CHAPTER 2

Voice over Data: Many Conversations, One Network

Conversations are the basis of human communication. Conversations can be spoken, written, or gestured. Conversations can even be one directional, such as a coach bawling out his star quarterback after an uncharacteristic interception. Conversations may be "one-to-many" (such as a political candidate giving a stump speech) or "many-to-one" (such as a constituency lobbying that candidate after she's in office). Conversations are more than just an analogy for networks—they literally *are* modern networking.

The underpinnings of enterprise networks are also conversations. IP data networks run on protocols that use a conversational approach to data exchange. The most common protocols for web browsing (HTTP) and email (SMTP) use a two-way "data conversation" in order to communicate. The process is simple: a client host sends an inquiry to a server host or a peer host, and then the server or peer sends a response back to the client.

Conversations between hosts on an Internet Protocol (IP) network are similar to those between people, except that instead of using words, the messages are communicated across the networks using units called datagrams. A *datagram* is like a letter in an envelope. Once it has the proper markings, namely the recipient's address and return address, and a stamp, the entire letter can be delivered by the postal service. A datagram's markings are called *headers*, and they contain delivery information, like postal letters: instead of postal addresses, datagrams use something called *host addresses*. Different networking technologies have different names for datagrams, including *cells*, *frames*, and *packets*. Having a good understanding of IP networks is crucial to your success with Voice over IP. An excellent reference on the subject is *TCP/IP Network Administration* (O'Reilly).

When voice sounds are transmitted using datagrams on the IP network, telephony gains all the same characteristics as the data network itself. Just like applications for file sharing and printing via the network, software can be made to perform useful tasks using the datagrams of voice streams and signals—tasks like conference calling

and voice mail. These tasks are the *applications* of Voice over IP. Voice applications delivered using IP datagrams is the essence of VoIP.

VoIP, like the network that carries it, is therefore not an application by itself, but a way to build applications using myriad software tools and devices. These building blocks can be specialized VoIP server hardware like an analog telephone adapter (ATA), or they can be highly programmable servers that do the job of a PBX. Regardless, all VoIP components must participate in the protocol conversations that make the audible, human phone conversations possible. That means that all VoIP components must be speaking the same language.

In human conversation, people can speak many different languages. Even among different dialects of the same language, people can have a hard time understanding each other—a Bostonian and a Texan sound about as different as a Canadian and an Australian, even though they all speak English. Unfortunately, telephony standards have had similar challenges.

Many standards govern the world of Voice over IP, and some have interoperability problems, just as people with local accents sometimes confuse each other. One such annoyance lies in the definition of the word *VoIP* itself.

VoIP or IP Telephony

Are "VoIP" and "IP telephony" two different technologies, or do they both describe the same thing? Well, it really depends on whom you ask. Some vendors prefer IP telephony when referring to their IP-based voice offerings, arguing that VoIP refers to the specific act of transmitting digitized sound data on an IP network and IP telephony refers to the overall technology family. Others give VoIP the broader definition, identifying it as inclusive of IP telephony, and referring to IP telephony only as the act of mimicking traditional telephony applications.

For the purposes of this book, we'll take the latter tack: VoIP refers to the overall technology family, while IP telephony means specific application functions such as call signaling and voice mail. So when we talk about conference calling, we might call it telephony, but when we talk about conference calling, call-waiting, and voice encoding, we will refer to them collectively as VoIP. In general conversation, though, VoIP and IP telephony can be used interchangeably.

VoIP's Pros and Cons

VoIP certainly has a few disadvantages when compared to old-school phone hardware. High-utilization service guarantees are harder to deliver with VoIP than with an old-fashioned PBX. The same scalability characteristics that attract people to VoIP can ultimately be the reasons their implementations fail: a VoIP network can be so extensible that service-level guarantees are hard to make, whereas a traditional

circuit-switched voice network has hard capacity limits, around which levels of service tolerance can be guaranteed easily. Certain broadcast audio applications, like overhead paging, can be difficult with VoIP, too.

The gains VoIP brings to the table far exceed the few difficulties it imposes, though. There's nothing that old PBX can do that a VoIP telephony system can't, even if VoIP makes a few things tougher.

One thing VoIP makes *easier* is physical provisioning. While a PBX requires a network of electrical, usually copper wire, loops, VoIP requires an IP network. Since IP networks are a staple of every modern business, the logistics of building a network for voice is largely simplified because the required physical elements are already in place for other common business applications: databases, messaging, Internet access, and so on. VoIP is carried on the network the same way those are.

