

A DEMONSTRATION OF THE INTEROPERABILITY OF LEGACY AND IP BASED  
TELEPHONY AND VOICE MAIL SYSTEMS

by  
John S. Alling

A Master's paper submitted to the faculty  
of the School of Information and Library Science  
of the University of North Carolina at Chapel Hill  
in partial fulfillment of the requirements  
for the degree of Master of Science in  
Information Science

Chapel Hill, North Carolina

April, 2000

Approved by:

---

Advisor

## Table of Contents

Abstract.....	2
Introduction.....	3
Significance.....	3
Packet Telephony Background.....	5
Method and Results.....	9
CallManager and IP Phone Setup.....	9
uOne Installation.....	13
uOne User Configuration and Message Notification.....	17
Octel Integration.....	19
Call Routing.....	23
Troubleshooting.....	26
Analog Access.....	28
Conclusion.....	32
Bibliography.....	34
Appendix.....	35

John S. Alling. A Demonstration of the Interoperability of Legacy and IP Based Telephony and Voice Mail Systems. A Master's paper for the M.S. in I.S. degree. April, 2000. 36 pages. Advisor: Bert J. Dempsey

This project describes the equipment and processes required to create a small laboratory voice network for the purposes of demonstrating the interoperability of Internet Protocol (IP) based telephones and traditional analog phones with a Windows NT server IP-based voice mail application, as well as with a legacy Octel voice mail system. Background information necessary to understand IP packet telephony is presented along with step-by-step discussion of the processes involved and a number of diagrams to gradually build the concepts necessary to understand the workings of the system.

Headings:

Telecommunications

Telecommunications -- Evaluation

Voice mail systems

Voice mail systems -- Evaluation

Electronic mail systems

Electronic mail systems -- Evaluation

## Introduction

### Significance

Computer networks arose as a way to alleviate and prevent the effects that occurred when information and other resources were maintained on separate, incompatible systems inaccessible to one another. While network connectivity is now nearly ubiquitous in the corporate environment, many such electronically connected users still have a need for older means of communication, specifically telephone-based applications such as voice, voice mail, and fax. Though these older systems complement one of the most popular networked applications, email, it has led to the recreation of so-called islands of information.

For many the situation is beginning to represent the state of computing prior to the evolution of corporate computer networks, as a diversity of computer and communications hardware platforms are being used to store and maintain a variety of message databases, specifically email, fax and voice mail communications. In addition to the end-user hardware necessary to maintain and access these systems as they exist in most environments, separate voice and data network infrastructures are in place to carry these different traffic types to the users. Consequently there is currently a great deal of talk about convergence, the merging of these two discrete types of networks into a single multiservice entity. While convergence promises to save money by merging redundant connections into a single multiservice network, this in itself will not prove directly

beneficial to users. One proposed way that users can benefit from data and voice convergence is through unified messaging. The goal of unified messaging is to bring a suite of message types under a single application which collects, stores and retrieves from a variety of client platforms and applications (Cisco Systems, Inc. [Cisco], 1999)

Cisco Systems Inc. announced their Architecture for Voice, Video and Integrated Data (AVVID) initiative in 1999. Through selective acquisitions and integration of existing Cisco technology the company intends to offer complete end-to-end solutions and migration strategies to converged networks to its customer base.

For new installations and small branch office locations, a solution based upon Cisco AVVID can be implemented immediately . . . . For larger enterprise customers and those with a larger installed base of legacy telephony, the process will require an evolutionary approach where the New World integrates with, expands, and encompasses the old, slowly removing the obsolete equipment . . . . Once the technology has been evaluated and the operation and benefits verified, a parallel deployment will occur where users are migrated slowly to the new system . . . . Once the process of migrating users to the Cisco AVVID architectural model has begun, it will also be necessary to integrate with the existing voice mail system pending a migration to a standards-based unified messaging solution (Cisco, pp. 20-21).

With a strong belief that a key feature of converged networks and unified messaging will be open standards interoperability between Cisco and third-party products an effort has been made to provide demonstrations of these capabilities in the Performance Design and Verification Lab (PDVC) in Research Triangle Park, NC. Currently the PDVC provides customized, full-scale replication testing services for the design of customer networks. However due to the complexity of multiservice networks and limited availability of non-Cisco products to the lab, AVVID functionality is instead being demonstrated to customers through a handful of paradigms. Engineers completed setting up a number of these proof-of-concept network paradigms and began offering

tours and demonstrations in February 2000. To complement the network already in place the PDVC acquired an Octel 250 voice mail system from Lucent. This will be used in conjunction with Cisco's voice mail solution, Unified Open Network Exchange (uOne); a product that is evolving into a multifunctional unified messaging system.

The purpose of this project is to create a small laboratory voice network that demonstrates the interoperability of various components of Cisco's AVVID initiative with the legacy Octel voice mail system. By demonstrating Cisco's compatibility with Octel, visitors to the lab who are attracted to the features and capabilities of uOne will see how the two systems can exist together as part of a migration strategy. The steps to complete the project are presented in a manner that piece-by-piece builds the desired structure as illustrated by accompanying network diagrams that reflect the changing topology. Packet telephony protocol essentials are presented before stepping through the basic phone setup, which is then used to illustrate the basics of call setup. Continuing to the installation and configuration of unified messaging, preparation of the Octel 250, call routing, and the integration of analog phones, a final diagram will be presented to illustrate the protocols being used and their alternatives.

### Packet Telephony Background

Generally speaking a packet telephony network has three layers: a packet infrastructure that carries the voice traffic, call-control for setting up and disconnecting services, and an application interface (API) that defines the available services (Davidson, Peters, & Gracely, 2000). As discussed in the introduction an attractive part of packet telephony is a rich set of features in the API, including the integration of email, fax

delivery and voice mail. Since the implementation of and rationale for these features has been covered, the other two layers will now be considered.

Given its widespread popularity, IP networks are generally assumed to be the packet infrastructure of choice. On these networks for voice traffic, like most real-time applications, it is preferable for the media streams that actually carry the voice traffic on these IP networks to lose packets rather than deliver them out of order in a retransmission. Though it might seem that there would be a wide variety of protocol implementations to accomplish this, according to Davidson et al. (2000) “To date, all VoIP [Voice over IP] signaling protocols utilize RTP/UDP/IP as their transport mechanism for voice traffic” (p. 23). User Datagram Protocol (UDP) is a logical choice for these criteria since it provides connectionless transport services, leaving the implementation of reliability and therefore retransmission to higher-level applications. Given that the UDP header contains no information to enable the receiver to distinguish the order in which the data was sent (Stevens, 1994), and knowing that IP networks can deliver packets out of order, the time stamping provided by Real Time Protocol (RTP) allows the application to determine the proper order of data and ensure its timeliness.

Davidson et al. (2000) go on to say that in addition to terminating call streams, call-control “in a nutshell, is the process of making a routing decision about where a call needs to go and somehow making the call happen” (p. 25). While there is an analogy between the call-control layer routing a stream based on a target phone number and IP routing a packet based on destination address there is an important difference. IP routes every packet individually, making a separate decision for each and every packet. Call-control protocols on the other hand typically make these decisions once, when the call is

initiated, through a series of steps known as the setup process. While packet infrastructure has virtually standardized on RTP/UDP/IP, many choices exist for call-control.

