / disconnect times for accounting

- Dynamically adjusting to changing network conditions

As with any Cisco exam, it's a good idea to know the number of channels and the bandwidth used for signaling.

BRI: Two 64kbps channels for voice and regular data, uses 16kbps for signaling (remember the D-channel?)

T1 CAS (*Channel Associated Signaling*): 24 16-kbps channels, in-band signaling

T1 CCS (*Common Channel Signaling*): 23 16-kbps channels, uses 64kbps for signaling (again, a D-channel)

E1 CAS: 30 64-kbps channels, 64 kbps for signaling

E1 CCS: 30 64-kbps channels, 64 kbps for signaling

With ISDN BRI, we have two separate b-channels, each with a capacity of 64 kbps. The BRI channels are dedicated channels; when a single call is placed, it does not share any of that channel's capacity. VoIP calls do not use such a dedicated circuit - one reason we have to be so concerned with QoS for voice packets is that the voice packets share bandwidth with regular data packets.

VoIP Signaling Protocols

The overall stages of a VoIP call are just like the stages of an ISDN call:

- Setup (Call Routing process)
- Maintenance
- Teardown

That's pretty much where the similarities end! By the way, the Call Routing process during the Setup phase is the point at which the dialer number must be mapped to the appropriate IP address. The Setup stage can also involve *Call Admission Control (CAC)*, which is an optional but (in my humble opinion) vital step toward a healthy VoIP network.

In ISDN, we left signaling to the D-channel, and that was that. With VoIP, however, we've got three separate protocols that can be used for signaling:

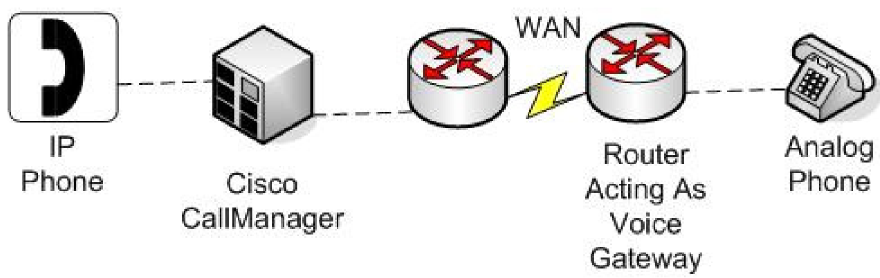
- *H.323*, the International Telecommunication Union standard (ITU)
- *Media Gateway Control Protocol* (MGCP), an IETF standard
- *Session Initiation Protocol* (SIP), another IETF standard

Beyond the ITU/IETF difference, there's a more critical difference:

- MGCP is a *centralized* call control signaling protocol
- H.323 and SIP are *distributed* call control signaling protocols

I'll remind you of that difference when we discuss VoIP design models later in this section.

Let's take a wide-angle look at one possible VoIP WAN topology.



We've got an analog phone at one side of the WAN, so we must have a gateway - and as I mentioned earlier, Cisco routers make great voice gateways. Note also that one side of the WAN has a Cisco CallManager, and we'll discuss the CallManager's role later in this section.

Without concerning ourselves with the specific hardware yet, it's obvious that we've got two different conversions in that network:

- Analog signals must be translated to digital signals to allow transmission to the remote end of the connection
- Digital signals must be translated back to analog by that remote end to allow the analog phone to understand the signals

It's a good idea to understand the steps in each process. It's pretty simple, but note that there are some required steps and some optional steps. In the following lists, if a step is not indicated as optional, it's mandatory.

Analog-to-digital steps:

- *Sample* the analog signal
- *Quantize* that sample
- *Encode* the signal
- *Compress* the samples (optional, helps to conserve bandwidth)

Sampling occurs at the "microphone level" of the entire conversation, since samples are captured as the sounds themselves are spoken into the microphone. The sampling process in turn creates a Pulse Amplitude Modulation (PAM) signal. Before we talk about the next step, let's discuss the Nyquist Theorem. (Don't worry, it sounds complicated, but it's quite simple.)

Sampling And The Nyquist Theorem

You'll often hear the Nyquist Theorem called simply "the sampling theorem", and there's a good reason why. There's also a quick formula involved, and we all know these formulas have a funny way of showing up on Cisco exams!

We need a guideline for how many of these samples to take, and the Nyquist Theorem is that guideline. According to this theorem, the sampling rate should be twice as high as the highest frequency of the signal for the signal to be accurately rebuilt at the destination.