If you're an Internet user (and who isn't these days), then you know TCP/IP is the core protocol that defines the architecture of the Internet. In most organizations, and even in many homes, a TCP/IP local area network is an important interpersonal communications tool, used for email, web surfing, and instant messaging. When VoIP replaces the traditional telephone using TCP/IP, the local area network becomes *the key piece* of telecommunications infrastructure.

Once that key piece is standardized within the enterprise, VoIP administrators have only one network to maintain—the one that supports TCP/IP. This means supporting a single network cabling system, rather than separate ones for voice and data. If you use wireless Ethernet, you don't need local area cabling at all—VoIP will still work. Meanwhile, old-school PBX administrators still have to maintain a separate local area cabling plant that serves only the PBX system.

But that key piece of telecom can be a key failure point, too. When the voice and data networks are separated, as they are in traditional telephony, their physical paths lie separately, protecting the voice system from failures isolated on the data network, and vice versa.

But with VoIP, these paths converge. When the path is broken by an equipment failure, a power failure, or a construction crew accidentally slicing underground cables, the data network fails. When a computer virus swamps your data network, VoIP phone calls may no longer be possible. When data fails, voice fails, too.

Even in the home, where you might rely on a cable or DSL Internet connection, your VoIP calling capability will swiftly disappear when your broadband provider's service fails or your power goes out.

VoIP Network Fundamentals

Since VoIP is layered on top of TCP/IP, you must have some form of TCP/IP network in order to use it. For small VoIP experiments, any Ethernet LAN will do, even

a hub-based or wireless one. But for larger, critical VoIP implementations, your choice of network infrastructure will be critical. For starters, using broadcast Ethernet devices like hubs is a poor choice, as is using early-generation Ethernet switches that lack quality-of-service features. Wide area equipment, like routers, will need to support these features, too. (Quality-of-service features are covered in detail in Chapter 9.)

Generally speaking, the faster your switches, routers, and network links are, the better your VoIP network will perform. Nothing beats good ol' speed for increasing the performance of a wide area network, but sometimes a slow network link is an economic or geographic necessity. VoIP is a speed-sensitive business, as you'll find out.

The Layers of a VoIP Network

Like other networks, VoIP networking can be described using the Open Systems Interconnect (OSI) reference model, a standardized way of describing the different parts of the data communications process. The OSI model has seven layers that represent each part: physical, data link, network, transport, session, presentation, and application. The purpose of the OSI model is to simplify connections between different types of networks and to allow engineers who design network applications to assume a standardized platform upon which to build.

The physical layer

The OSI physical layer is the most fundamental part of the datacom process. It's the layer that provides for the electrical, mechanical, radiant, or optical signaling pathways that are required in order to move data across any data network. In an IP network, the physical layer can include twisted-pair LAN cabling, plugs, cross-connects and patch panels, power sources, V.35 cables like those often used with serial interfaces on routers, and so on.

Though the physical layer is itself intended to be permanent and stable, its assortment of connective technologies (copper twisted-pair, fiber-optic cables, etc.) are prone to noise and distortion, two problems that cause data transmission errors. The physical layer has no way of dealing with these problems, and that's why certain guidelines related to distance and interference exist at the physical layer. For example, a 100BaseT Ethernet connection on twisted-pair copper cable cannot be longer than 100 meters.

The data link layer

Since the physical layer is not immune to the laws of physics and the signal degradation they incur, the data link layer provides a medium for detecting errors in data transmission. Error detection at the data link layer works on behalf of a single physical link, such as an Ethernet segment or a single T1 circuit.

The data link layer "frames" the continuous stream of signals flowing across a link. *Framing* means delimiting that signal into manageable pieces, called frames (imagine that). For error detection, each frame can be subject to a CRC, or cyclic redundancy check. With certain types of connections, error correction can be attempted.

The data link and physical layers are often viewed as one and the same, and in many network substrates, such as Ethernet, their functions tend to be inseparable. That is to say, you can't build an Ethernet physical layer without building its data link layer too—both layers are facilitated by the same device, which is usually an Ethernet interface, a hub, an Ethernet switch, or an Ethernet coax bus. The data link layer is the lowest layer that VoIP applications can reference, and usually only in an indirect manner (only quality-of-service functions interact with the data link layer—more on this in Chapter 9).