The typical call placed today over the Public Switched Telephone Network (PSTN) uses Signaling System 7 (SS7) as part of its call-control system. The well-established PSTN is generally homogenous and implements a standard set of services, and therefore the needs addressed by a single protocol have wide applicability. The same is not true of IP networks, which have varying capabilities and purposes, therefore requiring the flexibility to be customized and the need for a variety of choices. While there are currently only two call-control standards supported by the infrastructure used in this project, additional options will be mentioned, as the AVVID environment will incorporate them in the future.

Nearly all the protocols discussed in this paper are open, standards-based internet protocols. One exception is the call-control layer implemented on most of the Cisco end-user equipment that communicates via a proprietary “Skinny” station protocol (Lewis, 2000). Given its proprietary status, information about Skinny implementations is not widely available. Later in this paper when the first Cisco phones are described, information will be presented detailing Skinny’s call setup mechanism. While the use of Skinny might seem at odds with stated goals of interoperability, the AVVID CallManager supports Skinny as well as H.323 and as such serves to connect the two protocols. While usually referred to as a single protocol, H.323 is actually a suite of protocols that encompasses many different functions into three different elements, endpoints, gateways and gatekeepers.



Endpoints provide point-to-point and multipoint conferencing for data voice and video. Endpoints can be devices, such as a videoconferencing card for a computer's PCI slot, or a software application like Microsoft's NetMeeting. Gateways are in essence translators, connecting different networks layers that might even use different stream transmission formats. In the AVVID architecture CallManagers can serve as H.323 gateways, which is what provides the interoperability between Cisco's proprietary Skinny protocol devices and other manufacturers' equipment. Gatekeepers provide additional services to endpoints beyond what the gateways offer, including bandwidth control. They are, however, optional and were not implemented in this project since there are few bandwidth limitations in the local area network environment used.

Though H.323 is the most widely implemented VoIP call-control protocol, due to its wide range of features and complexity it is not considered to be scaleable to PSTN networks (Davidson et al., 2000). Alternatives include Session Initiation Protocol, or SIP. While SIP is known for its simplicity, it is still quite flexible; in part due to its basis in Hypertext Transport Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP) that facilitates modularity and code reuse (Lewis, 2000). As such, SIP is a text-based protocol based on client-server architecture that is easy to debug and troubleshoot. SIP has many of the same signaling capabilities as H.323 and other standards, and can even be used in conjunction with them. Davidson et al. (2000) state that though SIP implementations are generally faster at call setup than H.323, its standardization was not complete at the time they went to press.

A fitting complement to SIP is Media Gateway Control Protocol (MGCP), which came from a combination of the Simple Gateway Control Protocol (SGCP) and Internet

Protocol Device Control (IPDC) standards. MGCP focuses on splitting gateway functions between a media gateway and a Media Gateway Controller (MGC). The MGC controls the gateways and provides centralized intelligence. In this way call control is handled by the MGC, while gateways can be designed with hardware that optimizes voice compression and packetization (Lewis, 2000). In the words of Davidson et al. (2000) “SGCP and MGCP are vital components used during the transition from a network whose components are in one monolithic platform to a network whose components are distributed” (p. 283). This description of an environment that requires the characteristics of these protocols seems to mirror the direction of current changes in networks. The authors continue and say that since vendors have been implementing the MGCP standard, “deployment of this technology is imminent. Although this protocol might seem complex, it is actually a simple, yet powerful, protocol that will be the basis for packet-based voice networks for many years to come” (p. 283).

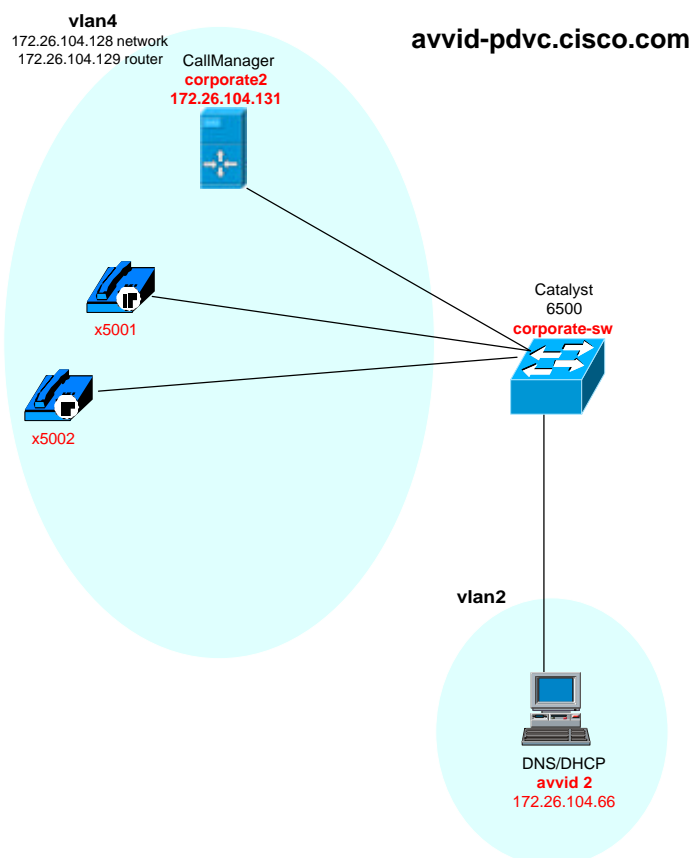
## Method and Results

### CallManager and IP Phone Setup

This first stage involved the configuration of two model 30 VIP Cisco IP Telephones and a machine running Cisco CallManager software, then verification that the two IP phones could communicate with one another. CallManager is a replacement for the traditional Public Branch eXchange (PBX) that controls the telephone system in most enterprise telephony environments. The enterprise version 4.1E of CallManager used for

this project runs on Windows NT Server and communicates over an IP network with other devices.

The first step was to setup and configure a Cisco Media Convergence Server (MCS) 7830 with the prerequisite software needed to run CallManager. Though CallManager could work on any computer running Windows NT 4.0, Cisco only supports the application on the MCS 7830 and MCS 7820. Both machines are configured with Intel Pentium III 500 MHz processors and 256 MB RAM, though the 7830 has better fault tolerance with redundant power supplies and a RAID-1 disk mirroring SCSI hard drive. Given that the uOne application stores a large quantity of user data in the form of audio files in a database of voice mail messages it is only supported on the MCS 7830. The machine I was allocated, corporate2.avvid-pdvc.cisco.com, was already setup with the required system software (NT Service Pack 5, Microsoft IIS Option Pack 4, Internet Explorer 4 and Active Server Pages) along with an older version of CallManager, release 2.3. At that time the latest released version of CallManager was 2.4(2b), which I downloaded from an internal Cisco source and installed over the existing copy. On the Catalyst 6500 switch at the core of the AVVID test network a new virtual LAN (vlan) was created to host the CallManager and two IP phones as shown in Figure 1. Having been allotted the address space 172.26.104.128 – 172.26.104.159, I choose to use the network mask 255.255.255.224 in order to create address ranges for two separate networks. As a result the IP address range 172.26.104.129 through 172.26.104.142 was available for hosts and router interfaces on vlan 4. Since the AVVID DNS and DHCP server avvid2 was located in vlan 2, it proved necessary to route traffic between the vlans. The Catalyst 6500 being used, corporate-sw.avvid-pdvc.cisco.com, has routing