This is one of those formulas that sounds complicated, but really isn't. Let's assume the highest frequency to be transmitted is 3000 Hertz (Hz). Double that, you get 6000. The sampling rate is 6000 samples per second.

As another example, a common highest frequency value is 4000 Hz. Double that, you've got 8000 samples per second.

Simple, right? :) That shouldn't take much practice, but do be familiar with how the sampling rate is determined.

Quantization, Encoding, and Compression

Now back to the analog-to-digital process! Let's review the basic steps..

Analog-to-digital steps:

- *Sample* the analog signal
- *Quantize* that sample
- *Encode* the signal
- *Compress* the samples (*optional,* helps to conserve bandwidth)

The *quantization* process involves the PAM signals mentioned earlier. Each PAM signal is assigned a value in accordance with its amplitude. ("amplitude" is the measurement of degree of change from an average, which is more than you need to know about it to pass the ONT exam.)

During your CCNA studies, you learned that PRI offers a different number of channels in Japan than it does in the United States. In a similar vein, quantization techniques differ according to geographic location:

- *Linear,* primarily used in the United States
- *Logarithmic*, primarily used everywhere else

Encoding is much more straightforward. All we're doing there is taking the quantization result and putting it into - you guessed it - binary! However, this wouldn't be networking if we didn't have options, and those options come in when it comes to which CODEC (COder / DECoder) to use. Here are some common CODECs along with their bit rate and MOS - *Mean Opinion Score.*

G.711 - 64 kbps bit rate, MOS 4.1 (highest MOS)

G.729 - 8 kbps bit rate, MOS 3.92

G.729A - 8 kbps bit rate, MOS 3.90

G.726 - 32, 24, or 16 kbps; MOS 3.85

The process for determining a CODEC's MOS is an interesting one - a group of people listen to a sound and then give it a grade from 1 - 5, with 5 being "excellent" and 1, well, *isn't* excellent. Any score over 4 is considered a high-quality CODEC.

The other end of the process is straightforward:

Digital-to-analog steps:

- *Decompress* the signal (*optional*, only needed if the signal was compressed to begin with)
- *Decode and filter* - *Decode* the signal from binary, which rebuilds the PAM signal, and then *filter* unwanted noise from the signal.
- *Reconstruct* the analog signal from the PAM signal.

Compression

Compression is optional, but as you'll see later in this section, it's important to consider configuring compression. (Which is easier than *saying* "consider configuring compression".)

While UDP provides multiplexing of voice traffic, it's the Real-Time Protocol (RTP) that actually delivers voice traffic from source to destination. I'm bringing that up because one of our compression techniques is Real-Time Protocol Header Compression, which I'll discuss later in this section and configure elsewhere in the course.

Packeting Voice For Transport

The somewhat awkward term "packetization" is used to describe the process of preparing voice streams for transmission across the WAN by - you guessed it - placing the voice data into packets.

Again, we've got two conversions that have to take place:

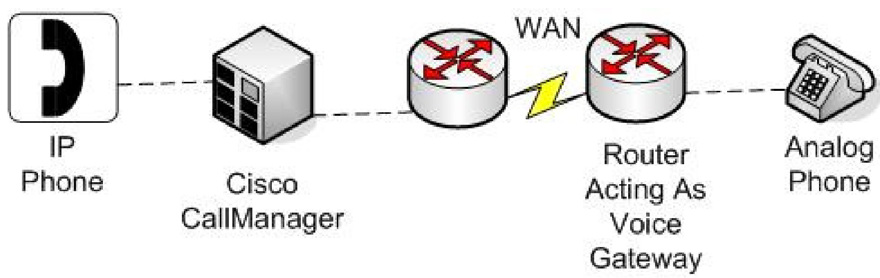
- First, the voice data has to be placed into packets.
- Once the packets have reached their destination, the voice data has to be converted back to its original format - voice!

The terms *packetization period* and *packet rate* are used to measure two separate processes here. They're often confused, so make sure *you* don't confuse them!

Packetization period is the actual amount of voice that's encapsulated in each packet. Measured in milliseconds, the normal packetization period is 20 ms. If the packetization period is shortened, the resulting packets are smaller; if this period is lengthened, the packets will be larger.

Packet rate is simply the number of packets sent in a given time period, usually one second ("packets per second", or PPS). The larger the packets, the lower the packet rate.

To break this process down a bit, let's use the VoIP setup from earlier in this section.