The network layer

While the data link layer provides data framing over a single physical connection, such as an Ethernet segment, the network layer provides the logistical intelligence required for a single network to exist across many physical connections—say, two Ethernet segments connected by a wide area connection. The network layer is more visible to applications than the data link and physical layers. Its job is to provide:

- A scheme for data routing across wide area links
- An addressing scheme, so that disparate physical connections can be referenced by higher-layer services and by each other
- A definition for connection-oriented and connectionless datagram structures

The network layer isn't the lowest layer that is relevant to VoIP, but it is the lowest layer that VoIP applications must reference in order to function. For example, datagrams and addresses—things implemented at the network layer—are critical to the functioning of VoIP applications.

The addressing scheme used by VoIP is inherited from IP. Each device on an IP network has an IP address, so each VoIP endpoint has one too. An IP address consists of 32 bits, commonly presented by four 8-bit figures separated by dots:

```
10.1.1.204
```

Each figure in the address allows for 256 values, so the overall address space (32 bits) of IP's addressing scheme allows for about 4.3 billion addresses. The newer version of IP, Version 6, allows for a 128-bit address space, but adoption of IP Version 6 has been slow, and this book deals strictly with the 32-bit address scheme of IP Version 4—the protocol that today's Internet runs on. In the context of the Internet and IP networks, the network layer is sometimes referred to as the *Internet layer*.

Using IP addresses, the network layer can facilitate wide area networking over dozens, hundreds, thousands, or millions of physical links. Consider the Internet, which uses

IP to connect millions of disparate networks. Individually, each of these networks tends to share a group of related addresses. Each group is what IP calls a *subnet*.

Every datagram sent across an IP network contains a source and destination address so that the devices responsible for maintaining the network layer know where to route the datagram. However, the network layer isn't responsible for any kind of error control—that's the job of the next layer up.

The transport layer

Even though the data link layer provides error detection on an individual network link, this alone isn't enough to satisfy the needs of a large, application-intensive network. That's why the transport layer provides error control across the entire network—from sender to receiver—regardless of the number of physical links between them. Transport layer error control operates independently of the measures provided by the data link layer, which tend to be specific to the type of link they are responsible for.

On the transport layer, protocols have been designed for two kinds of service:

- Datagram delivery is highly reliable, complex, and has high overhead.
- Datagram delivery is less reliable, less complex, and has lower overhead.

The kind of service elected depends on the needs of the application. Some applications don't need a high degree of reliability (video gaming, for example), while others must have absolute reliability (bank transactions). Within the transport layer, IP provides protocols—UDP and TCP—that handle both needs.

To connect or not to connect

Within IP, datagrams can be delivered using a "best effort" approach—that is, the host transmitting the datagram will not know whether it was received by the intended recipient. Also called *connectionless networking*, this method is employed by the User Datagram Protocol (UDP).

If you've ever played the Quake series of video games over a network, you've used the UDP Protocol. UDP excels in situations in which very fast delivery of data is a requirement, and reliability features, like confirming that the data has been delivered, would be a waste. In a multiplayer network game such as Quake, you and the other players each control an armed character that is trying to kill the others in a virtual 3D world. Real-time delivery of characters' location and trajectory data within the virtual world is critical to the game play. Even a slight delay in delivery of these datagrams could mean life or death for your Quake warrior. Delivery guarantees impose too much overhead—because dozens of UDP datagrams can be used by Quake in a second.

The same is true of the traffic carried over the network during VoIP phone calls. This traffic is carried across the network at a rate of between 30 and 50 datagrams per second. To verify delivery of each one would introduce a performance bottleneck that is unacceptable in a voice application. Therefore, almost all voice data flowing across a VoIP network is considered connectionless and carried by UDP.

The more reliable protocol for data transmission in an IP network is called Transmission Control Protocol, or TCP. Like UDP, TCP is encapsulated within IP. TCP's distinguishing characteristic is that transmitters using TCP must set up a transmission channel, or connection, before they send data to their receivers. For this reason, TCP is considered a connection-oriented protocol.

Error control takes place during a TCP transmission. At the end of the transmission, the sender and receiver agree to end their conversation, and the connection is closed. TCP guarantees that packets will arrive in the correct order, too. Because TCP is so cautious compared to UDP, it isn't normally used to carry voice data, but it can be used to carry call-signaling data: the bits of information that a VoIP network uses to establish, monitor, and end calls. TCP datagrams are called packets, though you often hear people refer to UDP datagrams as packets too.