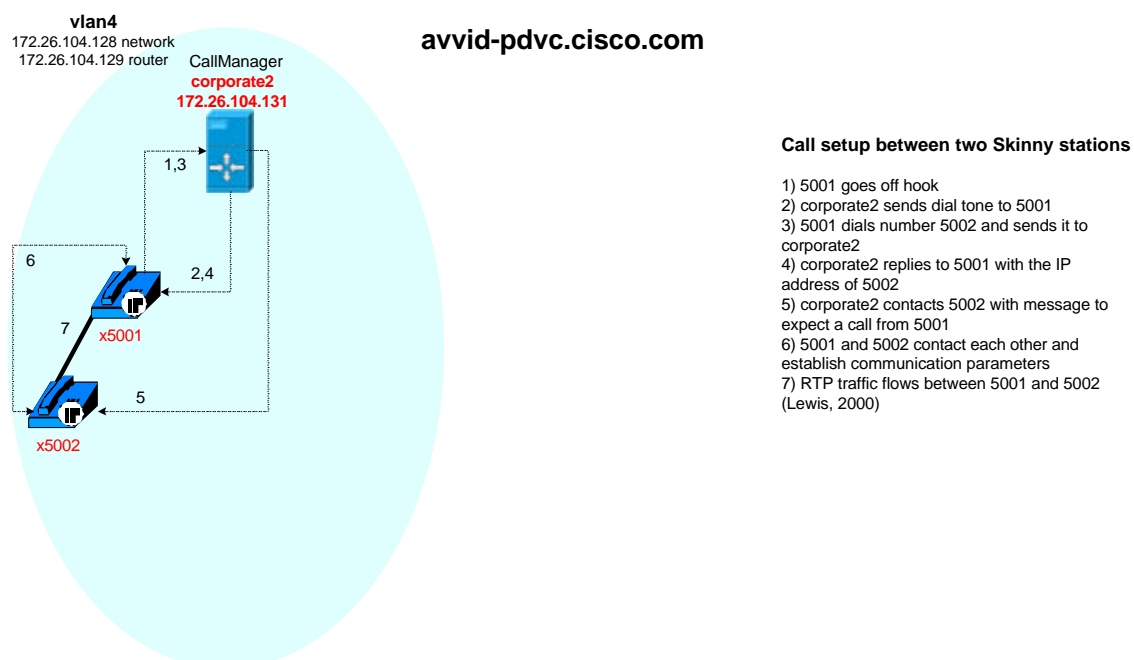
**Figure 1.** The initial network configuration which shows the logical grouping of equipment into virtual LANs (vlans) and physical connectivity to core device corporate-sw. At this stage the two phones shown are operational and can call one another.

functionality built into a module in its supervisor card. That module was already running the EIGRP routing protocol to direct traffic between vlans 1, 2, and 3. Consequently a virtual interface was configured in the routing module on the supervisor card that gave vlan 4 a router interface, 172.26.104.129. Because DHCP requests are typically only broadcast to hosts on the local network, this virtual router interface was configured with the Cisco IOS “ip helper” command to enable DHCP requests issued from this network to be forwarded to the server on vlan 2 and returned to vlan 4 after being fulfilled. This server, a Dell workstation, uses Cisco Network Registrar software to provide DHCP along with DNS service to the entire AVVID PDVC network. Though the primary

function of a DHCP server is to allow the dynamic configuration of hosts with IP addresses and subnet masks, other configurable options are available. The DHCP scope I created for the address range of 172.26.104.137 – 142 was configured to supply hosts with the IP address of their default DNS server, as well as the name of their default TFTP server.

Next the CallManager software on corporate2 needed to be configured with information about each of the IP phones to be used. Virtually all administration for a CallManager is done from a web interface, in this case a web-based configuration wizard. Information the procedure needs includes the hardware MAC address of the phone, device model, a directory number (phone number) that is assigned to the phone, along with an administrator assigned name and description. At the end of the procedure the phone is reset, a process that causes a configuration file for the device to be generated and stored in the CallManager's TFTP directory. Also stored in this directory are operating system binary files, known as loads in AVVID parlance, for the devices controlled by CallManager. A special Ethernet frame is then sent to the MAC address supplied during configuration, which forces the device to reboot. Upon booting the phone issues a DHCP request, and after receiving the reply does a name server lookup for its default TFTP server, which is the CallManager corporate2. The phone then sends a request to the TFTP server for its configuration file. The configuration file includes information such as programmed speed-dial buttons, the phone's directory number, and the name of its load. Although the phone might already have a load image stored locally, if the version specified in the configuration file is different, the device simply requests this particular load from the TFTP server (Lewis, 2000).

After this the phone is fully operational and provided another phone is similarly configured on the same CallManager, calls can be completed between the two. This functionality was verified for the first phones setup, directory numbers 5001 and 5002, which use CallManager corporate2. These phones can only communicate using Skinny and require the help of CallManager for call setup and teardown (see Figure 2 below).



**Figure 2.** A graphical representation of call setup process using the Skinny protocol. Overall network configuration is the same as is represented in Figure 1, though physical connections and equipment have been removed to emphasize the logical call flow.

### uOne Installation

With this basic call functionality in place the next goal was to install and configure the components of uOne on corporate2 as summarized in Table 1. Though its accounts are administered through a separated interface provided by Netscape Directory Server, uOne is an add-on to the CallManager software. As such some modifications to the configuration of CallManager, including additional components, are required. Cisco

Table 1.

Summary of uOne components and their functions

uOne Component	Function
Cisco Directory Services	A prerequisite for installing uOne. Web based phone book which allows users to access information entered into CallManager by administrator.
Conference Bridge	A prerequisite for installing uOne. Allows for call conferencing.
Media Termination Point	Another prerequisite for installing uOne. Enables supplementary services, such as putting a call on hold.
Apache Web Server	Provides a web interface to administer Netscape Directory Server and Netscape Messaging Server. Runs on port 280.
Netscape Directory Server and Messaging Server	Stores, maintains and retrieves user information, profiles, and voice mail messages. Later implementations will also store faxes and email with voice messages and allow them to be retrieved through multiple applications.
Netscape Communicator	Necessary to ensure compatibility when accessing and maintaining Netscape Servers through uPower.
uPower	Web based application with interface into Netscape servers. Used to administer users, configure options.
uOne pilot directory number	Phone number users dial (x5500) to access uOne voice mail system. If line associated with this number is in use, configuration allows for automatic rollover to directory numbers for three remaining lines.
MWI	Message Waiting Indicator.
MWI On MWI Off	Numbers used to (de)activate MWI lamp on users phone. x5901 and x5902 respectively, but users need not know these as this information is used transparently by CallManager.
Cisco Messaging Interface	Allows user to press button next to MWI to automatically connect with uOne or Octel.

Directory Services, Conference Bridge, and Media Termination Point all must be installed on the CallManager before uOne installation can begin. Binaries to install these programs came with the CallManager distribution and installation was achieved through CallManager's web interface with no special intervention or configuration.