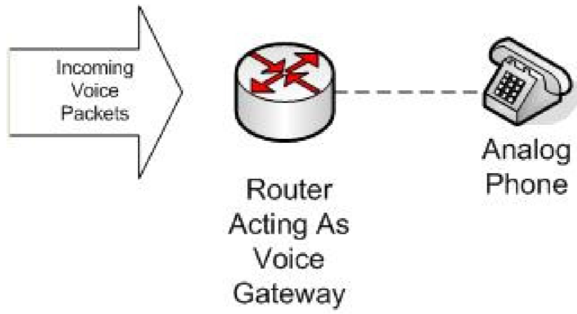
There are two issues to be aware of, one on the IP Phone side and the other on the analog phone side.

Overhead can be a real issue on the transmitting side. Before voice packets can be transmitted, several headers are going to be added to every packet:

- Data link header - Ethernet will add 18 bytes overall
- IP header - 20 bytes
- UDP header - 8 bytes
- RTP header - 12 bytes

That starts to add up, and since voice traffic is more delay-sensitive than any other traffic, we've got to pay extra attention to keeping overhead down. That's an excellent argument for configuring RTP header compression.

The issue on the receiving end has to do with the router receiving the voice packets and the device that needs to receive a voice stream - the analog phone.



While the router certainly understands what a packet is, the analog phone doesn't know a packet from a hole in the ground. That's a problem. :) It's not enough for the router to forward voice packets to the analog phone; the router has to convert the incoming packets into signals that the analog phone can understand -- namely, an analog signal!

We have to have a *gateway* that can perform this signal translation, and in this case, that will be the Cisco router itself.

In addition to FXO and FXS, Cisco routers can support a third type of analog connector, E&M. Depending on who you ask, E&M stands for...

- Earth & Magneto
- Ear & Mouth

... so be prepared to see E&M referred to by either of those terms on your ONT exam, as well as any books you read in your future Voice studies.

When we have a Cisco router acting as a voice gateway, the router performs several tasks in addition to the above:

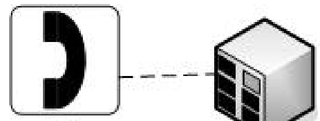
- Converts analog signals to digital signals in preparation for encapsulation and transmission

- Performs RTP Header Compression (cRTP)

- Allows redundancy for IP Phones via *Survivable Remote Site Telephony,* which we'll also discuss later in this section

Cisco CallManager

On the IP phone side of our network, we have a Cisco CallManager.



Throughout this section, I'll mention "CallManager cluster" several times. You can configure a CM stand-alone device, but more than likely you'll have the workload spread out over multiple machines (the cluster). Clustering also delivers - let's say it together! - *redundancy*!

CallManager makes a network admin's life a lot easier, and here are just few reasons why:

- CM has accounting capabilities to allow you to keep track of who's making calls, when they successfully connected, and how long the call lasted. CM can also create reports based on this information.
- CM can detect when call quality deteriorates and can dynamically adjust resources to prevent further deterioration.
- CM will check if resources are available for a call before the actual call is placed (during the call setup phase).

Call Admission Control

As I've mentioned 1000 times already, voice traffic is the most delay-sensitive traffic we deal with, and there's no middle ground - a voice stream is good or it's bad (a lot of jitter). Frankly, if the resources aren't available for a high-quality Voice transmission, we'd rather have no transmission than a poor-quality one!

CallManager's *Call Admission Control* allows you to configure a threshold for concurrent VoIP calls, and when that threshold is reached, no more calls will be placed until one of those concurrent calls ends. Using CAC ensures that the overall WAN resources aren't overwhelmed by voice calls.

You're not required to configure CAC in a VOIP network, but it's a *really* good idea. Let's say you have enough bandwidth for eight high-quality calls. You figure that's great, since you only have twelve staff members and they're not all in the office at one time - but then one day they are all in the office at once, and by an amazing coincidence, they all place a call. (If this sounds like a true story, that's because it is!)

It's not just the "extra" calls that will be affected - *all* of the calls will be low-quality calls. If you had configured CAC to allow eight calls, those eight calls will be high-quality and the other calls will simply be prevented until one of those eight calls is terminated. If you have the choice between making someone wait for a high-quality call or giving them an on-demand low-quality call, you're a lot better off making them wait.

CAC is actually made possible by our old friend RSVP - or more accurately, the *Cisco RSVP Agent*. The RSVP Agent is signaling-protocol independent, so there's no problem with running SIP, MGCP, or H.323.