IP provides both connection-oriented (TCP) and connectionless (UDP) network protocols at the transport layer, which allows it to replace both functions of the PSTN: voice transmission and call signaling.

The session, presentation, and application layers

Operating systems, end user applications, application services (like DNS), and user interfaces are provided at the topmost layers of the OSI reference model. Your interaction with a computer system or a network is most directly affected by the systems running at the application layer. The application layer's job is to take input and drive underlying functionality down through the other six layers without you, the user, having to know the details of what's going on down there.

In a VoIP network, the user interface to the telephony functions—often just a telephone receiver with a 12-key dial-pad—is provided at the application layer. A VoIP-adapted OSI model is shown in Figure 2-1.

A VoIP network is a set of networked applications and endpoints (agents that allow humans to use the applications), just as the World Wide Web is a set of networked applications and endpoints. In the case of the World Wide Web, the applications are web sites, and the endpoints are web browsers that request and display web pages. But in the case of a VoIP network, the applications are telephone calls, conference calls, voice mail, automated attendants, and even video conferencing or text messaging, while the endpoints are traditional telephones, IP phones, and software phones (softphones) that run on PCs.

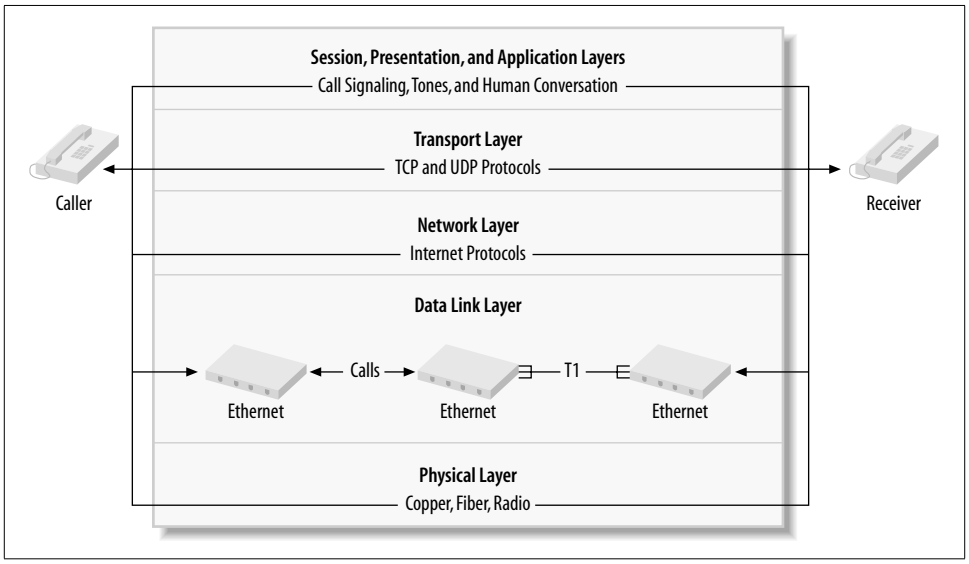


Figure 2-1. OSI reference model layers

On the WWW, web sites are hosted using software like Apache, a web server. This software communicates with the endpoints (the browsers) in order to facilitate user interaction with the application (web sites).

This model works much the same with Voice over IP. In a VoIP network, specialized servers, which we'll call VoIP servers for now, communicate with the IP or traditional phones (endpoints) in order to facilitate calling (the application).

VoIP Servers

VoIP servers, which are software-based devices that direct or participate in Voice over IP data conversations in order to facilitate calling and other VoIP applications, are usually connected to the network using Ethernet. An ATA is itself a highly specialized VoIP server and so is a VoIP-enabled voice mail system.

Carrier-class users of VoIP might have reasons to connect VoIP servers to a different type of data link, such as ATM (asynchronous transfer mode), but most enterprise implementers will use Ethernet.

VoIP servers fulfill many telephony roles:

- Call switching and connection management, such as a traditional PBX. A VoIP server in this role is usually called a *softPBX*.
- Call recording and autoattendant functions, such as a traditional voice mail system.
- Call conferencing, such as a traditional teleconferencing service.