Until recently the installation of uOne was known to be an involved process that could take as much as a half a day, as it involved installation of numerous third-party components such as Apache Server and three Netscape products, Communicator, Directory Server and Messaging Server. Only after installing and completing configuration of these programs could uOne be properly installed. Since Cisco had moved to supporting only the standardized MCS 7830 hardware platform for uOne, it became feasible to roll these third party components, their configuration parameters and uOne into a single unattended installation program. I attempted to use this greatly simplified unattended distribution hoping it would complete in the estimated 7 –10 minutes. Instead the machine remained non-responsive for approximately an hour and a half, prompting me to abort the process. As recommended by the support literature the numerous components of the installation routine were uninstalled, and the procedure attempted again. These and other ensuing attempts at unattended installation were aborted after running for one hour or more. After formatting the disk and installing all system and Cisco software from scratch to no avail I tried manual installation of the individual elements and isolated the problem to an error code during the installation of Netscape Directory Server.

As I searched internal Cisco newsgroups for a solution I learned that other users had also encountered similar problems when the fully qualified domain name (FQDN) of



the CallManager exceeded 32 characters. The FQDN of the server was intended to be corporate2.avvid-pdvc.cisco.com, similar to the names being used for the rest of the devices in the AVVID demonstration network being built and less than 32 characters. Apparently this error code had been appearing during the unattended installation routine, but was obscured from view by other windows on the screen. Netscape had no documentation about this error available, though additional newsgroup postings indicated that the error occurred not only for FQDNs with lengths greater than 32, but also for those with more than two periods. Though the name we choose was a valid FQDN according to the specifications presented in Stevens (1994) the Netscape Directory software would not support it.

Accordingly the machine was renamed corporate2.cisco.com. Though outside the name space controlled by our lab, this name was not in use elsewhere at Cisco. Rather than renaming all other machines in the test network, the domain suffix search order for corporate2 was modified to first append the domain name avvid-pdvc.cisco.com to host names. As a result, though the machine has a FQDN of corporate2.cisco.com, if a name server lookup is done from this machine for the host corporate2 the first name the machine will try to get resolved will be corporate2.avvid-pdvc.cisco.com. This request returns the desired IP address 172.26.104.131. So though this machine is configured to be corporate2.cisco.com, to all other hosts it appears the same when referenced as corporate2.avvid-pdvc.cisco.com. After devising this workaround the uOne installation proceeded successfully in the advertised 7 – 10 minute time frame. Subsequent to these events I learned that release of uOne was halted to fix this and other bugs.

### uOne User Configuration and Message Notification

Although uOne currently only supports voice mail it is evolving into a true unified messaging system that also supports email and fax communication. For this reason uOne user accounts actually exist in Netscape's Messaging Server, a traditional email server. The installation routine installed an Apache server accessible at port 280 rather than the standard web server port of 80, which is bound to Microsoft's web server implementation to support the CallManager web interface. Through this Apache interface into the Netscape servers, known as uPower, I created a profile in the directory server for two dummy users at phone numbers 5001 and 5002. In addition to creating entries in the directory server this interface automatically created email boxes for the users in the messaging server. These directory server entries were created based on a class of service (COS) template pre-programmed into uPower, which simplifies the process of creating users with a common set of default parameters. Apart from parameters already defined by the COS, the information that was explicitly required to add users into uPower consisted only of name, telephone number padded with leading zeros to ten digit length, email address (user@corporate2.cisco.com), email and telephone access passwords, type of user (experienced or first-time), gender, and time-of-day and day-of-the-week profile information.

In order for a voice message to be left or retrieved, CallManager employs directory numbers that it will use to transfer callers into the uOne application. Normal licensing for uOne provides up to four simultaneous connections to the uOne interface, though an option exists to purchase licenses for additional connections. Each connection must be configured with its own directory number as well as an alternate to try should

that connection be busy. A pilot directory number, the only number users need remember, 5500 was selected and assigned to the first uOne port. If this directory number were dialed with a call in already in progress on it, an attempt would be made for the next entry, 5501. If that was also busy 5502 would be tried, then finally the fourth uOne entry 5503. In the event all four are simultaneously in use after the fourth directory number failed, the system could be set to transfer the caller to an operator. Since the lab has no such permanent attendant our system is configured to continuously cycle attempts back through the set of four numbers, starting over again at 5500.

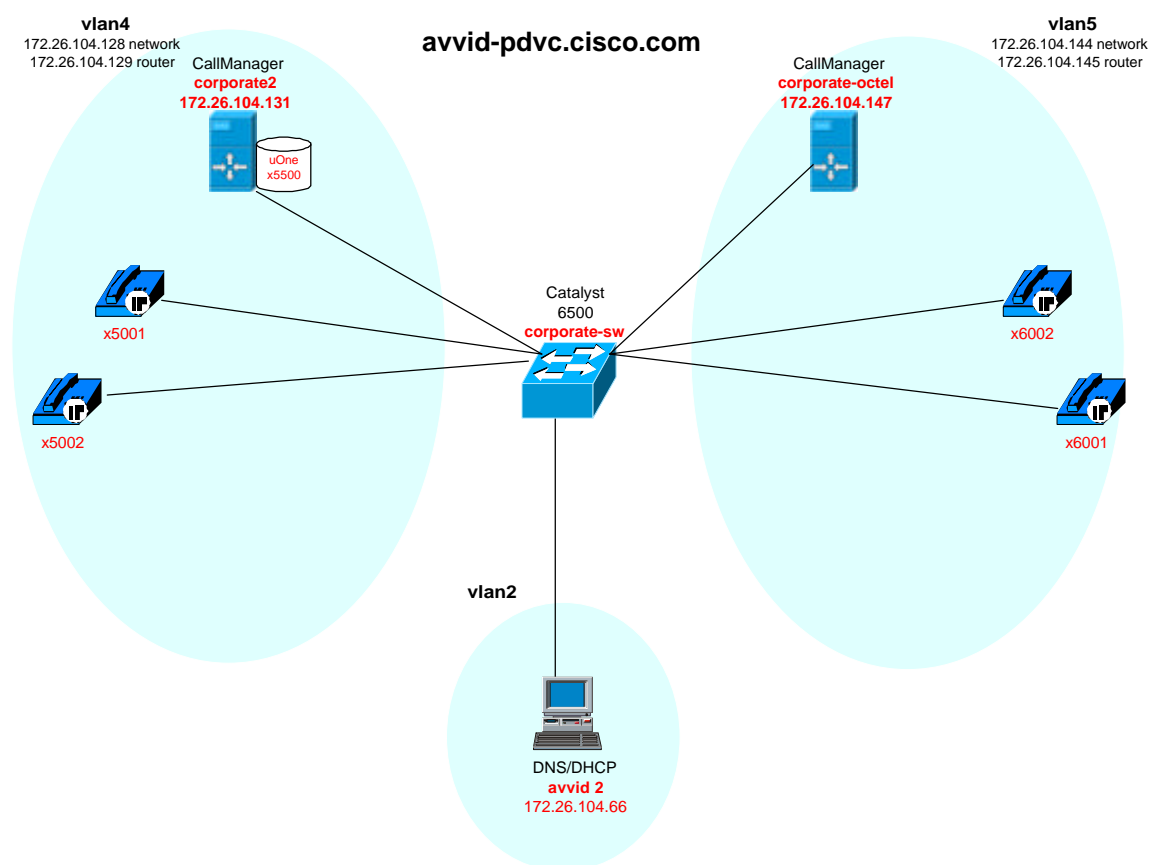
Whereas most users leave their email client running continuously on their computers, allowing for prompts to appear and sounds to be played when a new message arrives, a different notification scheme is typically used to indicate waiting voice mail messages. One way this is achieved is by having a Message Waiting Indicator (MWI) lamp on the phone light when new messages have arrived. The CallManager interface was used to specify some additional uOne parameters, the MWI On and MWI Off numbers, 5901 and 5902 respectively. These numbers are used by the uLite service, a component of uOne, to have a special call placed to a user's phone to have the MWI light turned on or off based on the presence or absence of waiting, unheard messages. Though no voice traffic is sent, CallManager tracks the progress and results of the call, enabling uLite to be guaranteed to know the state of each phone's MWI light. Finally the Cisco Messaging Interface within CallManager was configured. This feature maps the button next to the MWI light on a Cisco IP phone as a speed-dial button for the voice mail pilot directory number.