Automated Alternate Routing (AAR)

When CallManager decides that there isn't enough bandwidth for a high-quality call, AAR can step in and make the call. AAR can actually reroute the call to the PSTN *without* user intervention - in fact, the user doesn't even have to hang up!

AAR isn't a default behavior of the CallManager, though. You've got to create *AAR groups*, which isn't difficult, but it does require some time and a little research first. Configuration of AAR groups is beyond the scope of the ONT exam.

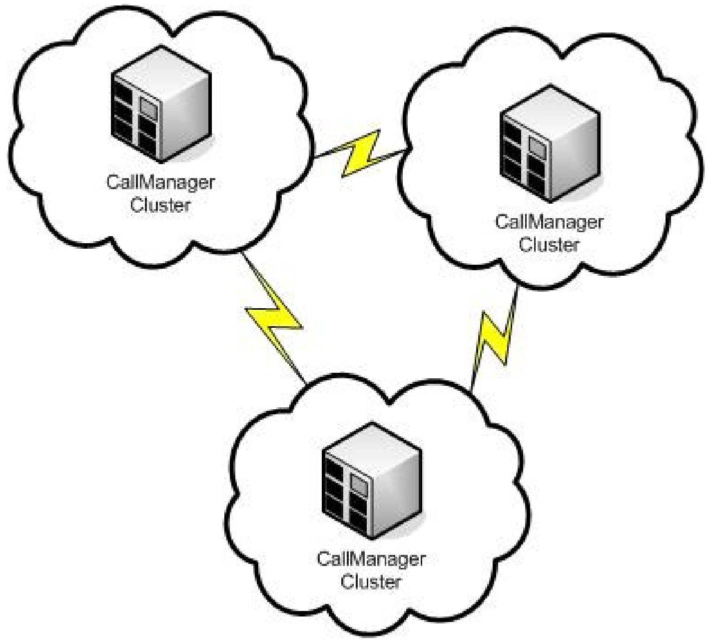
CallManager Deployment Models

There are four basic models for deploying CallManager:

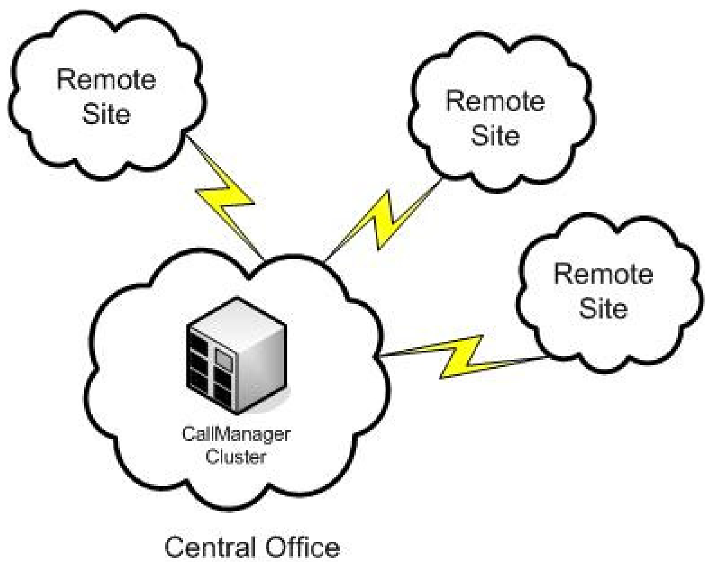
- Single-Site ("one-site")
- Multisite with Distributed Call Processing
- Multisite with Centralized Call Processing
- Clustering Over WAN

Single-site is just what it sounds like - there's one site, and there is a cluster of CallManagers at that site.

With *multisite with distributed call processing,* each site has a CallManager cluster of its own, so the call processing is distributed between sites. This allows each site to handle its own local calls with its own CallManager cluster, and every site can handle the setup and teardown of calls on its own. (This model is sometimes referred to simply as the *distributed call processing model* or *distributed call control model*.)



