Cisco Systems

CCVP CIPT Quick Reference Sheets

Kevin Wallace

shortcut

Your Short Cut to Knowledge

ciscopress.com

The Cisco CallManager

Introduction

In years past, voice, video, and data traffic resided on their own physical networks. However, the vision for the future, and today, is for voice, video, and data to reside on the same network. Having such a "converged" solution provides numerous economic benefits, including lower administrative costs. However, connectivity between converged networks and legacy Private Branch Exchanges (PBXs) and the Public Switched Telephony Network (PSTN) is a necessity in your design.

In this section, you examine a central component in your IP telephony network: the Cisco CallManager (CCM). You examine how the CCM fits into the Cisco Architecture for Voice, Video, and Integrated Data (AVVID) strategy. CCM hardware platforms and deployment models are also examined, in addition to CCM installation and upgrading procedures.

AVIDD

Cisco's AVVID solution brings voice, video, and data communication technologies under a common umbrella. Specifically, Cisco's AVVID model contains the following four layers, as it relates to voice:

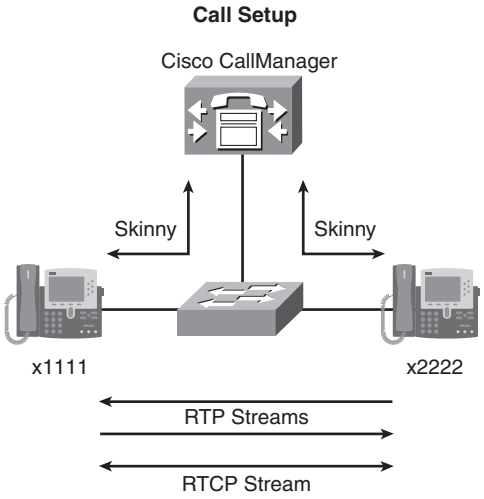
- **Client**—The client layer encompasses the devices that interface with the network. For example, IP phones, PCs, and Cisco IP SoftPhones fall under this layer.
- **Application**—The application layer includes applications that enhance the services that are available to the converged network. For example, Cisco's Unity application provides converged messaging solutions and would be categorized under the application layer.
- **Call-processing**—The call-processing layer performs the call setup functions, such as those that traditionally are performed by the PBX. The Cisco CallManager, for example, resides at the call-processing layer.
- **Infrastructure**—The infrastructure layer is the underlying data network, which includes components such as routers and switches.

The primary focus of these quick reference sheets is the Cisco CallManager.

Cisco CallManager

The Cisco CallManager (CCM) is a software component that runs on approved server platforms. The CCM makes call-forwarding decisions, controls IP phones, and can support other optional features (for example, automatic route selection and call transfer).

Consider the following scenario. A Cisco IP Phone goes off-hook and communicates with a CCM using the Skinny protocol. The IP phone dials digits, which the CCM interprets. The CCM then signals to the destination IP phone that the phone is receiving a call. After the destination phone goes off-hook, a Real-Time Transport Protocol (RTP) stream is set up directly between the IP phones.



The CCM is administered through the Cisco CallManager Administration web interface. If you know the IP address or domain name of a CCM server, you can access the CCM administration interface at http://*server_IP_or_name*/ccmadmin.

Throughout these quick reference sheets, you should spend quite a bit of time exploring the menus that are available under this interface, including the System, Route Plan, Service, Feature, Device, and User menus.

As mentioned earlier, the CCM is software. Therefore, it needs to run on a server. Specifically, version 3.3 of the CCM software runs on a Windows 2000 server. The CCM needs to maintain database information to store information such as IP phone registrations. Instead of trying to create its own proprietary database, Cisco decided to use a well-established database application, Microsoft SQL Server 2000.

User information is stored in a *directory*. You could use the DC Directory that comes with the CCM software. Alternatively, you could leverage Microsoft Active Directory Service. For disaster recovery, you also need some backup application. As an example, you can use the Spirian Technologies Inc. (STI) Backup utility to back up the CCM database.

IP Telephony Design Considerations

The cost savings that an IP telephony environment offers are numerous. For example, an IP telephony solution usually lends itself to reduced staffing expenses, reduced wiring costs, and reduced administrative costs for functions such as moves, adds, and changes. In addition to cost savings, IP telephony solutions also enhance the functionality of your overall telephony solution by providing value-added feature such as extension mobility and a consolidated messaging system.

Although the CCM 3.3 software is an application that runs on a Windows 2000 server, the server hardware must be a specific platform that identifies Cisco. Although many of these approved servers are Cisco branded, the hardware often is manufactured by IBM or Compaq.

The following table details several characteristics for comparison.

Server Model	RAM (MB)	Hard Drive(s)	Number of IP Phones Supported	Rack Mount Information
Cisco ICS SPE 310	512	One 20.4-GB HD	Up to 500	N/A (The server is a module that installs in an ICS 7750.)
Cisco MCS 7815-1000	512	One 20.4-GB HD	Up to 200	Tower case with rack-mount kit available.
Cisco MCS 7825-1133	1024	One 40-GB HD	Up to 1000	1 RU (Rack Unit).
Cisco MCS 7835-1266	1024	Dual 18.2-GB HDs with hardware RAID	Up to 2500	2 RUs.
Cisco MCS 7845-1400	2048	Four 72-MB HDs with hardware RAID	Up to 7500	2 RUs.

The amount of "work" that a CCM can do is measured in *device units*. Each device that registers with the CCM requires one or more of these device units, but the number of device units supported can vary depending on the number of calls that the CCM must support simultaneously. This call load is referred to as Busy Hour Call Attempts (BHCAs), which is the maximum number of active calls during the busiest hour of the day.