With these steps complete I dialed the uOne pilot directory number, 5500, from each phone and by following prompts given by uOne changed user passwords, recorded user names and default greetings. To test the system I phoned from 5001 to 5002 and left a sample message, then went to 5502 to verify the MWI light was on. Pressing the MWI button I was connected to the uOne system from which I retrieved and deleted the message, resulting in the MWI light for 5002 turning off. I then used uOne prompts to leave a message for 5001's mailbox, which was achieved without ever ringing 5001 directly. Returning to 5001 to verify MWI operation I retrieved and deleted the second sample message having demonstrated basic uOne operation. The phones configurations on CallManager were changed to automatically forward all calls to 5500 for the purposes of speeding up future testing by eliminating the 6 rings that had been occurring before the call attempt was rolled over to uOne.

### Octel Integration

Each CallManager can support a single voice mail system, so in order to use the Octel two additional phones and another MCS 7830 were obtained and configured similarly to the first CallManager and phones. Rather than administering directory numbers beginning with 5, this CallManager, corporate-octel.avvid-pdvc.cisco.com, would be responsible for numbers starting with 6. Given that IP phones in this portion of the network need to be configured with the CallManager corporate-octel as their default TFTP server I added another scope of IP addresses on the DHCP server. To accommodate this new network vlan 5 was created on corporate-sw for corporate-octel and its associated devices. As with the previous setup a virtual router interface,

172.26.104.145 in this case, was created in corporate-sw's routing module for vlan 5 and configured to pass DHCP requests. The address range 172.26.104.153 – 158 was added to the DHCP server for use on vlan 5, with corporate-octel being assigned as TFTP server to clients receiving these addresses. After these changes the diagram found in Figure 3 represents the equipment configuration.



**Figure 3.** Updated network diagram showing uOne residing on corporate2. Voice mail accounts exist for users of phones x5001 and x5002, which they access by dialing uOne's main access number x5500. A second server, corporate-octel, has been setup in a separate vlan to support Octel voice mail and to provide call setup services for x6001 and x6002. Despite network connectivity between vlans 4 and 5, calls cannot be completed between them since the CallManagers are unaware of one another and therefore cannot cooperate in brokering call setup.

During an earlier portion of the project a technician from Lucent, the parent company of Octel, had come and assisted with the physical setup and basic out-of-box assembly of the Octel 250. Lucent has field configuration notes on setting up Octel systems with a variety of PBX systems, including one pertaining to Cisco CallManager. This document was also available internally at Cisco and was reviewed prior to the technician's arrival to ensure we had all necessary equipment.

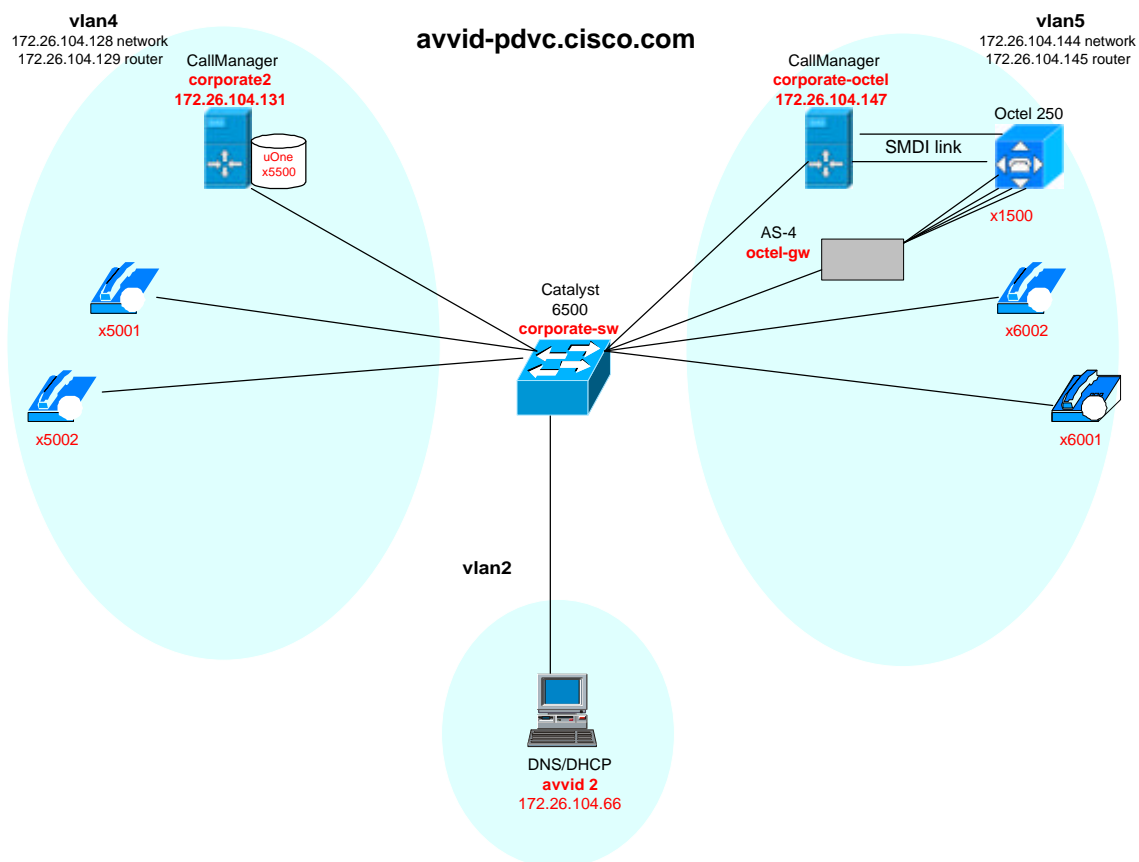
As with uOne, by default four phone lines can simultaneously access the Octel. To allow voice communication between the CallManager and Octel a four-port Cisco Analog Station gateway (AS-4) was used. The device has 4 analog voice ports and a 10BaseT Ethernet port, which was patched into vlan 5. Once patched in, the gateway was added into the CallManager's database through the same procedure used for the IP phones, and then forced to reboot in order to obtain a DHCP lease and TFTP its load and configuration file. In most situations this configuration file contains four directory numbers, which are used by four analog phones to communicate with the CallManager, essentially making the analog phones into IP phones with the help of the AS-4. However in this situation the AS-4 is used to provide up to four simultaneous analog voice connections into the Octel for the purpose of leaving and retrieving messages.