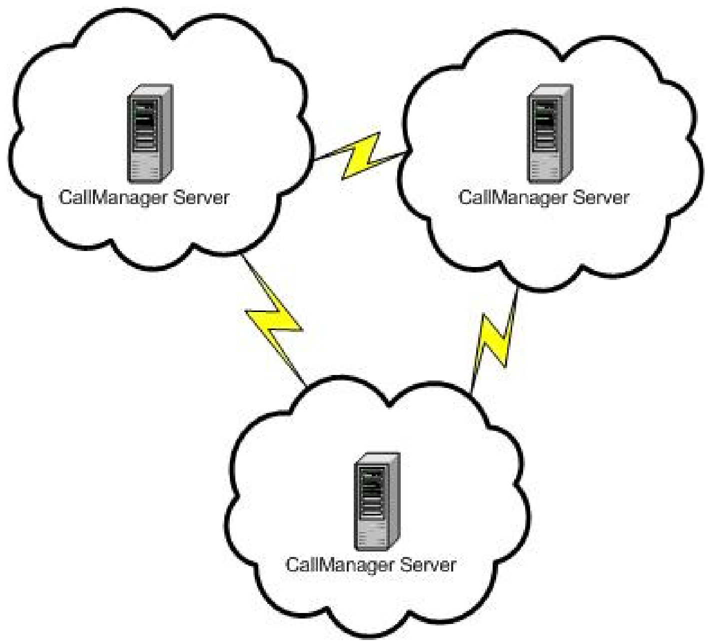
Multisite with centralized call processing has a single CallManager cluster at one site handling both local and remote calls. This is a typical setup when there's one central office and multiple spoke sites. Each remote site will use a *call agent* to communicate with the central location.



Of those two models, I personally prefer the distributed model. With the centralized model, you've got a single point of failure and all of the signaling traffic has to cross the WAN. The single point of failure and all of that traffic crossing the WAN are two good reasons to use other models.

Finally, the *clustering over WAN model* will consist of one CallManager cluster, but the CM servers themselves are spread out over the corporate

WAN.   This model is probably the trickiest of the four to configure.

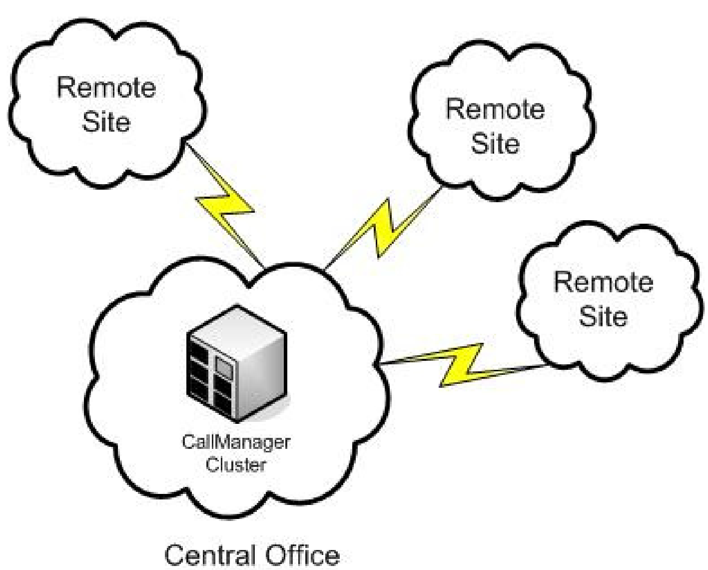


At the beginning of this section, I mentioned a major difference between the three VoIP signaling protocols. Since we just discussed centralized and distributed models, this is a good spot for a reminder!

- MCGP is a *centralized* call control signaling protocol
- H.323 and SIP are *distributed* call control signaling protocols

The Centralized Call Model And Redundancy

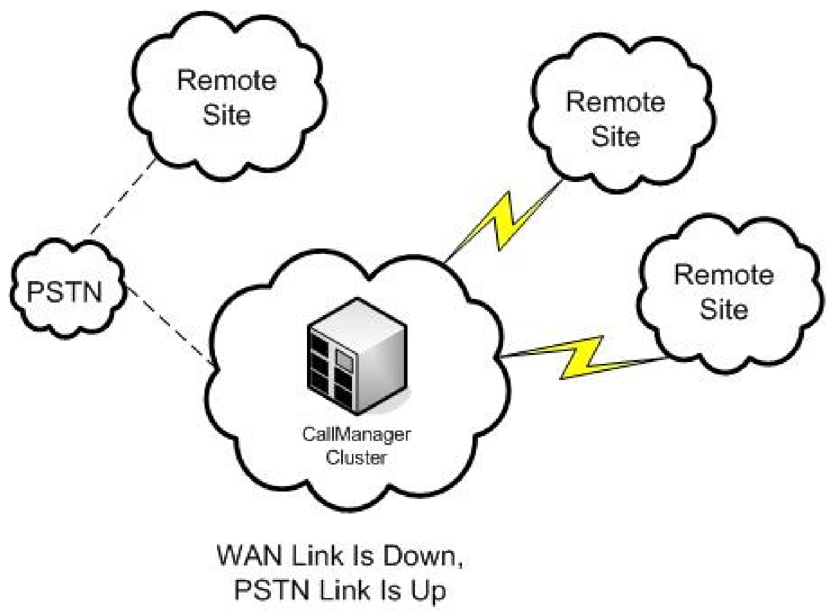
Let's take another look at the multisite centralized call model.