To achieve the level of availability that is found in a PBX environment, you need to have more than one CCM at your disposal. Groupings of CCMs are called *clusters*. One CCM in a cluster is designated as a *Publisher*, and that CCM replicates its database information to *Subscribers*. This database replication includes information such as registration and data logging.

When a configuration change is made, the change is written to the Publisher. You still can access the CCM web interface, even if the Publisher is down. However, you cannot write changes to the database. When a Publisher is again available, it immediately sends configuration changes to Subscribers.

The Publisher's database is replicated to Subscribers periodically, but under normal operation, the Subscribers still retrieve information from the Publisher's database. If the Publisher becomes unavailable, Subscribers use their local copy of the database.

The CCMs in a cluster also communicate run-time data directly to each other, using a logical full-mesh topology. This run-time data includes information such as calls in progress, gateway and IP phone registration, and information about digital signal processor (DSP) resources.

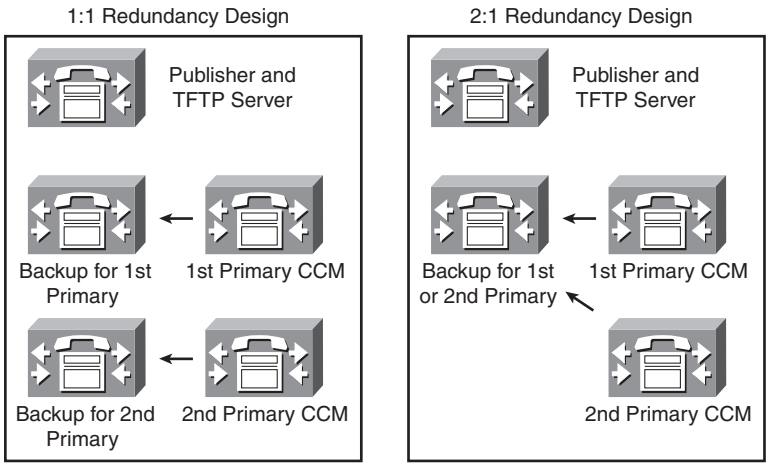
As an example of run-time data, consider an IP phone that is registering with a CCM. The CCM lets all other CCMs in the cluster know about the registration. Then, the IP phone sends a keep-alive message to its primary CCM, which it registered with, every 30 seconds. For redundancy, the IP phone also sends a TCP "connect" message to a backup CCM so that the IP phone can fail over to the backup CCM. The IP phone also can be configured with a third (that is, tertiary) CCM, to which it can fall back if both the primary and backup CCMs fail.

IP Telephony Design Options

When you create your cluster, you need to identify the roles that the various CCMs play. For example, because the Publisher plays such a vital role, you probably do not want IP phones registering with the Publisher. Also, the IP phones' configuration files are stored on a Trivial File Transfer Protocol (TFTP) server, and you might want to use the Publisher for that function. Finally, you might want to designate which CCMs are going to be backup CCMs for primary CCMs.

Also, you need to select your redundancy design approach, either 1:1 redundancy or 2:1 redundancy. With 1:1 redundancy, each primary CCM has a dedicated backup CCM to take over in the event of a primary CCM failure. To reduce costs, however, you can opt for a 2:1 redundancy model, where you have one backup CCM for every two primary CCMs. Although this provides cost savings, you sacrifice an extra layer of redundancy.

Redundancy Design Options



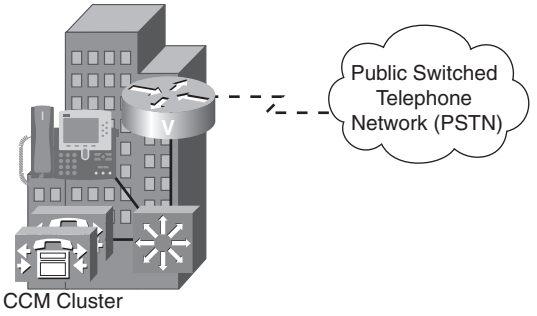
The following table, which is specific to the MCS-7835 server, indicates the number of CCM servers that Cisco recommends to support a specific number of IP phones.

IP Phones	1:1 Redundancy	2:1 Redundancy
Up to 2500 (5000 device units)	3 CCM servers: • 1 Publisher and TFTP server (for 1000 or more IP phones) • 1 primary CCM • 1 backup CCM	3 CCM servers: • 1 Publisher and TFTP server (for 1000 or more IP phones) • 2 primary CCM • 1 primary CCM
Up to 5000 (10,000 device units) units)	5 CCM servers: • 1 Publisher and TFTP server • 2 primary CCMS • 2 backup CCMS	4 CCM servers: • 1 Publisher and TFTP server • 2 primary CCMS • 1 backup CCMS
Up to 10,000 (20,000 device units)	9 CCM servers: • 1 Publisher and TFTP server • 4 primary CCMS • 4 backup CCMS	7 CCM servers: • 1 Publisher and TFTP server • 4 primary CCMS • 2 backup CCMS

When designing an IP telephony network, select from one of the following four CCM design models:

- **Single-Site**—IP phones and CCMs are located at a single site. This model has the following features:
  - — Uses the G.711 codec for all calls.
  - — Leverages high-availability features in the network infrastructure.
  - — Has a simplified dial plan.

Single-Site Model



With Single-Site Model, all CCMs and IP Phones reside at a single location. The PSTN is used for all calls between the campus and the outside world.

- **Centralized Call Processing**—IP phones are at multiple sites, and all CCMs are at a single site. This model has the following features:
  - — CCM cluster is located at a central site.
  - — IP phones are located at multiple sites.
  - — Has lower maintenance costs than the Distributed Call Processing model.
  - — Uses Survivable Remote Site Telephony (SRST) for remote site redundancy.
  - — Uses WAN bandwidth for call setup.