To interface the analog lines from the AS-4 with the Octel a 25-pair amphenol cable was connected to a modular interface on the FLT-8 (Fax Line TIC) board in the Octel. The other end of the amphenol cable was distributed into a breakout of unnumbered RJ-11 interfaces. The configuration note stated that the AS-4 had to connect with pairs 1-26, 3-28, 5-30 and 7-32, which were traced within the breakout box to the appropriate RJ-11 connectors. After the Lucent technician performed the typical initial

out-of-box system configuration and created mailboxes for extensions 6001 and 6002, telephone connectivity to the Octel from 6001 was verified by dialing its voice mail directory number, 1500 and obtaining the Octel greeting.

Since uOne services run on the same NT Server as CallManager, communication between the two applications is simple compared to the Octel. Another link was necessary to pass call information back and forth between the Octel and the CallManager. Simplified Message Desk Interface (SMDI) is an industry standard owned by Telcordia, previously Bellcore, and supported by a variety of phones, voice mail systems and PBXs to allow an assortment of features to interoperate between these different types of equipment. In this environment the SMDI information is passed using a serial cable to connect an integration link DB-25 port on the Octel's CF card to the DB-9 serial port of the CallManager. The Cisco Messaging Interface software, which had earlier been added to corporate-octel, was configured with the directory number of the Octel, 1500, and the port number, Com1, to which the SMDI serial cable was connected.

With the SMDI connection in place as shown in Figure 4, call information should have been passing over the serial link to accompany the voice communication over the analog line. To test this, another call from 6001 to the Octel was placed. Observing that the call was not automatically routed into the mailbox for 6001 we determined that the SMDI information was not being passed properly. Since SMDI is used to pass call information to the voice mail system as well as to signal the MWI to light, the message waiting functionality was not working either. At this point the Lucent technician left, requesting that we research the Cisco solutions to this problem and then contact him to attempt resolution.



**Figure 4.** Voice mail services are now available to x6001 and x6002 from the Octel 250. Voice traffic is carried to the Octel by four analog phone lines, each of which is reached by dialing the directory number x1500. The AS-4 provides these lines with the help of corporate-octel by sending and receiving voice traffic over IP on vlan 5. Additional call information should pass between CallManager and Octel over the serial SMDI link.

### Call Routing

In order to demonstrate a migration path from an Octel system to uOne users would need to be able to access both voice mail systems from phones connected to either CallManager. While trying to find additional information about the SMDI problem we had encountered I turned my focus to being able to call between phones 5001 and 6001. To accomplish this calls would need to be routed based on the destination number according to a dial plan.



As stated earlier we decided that devices with corporate2 in vlan 4 would have directory numbers beginning with 5, and those with corporate-octel in vlan 5 would begin with 6. This was in essence our dial plan, a structure we developed to simplify the process of call routing. Routing calls is not inherently hierarchical, though it can be in a well-designed dial plan. While it's feasible to assign phone numbers to devices seemingly at random, doing so can lead to inefficient use of processing capabilities when call routing decisions need to be made. Additionally for consistency voice mail directory numbers were assigned to end with 500, though this does not end up factoring in to call routing decisions. In the case of the Octel 250 we decided to use a directory number of 1500 for two reasons. First, it would show the flexibility of call routing by demonstrating the use of two route patterns for devices connected to corporate-octel. Second, the portion of the AVVID network setup for this project would eventually be torn down and reconfigured into the rest of the test network. The dial plan being developed there called for the Octel to be accessible at 1500, and configuring it as such now would minimize later reconfiguration of that device at the conclusion of the project.

Cisco CallManager accomplishes call routing through a three-tiered structure, route patterns, route groups and route points. Route patterns are created to match a range of destination numbers through the use of number, wildcards and access codes. Patterns that will be routed to the same final destination are aggregated together into route groups, sets of patterns that will be forwarded in a similar fashion. These route groups are then associated with one or more route points, or paths, to their final destination. As with IP routing there is frequently more than one pathway between networks. A simple ordinal ranking of preferred route points is made when a route group is associated with multiple

route points so that if the first choice is unavailable the second, third, and so on will be tried instead. In this project multiple paths to reach a given number pattern do not exist, and as such there is a one-to-one relationship between route points and route groups in this implementation.

For the devices already in place call routing was configured, first for the calls originating from devices in vlan 5. From corporate-octel's web interface a route point was created to correspond to the DNS-resolvable name corporate2. Though there are multiple types of route points, the only type used in this project were H.323 gateways, a function the CallManager can serve. After route point creation, a route group was formed and then populated with the route pattern 5XXX, where each X represents a single wildcard digit. Though this single pattern served to match all numbers (5001, 5002 and 5500) assigned to devices by corporate2, two patterns were needed when generating a routing structure to enable calling from vlan 4 to vlan 5. On corporate2 an H.323 gateway route point was configured pointing to corporate-octel. Next was the creation of a route group to use this route point. Finally the route group was populated with the patterns 6XXX and 1500. After completing the route configuration calls were successfully completed between all devices. It was even possible to check and leave messages on both voice mail systems from any of the IP phones.

### Troubleshooting

While researching the SMDI link problem I learned that the serial port on the Octel was configured differently than is typical for a serial port on a machine running Windows. Com1 on the Windows NT machine corporate-octel was set to the typical

default values of 8 data bits, no parity and 1 stop bit and the port was verified to be working properly by testing with another application. The Octel interface had a different setting for data bits, 7, as well as being set to even parity. After updating the configuration for Com1 through the Windows Control Panel no difference was seen in the behavior of the MWI lights for 6001 and 6002. However I did observe that when calling from either of those phones to the Octel system I was directed to login to the appropriate corresponding mailbox, so call information was apparently passing from the CallManager to the voice mail system over the SMDI serial link.

I contacted the Lucent technician by telephone after connecting a modem and phone line to the console of the Octel, enabling him to log in remotely. While I placed calls and left messages on the Octel the technician was logged into the console, redirecting the SMDI information to his console and logging it rather than having it sent out the serial interface. After reviewing this output we determined that there was an error in the original configuration note supplied by Lucent about some of the Octel settings. SMDI traces of calls going into the Octel showed a field named "Message Desk" with values of 001. However the configuration note had stated that SMDI should be configured to use a value of 000 for this field. The technician updated the Octel's configuration and MWI functionality from the Octel 250 to Cisco IP phones was confirmed.

However this MWI feature had been tested by calling from the phone 5001 into the Octel and using the system prompts to leave a message for mailbox 6001, the phone 6001 was never actually involved. Subsequent attempts calling from 6001 to 6002 and vice-versa connected the caller to the appropriate mailbox, but resulted in 10-minute

delays for the MWI to light. After checking and deleting the messages that were left, the reason became clear – the calls were not being properly disconnected. At the end of the brief sample message that I had recorded was the sound of the phone being placed back in the cradle and then a long interval of the “fast busy” or reorder tone, which continued for approximately 10 minutes.

Upon determining the problem with the error in the Message Desk variable in the Lucent configuration note, I tried to locate a revised copy of the document. I found one posted within Cisco that had corrected the Message Desk problem and also provided information I hoped would help with the disconnection problem. According to this revised document the Octel required hearing a dial tone from the AS-4 analog gateway to release the line. Following the updated notes I set the AS-4 to a call-restart value of 1234 ms, rather than its default of 5000 ms. If the default restart value had been closer to 10 minutes this might have been a plausible explanation. This had no effect, as the line was being restarted within just over a second instead of the previous value of 5 seconds.