The word "centralized" can also mean "single point of failure", and that's what we have here as far as the remote sites are concerned. The IP Phones at the remote sites will register with the CallManager cluster via the WAN - but what if the WAN link from a remote site goes down? The IP Phones are then cut off from the CallManager cluster.

We can avoid this single point of failure with the aptly-named *Survivable Remote Site Telephony.* SRST allows a Cisco router to take the place of the now-unavailable CallManager cluster. While the WAN is down, the connection to the Public Switched Telephone Network (PSTN) remains, and this allows the IP Phones at that remote site to function in the absence of the CallManager cluster.

The Cisco router can use a T1 or E1 trunk port for this connection to the PSTN; it can also use an FXO port.



Cisco Unified CallManager (And The Express Version)

Cisco's website promotes *Unified CallManager* (now known as *Unified Communications Manager*) as "scalable, distributable, and highly available". A great combination! CallManager does just what it sounds like, and more - CM will handle signaling, call processing, call routing, and more!

In many typical networks, though, not every spoke site is going to need Unified CallManager. That's where the "express" version comes in!

Cisco Unified CallManager Express

Most *Integrated Services Routers* (ISRs) can run CUCE, which basically allows the Cisco router to act as a CallManager. To quote Cisco's website, CUCE is "best suited for customers who are looking for an integrated, reliable, feature-rich telephony system up to 240 users".

In a multisite centralized configuration, it's common to use CUCE at the remote sites to act as the call agent.

- ***Overhead, Overhead, And More Overhead***

Telephony networks certainly have their benefits, but one drawback is the increased overhead. Here's some of the "basic" overhead we have to deal with:

- Data link header - Ethernet will add 18 bytes overall, dot1q will add 22, Frame Relay will add 6 bytes
- IP header - 20 bytes
- UDP header - 8 bytes
- RTP header - 12 bytes

And don't forget that tunneling can and will add even more overhead! The additional overhead from GRE, MPLS and IPSec varies, but it can be substantial. For example, using IPSec Tunnel mode can actually result in more than *doubling* the size of the packet, since the voice packet itself will be encapsulated into another packet! (The other IPSec mode, Transport mode, doesn't add that much overhead - that mode results in only an extra 20 bytes of overhead.)

Naturally, the more overhead we have, the greater the chance of delay - and that's what we're trying to avoid when it comes to voice transmission!

We can't eliminate overhead, but we can lower it with RTP Header Compression (cRTP). cRTP is configured at the interface level on a link-by-link basis:

```
R1(config-if)#ip rtp header-compression ?
 iphc-format  Compress using IPHC format
 passive      Compress only for destinations which send compressed headers
  <cr>
```

On an interface running frame relay, the command is a bit different:

```
R1(config)#int serial0
R1(config-if)#encap frame
R1(config-if)#frame-relay ip rtp header-compression
```

The router won't allow you to mix those two commands up, but the exam might. :)

cRTP actually works at both L3 and L4 of the OSI model, since it will compress the 40-byte IP/RTP/UDP headers. The degree of compression depends on whether UDP checksums are in use, but either way, the overhead savings are impressive.

- UDP checksums in use? 40-byte header compressed to 4 bytes
- UDP checksums not in use? 40-byte header compressed to 2 bytes

For proper two-way cRTP compression, this must be configured on both ends of the link.

Calculating Bandwidth Needed For A VoIP Call

So how much bandwidth does a VoIP call need? This one isn't quite as easy as the Nyquist Theorem! The formula for calculating BW needed for a specific VoIP call depends on several factors:

- Packetization period
- Packet rate
- Overhead (IP/UDP/RTP)
- L2 Overhead (Ethernet, frame, etc.)
- Tunneling overhead (if used)
- CODEC in use

Voice Activity Detection

VAD does just what it sounds like it does - it detects voice activity, and perhaps more importantly, it detects when there is no voice activity. When there is no talking on a call using VAD, no audio packets are transmitted. Studies show that a normal conversation is 35 - 40% silence, VAD can give us big-time bandwidth savings as well as cutting down on overall traffic.