My next step was to remove one of the analog line connections from the AS-4 to the Octel and replace it with an analog phone. I placed a call from an IP phone to the analog gateway and after answering the analog phone hung up the IP phone. Upon returning to the location of the AS-4 I noticed the line was still in use and heard the reorder tone when I listened to the analog phone. Doing this had enabled me to determine that the problem was not with the Octel, but with Cisco equipment. I later learned that the operating system load for the AS-4 that came with the CallManager version I was using, 2.4(2b), was configured to have the analog gateway play the reorder tone upon disconnect rather than a dial tone. A minor maintenance release had occurred

for CallManager since I'd installed it and both the newest 2.4(3) and older 2.3 versions contained loads for the AS-4 that have it play a dial-tone upon disconnecting. I obtained a copy of this load binary and put it in the TFTP directory of corporate-octel and then updated the AS-4's configuration in the CallManager to specify this new load. Upon reset the gateway rebooted and read its updated configuration file and the load change worked. The lines no longer stayed open after hanging up, the 10 minutes worth of reorder tone was eliminated from the end of messages, and the MWI light came on promptly after disconnecting.

### Analog Access

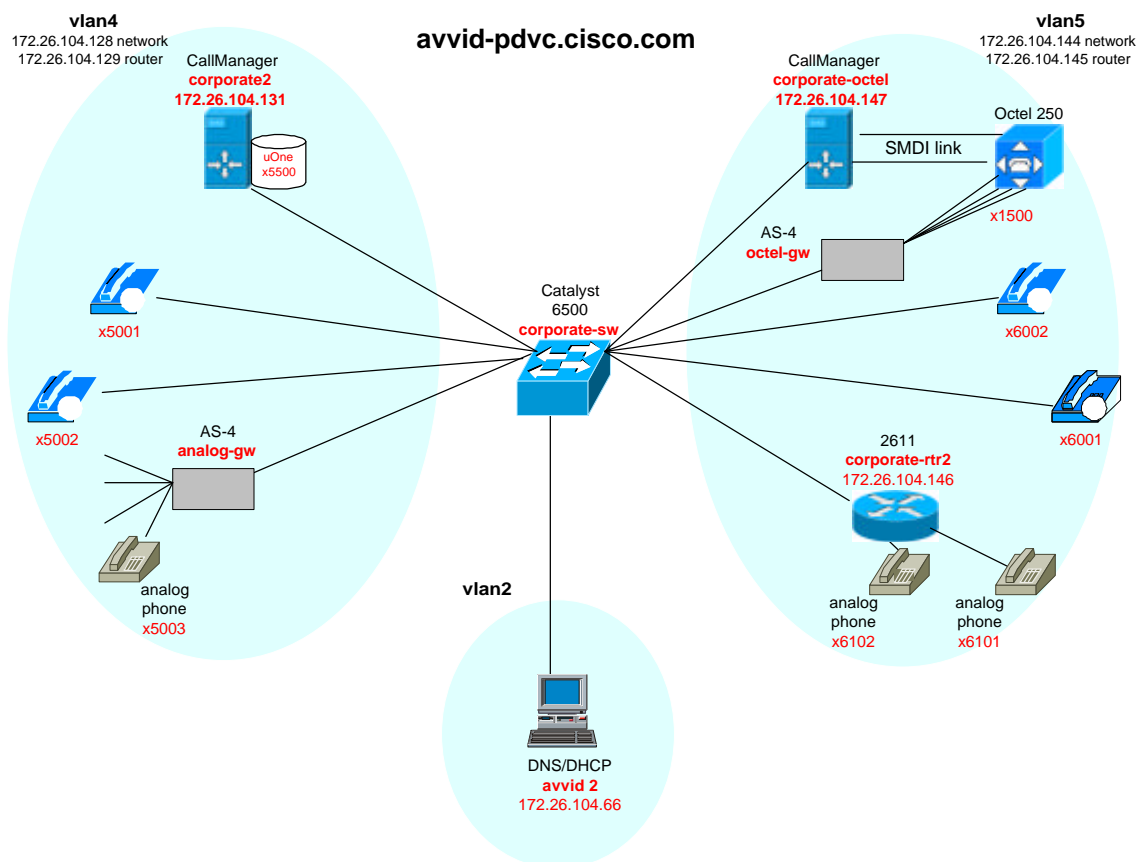
To this point I had shown the operation of Cisco IP-based phones with both a legacy and an IP based voice mail system. Another goal of demonstrating interoperability was to establish the ability of analog phones to use Cisco's uOne voice mail, as well as to show that analog phones could use the Octel over an IP network. There was an additional goal of showing these features for digital phones connected to a traditional PBX, however the PBX for our lab has not yet arrived.

One potential use of an AS-4 analog gateway had already been demonstrated through the Octel's analog connections to the CallManager. To show that the capability of the AS-4 to provide a connection for a typical user's analog phone to the CallManager I obtained another AS-4 and placed it in vlan 4 with corporate2. The load file I had obtained to fix the problems with the AS-4 to Octel disconnection problem was copied over to corporate2's TFTP directory to avoid the problems I'd encountered earlier. Directory number 5003 was assigned to the first port on the gateway and the device was

rebooted. Calls were made from this analog phone to all other numbers attempted, and incoming functionality verified from devices in both vlans.

A very different technique was used to put analog phones into vlan 5. A voice-capable Cisco 2611 router, corporate-rtr2, was obtained to demonstrate the use of analog phones without a CallManager. To accomplish this a special version of Cisco's router operating system software, IOS, was needed along with cards to plug into the router's modular slots. IOS version c2600-js-mz.120—7.T was selected as it supports voice features and was already available on the lab's TFTP server. A voice module was put in the slot, which was then populated with 2 FXS interface cards, each with two RJ-11 analog ports. Two phones were connected to analog ports and router software configuration started through the router's console port using a command-line interface. The router configuration that resulted can be found in the Appendix, and an updated network diagram in Figure 5.

First the configuration was done for the phone connected to the FXS voice port 1/0/0. Actually no configuration is actually applied to the port itself. Instead a voice dial-peer is created, in this case with the type POTS (Plain Old Telephone System). The dial-peer serves a similar function as the route point, route group and route pattern did in the CallManager configuration, routing calls to their destination. Dial-peer 6101 tells the router to send all calls with a destination of 6101 to port 1/0/0, and that the connection type that will be created for such a call is POTS. A similar dial-peer 6102 was created for a phone numbered 6102 on port 1/0/1. While in these cases the dial-peer identification numbers are the same as the number assigned to the telephone on that port, this is not a requirement and is done only for simplicity's sake.