This being networking, you just know there's gotta be a catch, and there is. If you're on your cell phone and you suddenly hear silence, you may think that you've been cut off and hang up on a live call. (After saying "Can you hear me now?" a couple of times, that is. Also, from personal experience I can tell you that the remote party's speech can sound clipped, almost as if the first syllable of every sentence is cut off.

Comfort noise can help prevent this. Comfort noise is artificial noise that's "pumped in" when the parties involved in the call go silent. It's very low, but it is there, which prevents you from thinking you've been cut off - and again, from personal experience, comfort noise really does make the conversation more comfortable.

Comfort noise is not on by default on a Cisco router, but it's easy enough to enable:

```
voice-port 2/0/0
   comfort-noise
```

Let's take a look at a basic dial plan configuration. The configs we'll look at here are very straightforward, but they commonly require troubleshooting due to a misconfiguration. Let's get you ready to troubleshoot these plans in the exam room and on the job!

Dial Plans and Dial Peers

Basically, a dial plan is the phone number or numeric pattern the router will dial to reach a destination, and the dial peer is the actual configuration on the router. Cisco's website compares dial peers to static routes, and that's an excellent comparison - the dial peer basically says "here's where we need to get, and how we're going to get there".

Dial peers can be more detailed than the very basic ones we're going to look at here, but most of the misconfigurations I've seen happen right at this point. We can configure POTS dial peers and VOIP dial peers, and the configs are just a bit different - but it's an important difference in the exam room and the real world!

Here's a typical POTS dial peer:

```
dial-peer voice 1 pots
      destination-pattern 5555555
      port 0/0/1
```

If you took the CCNA when ISDN was on that exam, you know that on occasion network admins configure the dialer map command with the local router's phone number rather than the remote router's number. The reason I bring that up is that on occasion, admins will put the local router's number in for the destination pattern rather than the remote router's phone number.

Remember - the command is *destination*-pattern.

Assuming R1's phone number is 6666666, and R2's is 5555555, R1's POTS dial peer would look like it does in the previous example.

You can configure a single destination-pattern command that matches more than one phone number. There are several wildcard symbols available with the *destination-pattern* command, but the most common is the simple period. The period is a single-digit wildcard. For instance, the destination-pattern 5555... would match any seven-digit number beginning with 5555.

The *destination-pattern* command is used with both POTS and VOIP dial peers, but not so the *port* command. That command is used with POTS dial peers, not VOIP peers. Trying to configure a VOIP peer with the port command is a common error, so keep an eye out for that one in the exam room and the real world!

What does the VOIP dial peer use, then? Glad you asked! Here's an example of a simple VOIP dial peer:

```
dial-peer voice 1 voip

    destination-pattern 5555....

    session-target ipv4:210.1.1.1
```

The IP address specified in the *session-target* command is - repeat after me - the IP address of the remote router that you want to call. Don't configure any of the local router's IP addresses here!

To sum up these basics...

> When configuring POTS dial peers, use the destination-pattern and port commands. The *destination-pattern* command references the remote device's phone number. The *port* command references the local router's voice port.
>
> When configuring VOIP dial peers, use the *destination-pattern* and se*ssion-target* commands. Both command reference a value on the remote router.

Hot Spots And Gotchas

Call Admission Control (CAC)

When too many active Voice calls are present, that creates a general degradation of all calls, not just the "extra" ones. QoS is a great tool and an important one on the ONT exam, but you can't use QoS to limit the actual number of calls.

We can use Call Admission Control, though! With CAC, we can set a limit on the number of simultaneous active calls, which prevents that overall call degradation.

We don't want to place a call and then have CAC cancel it - that wouldn't be practical. Therefore, CAC runs during the Call Setup phase. Calls that would not be permitted by CAC are never allowed in the first place.

CAC uses the RSVP Agent to do its job.

Automated Alternate Routing (AAR) can work in tandem with CAC to reroute calls that CAC would otherwise deny. Those calls will be sent through the good ol' reliable PSTN.