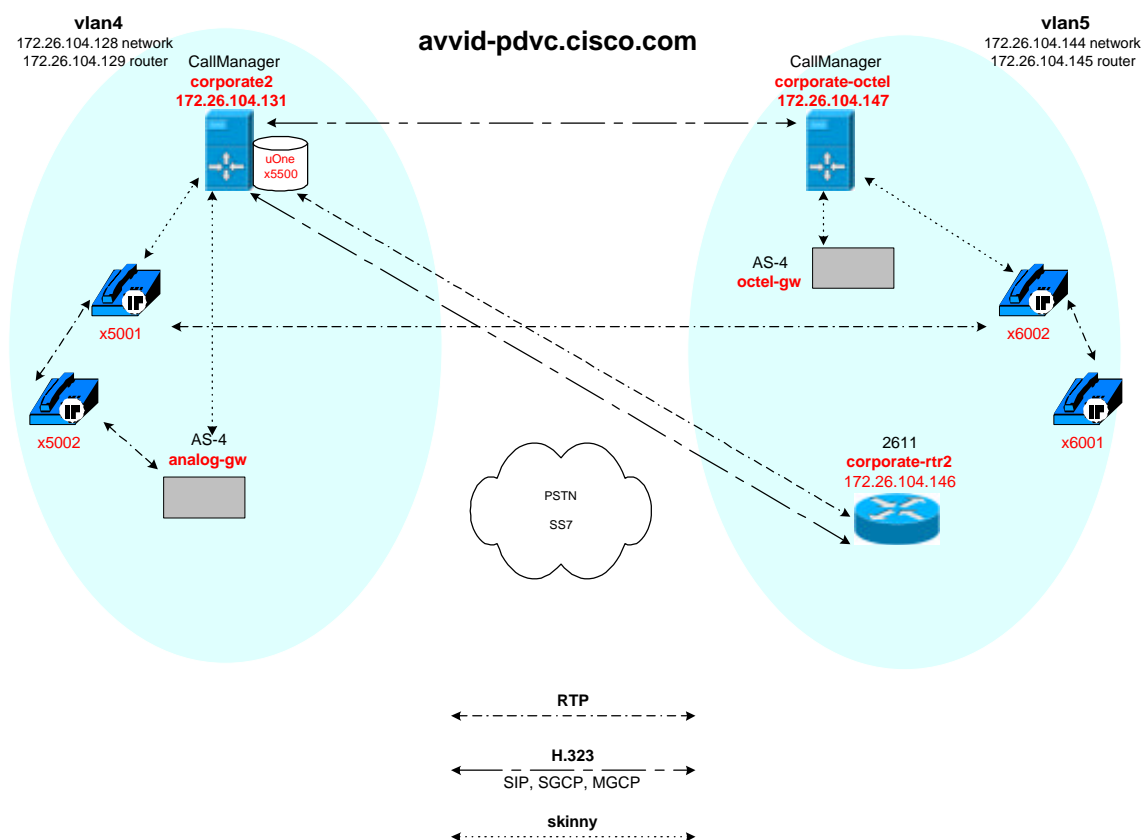


**Figure 5.** The completed network includes a voice mail system, IP phones and analog phones in each vlan. With call routing now in place on the CallManagers, once dial-peers are configured on corporate-rtr2 any phone will be able to call any other phone (analog or IP) as well as either voice mail system.

Without any additional configuration these two phones get a dial tone from their connection to the router and can complete a call between them without the use of IP.

However since we ultimately want to connect to voice mail systems and other IP devices in vlans 4 and 5 voice over IP (VOIP) configuration is required. To accomplish this additional dial-peers are created to match patterns of the destinations, though these are VOIP type dial-peers. Unlike the CallManager interface this information is done with IP addresses, not names, and with periods as wild card characters. A codec (code/decode) specification is required to be selected; here the type g711ulaw is used, as it is the only

codec supported by uOne. With the addition of three VOIP dial-peers the router now knows how to route calls to the two directly connected phones and all of the other devices we've setup for the project.



**Figure 6.** Call setup and voice transmission protocols (RTP, H.323, Skinny) used in the completed network for a number of sample calls. Future releases of AVVID software will allow SIP, SGCP, MGCP and others to be substituted for or used with H.323.

However, a key piece was missing – the ability for calls to get from the CallManagers to the router. For this the router must be enabled to act as a H.323 gateway device as reflected in Figure 6 above, which shows how the various protocols fit in to the completed network. This functionality is turned on with the gateway command (see Appendix), and the interface Ethernet 0/1 is configured to be an H.323 gateway. Then in



a fashion similar to how call routing was setup earlier between the two CallManagers, corporate-rtr2 is entered into both CallManagers as a route point, and a route group created on both for patterns matching 61XX. Two-way connectivity was then verified between these analog phones and the IP telephony devices, including voice mail systems.

### Conclusion

As hoped, this project demonstrated that a legacy voice mail system, an Octel 250, could be used in an enterprise environment without a traditional PBX, substituting a Cisco CallManager instead. Rather than just replacing like with like, this scenario can be implemented for the purpose of transitioning from separate telephone and voice networks to a single converged environment. Users with different types of phones, both IP based and analog can access either voice mail system, affording administrators time to migrate the company to a single solution. Though the full unified messaging capability of Cisco's uOne product is not yet available, the functionality of the network needed to support was demonstrated. The solution built in this project is currently being demonstrated to customers and receiving encouraging feedback.

In the coming months I anticipate receiving updated uOne software that will store email and fax messages along with voice mail, allowing users to check their voice mail using a email client or hear their email rendered by speech recognition software. Additionally, to meet scalability and compatibility demands, support for more call-control protocols will be incorporated. These developments will make converged networks built on Cisco's AVVID technology more attractive by adding functions not

currently support by more traditional telephony networks and facilitating open standards internetworking.

### Bibliography

Cisco Systems, Inc. (1999). Architecture for Voice, Video and Integrated Data (White Paper 9908R). San Jose, CA: Author.

Davidson, J., Peters, J., & Gracely, B. (2000). Voice Over IP Fundamentals. Indianapolis, IN: Cisco Press.

Lewis, E. (2000). Configuring Cisco Voice Over IP. Rockland, MA: Syngress Media.

Stevens, W.R. (1994). TCP/IP Illustrated, Volume 1: The Protocols. Reading, MA: Addison-Wesley.

## Appendix

Configuration file for router corporate-rtr2

```
!  
version 12.0  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
!  
hostname corporate-rtr2  
!  
!  
!  
!  
!  
!  
ip subnet-zero  
no ip domain-lookup  
!  
cns event-service server  
!  
!  
!  
voice-port 1/0/0  
!  
voice-port 1/0/1  
!  
voice-port 1/1/0  
!  
voice-port 1/1/1  
!  
!  
dial-peer voice 999 voip  
  destination-pattern 5...  
  codec g711ulaw  
  session target ipv4:172.26.104.131  
!  
dial-peer voice 6101 pots  
  destination-pattern 6101  
  port 1/0/0  
!  
dial-peer voice 6102 pots  
  destination-pattern 6102  
  port 1/0/1  
!  
dial-peer voice 997 voip
```

```
destination-pattern 60..
codec g711ulaw
session target ipv4:172.26.104.147
!
dial-peer voice 1500 voip
destination-pattern 1500
codec g711ulaw
session target ipv4:172.26.104.147
!
gateway

!
interface Ethernet0/0
no ip address
no ip directed-broadcast
no ip mroute-cache
!
interface Ethernet0/1
ip address 172.26.104.146 255.255.255.240
no ip directed-broadcast
no ip mroute-cache
h323-gateway voip interface
h323-gateway voip h323-id corporate-rtr2
!
ip default-gateway 172.26.104.145
ip classless
ip route 0.0.0.0 0.0.0.0 172.26.104.145
no ip http server
!
!
!
line con 0
transport input none
line aux 0
line vty 0 4
login
!
end
```