Dial Peers

These are fairly simple configs, but be ready to troubleshoot them, since it's kinda easy to get them backwards. So don't get them backwards!

```
dial-peer voice 1 pots

    destination-pattern 1111

    port 1/0/0
```

OR

```
dial-peer voice 1 voip

    destination-pattern 1111

    session-target ipv4:210.1.1.1
```

About the only thing these two configs have in common is the *destination-pattern* command. Remember, just like *dialer-map* commands, the number in *destination-pattern* is the remote host's number, not the local host. After all, there's no reason to place a call to yourself!

For POTS calls, use the *port* command.

For VOIP calls, you'll use the *session-target* command.

The Nyquist Theorem

According to this theorem, the sampling rate should be twice as high as the highest frequency of the signal for the signal to be accurately rebuilt at the destination.

This is one of those formulas that sounds complicated, but really isn't. Let's assume the highest frequency to be transmitted is 3000 Hertz (Hz). Double that, you get 6000. The sampling rate is 6000 samples per second.

Watch your Hz vs. KHz on the exam and when reading documentation. 12000 Hz = 12 KHz.

This 'n' That

Of the Mean Opinion Scores discussed in this section, the highest belongs to g.711, with a bit rate of 64kbps and a MOS of 4.1.

Voice packet overhead adds up quickly, since we've got an IP header, a UDP header, and an RTP header.

Our VOIP signaling protocols: H.323, SIP, and MGCP.

In contrast, CAS and CCS are used to signal the PSTN.

The FXO interface points to the Office - and if the FXO port is on a router, that's just what it connects to - the router serving as a gateway

- We can use RTP Header Compression (cRTP) to fight the overhead issue. The exact amount of compression of the IP/RTP/UDP header can vary...

    If UDP checksums are in use, compressed to 4 bytes

    If UDP checksums are not in use, compressed to 2 bytes

... but it's certainly better than 40-byte headers.

So why are we so concerned about overhead with voice packets?

While the overhead can add up quickly, the voice packets themselves are pretty small. We need to keep them small, since the human ear can begin to hear a problem with the voice delivery if the latency reaches anywhere from *100 - 150 milliseconds.* And when the latency gets worse than that, so does the end user's experience.

Speaking of overhead, be wary when running IPSec in Tunnel mode. As you'll see in the Tunneling section, tunnel mode carries quite a bit of extra overhead - enough overhead to nearly *double* the size of a Voice packet, and that's not exactly a way to speed a packet's delivery!

The gateway in a voice network is the local media termination point and the point at which the analog-to-digital and digital-to-analog conversoins we discussed take place.

If a Cisco router's serving as the gateway, the gateway has more capabilities - and responsibilities - that the two mentioned previously. *Survivable Remote Site Telephony* (SRST) allows us to survive a WAN failure by allowing the gateway to act as the call agent.

We don't have to have an emergency to have a Cisco router act as a call agent, though. The Cisco Unified Callmanager Express (CUCE) makes it possible for a Cisco router to fill that role permanently, with the router acting as a Cisco Unified Callmanager.

We're always wary of lists, so be sure to know the different factors that come into play when calculating the bandwidth of a VOIP call.

- Packetization period
- Packet rate
- Overhead (IP/UDP/RTP)
- L2 Overhead (Ethernet, frame, etc.)
- Tunneling overhead (if used)

- CODEC in use

- Another list, this one of available channels and their capacity:

  - BRI: Two 64kbps channels for voice and regular data, uses 16kbps for signaling (remember the D-channel?)
  - T1 CAS (*Channel Associated Signaling*): 24 16-kbps channels, in-band signaling
  - T1 CCS (*Common Channel Signaling*): 23 16-kbps channels, uses 64kbps for signaling (again, a D-channel)
  - E1 CAS: 30 64-kbps channels, 64 kbps for signaling
  - E1 CCS: 30 64-kbps channels, 64 kbps for signaling

Analog-to-digital conversion steps:

Sample > Quantize > Encode > (Optional) Compression

Digital-to-analog conversion steps:

Decompress (if header was compressed) > Decode > Reconstruct

Keep the terms "packetization period" and "packet rate" straight:

Packetization period is the actual amount of voice that's encapsulated in each packet. Measured in milliseconds, the normal packetization period is 20 ms. If the packetization period is shortened, the resulting packets are smaller; if this period is lengthened, the packets will be larger.

Packet rate is simply the number of packets sent in a given time period, usually one second ("packets per second", or PPS). The larger the packets, the lower the packet rate.

Other Terms

Voice Activity Detection (VAD) - VAD also detects when there is *no* voice activity, in which case it suppresses the transmission of voice packets. Since the average phone call is 25 - 35% silent, this saves a lot of bandwidth.