- Access interfacing, so that traditional phones and PBXs can participate in the VoIP network by way of media conversion.
- Translation of audio encoding standards (codecs) in real time to facilitate calls between endpoints that have different audio capabilities or between analog, digital, and IP endpoints.

When VoIP endpoints and servers are connected to the same IP network, VoIP becomes the call-switching and voice-transmission mechanism, replacing the traditional PBX.

What differentiates VoIP servers from voice endpoints is whether they provide a user interface for the telephony application. Phones do, so they are endpoints. Switches, ATAs, PSTN gateway devices, and other specialized VoIP devices don't, so they are VoIP servers. Another differentiator between endpoints and servers is their abundance on the network. Like the WWW, there are more endpoints than servers in a VoIP system, sometimes even in thousands-to-one ratios.

Voice Endpoints

Endpoints that are TCP/IP aware (that is, they are valid hosts on the IP network) and connect directly to a data link that carries TCP/IP (such as Ethernet) are usually called IP phones. *IP phones* resemble feature-enriched business telephones but differ in that they usually have an RJ45 twisted-pair Ethernet connection rather than an analog or digital loop connection. IP phones can be plugged directly into an Ethernet hub or switch using an Ethernet patch cable or through a cabling distribution frame like those found in many offices. Usually, IP phones have a 10/100BaseT autonegotiating interface, much like a desktop PC Ethernet adapter.

The voice applications that run on an IP phone facilitate calling in a manner similar to a traditional phone, but the mechanics of call signaling and voice transmission are worlds apart from the old-school telephony world, as you'll discover.

Even though they don't use RJ45 twisted-pair connections or have Ethernet smarts onboard, traditional analog telephones can be connected to Ethernet, through the use of an ATA (analog telephone adapter). The ATA is a device that converts the single-pair RJ11 analog connection into a four-pair 10/100BaseT Ethernet interface, as in Figure 2-2. ATA devices tend to be less expensive than IP phones. They provide fewer telephony features, too—after all, that old analog phone can't really run sophisticated VoIP applications, even with an ATA attached, because it doesn't contain integrated circuits or programmable components. In some cases, the limited functionality provided by a traditional analog telephone is enough.

IP telephones and ATAs are both hosts on an IP network. Like other TCP/IP hosts, they must have IP addresses and be compliant with the design of your IP network.

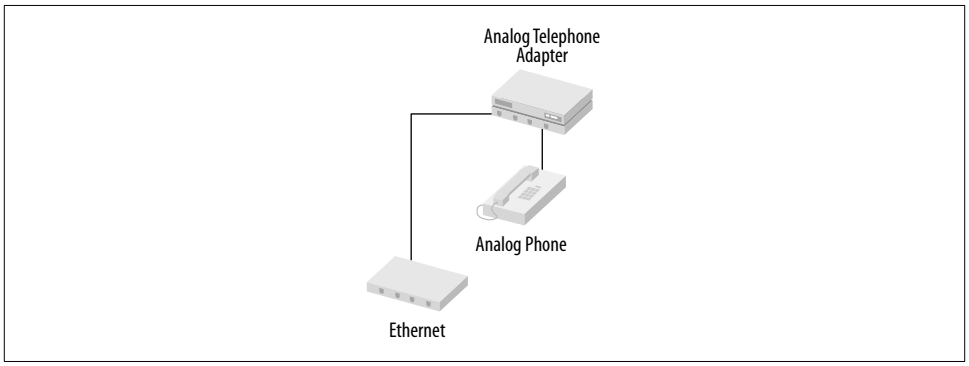


Figure 2-2. Analog phones are voice endpoints, and they can be used with VoIP networks in tandem with an ATA device

IP phones or traditional phones

The phones you choose are often determined by your project budget—traditional phones are often already there; it's just a matter of VoIP-enabling them. This can mean less capital investment during your transition to VoIP. But your choice of endpoints is also affected by your feature requirements: IP phones tend to have greater feature sets and programmability than do traditional phones.

Another big advantage of IP phones is that they can be completely software-based, so that they run as an application on a TCP/IP-aware Windows, Mac, or Linux PC. This can be appealing to mobile users who want to maintain a familiar look and feel on their telephone wherever they go. As long as the user has access to the Internet, the "softphone" can be made to function just like an IP "hardphone."

If you choose to integrate existing traditional phones into your VoIP network, then you must continue to support the wiring that is required for those phones. Often, this wiring is very simple, like the one-pair copper wiring used to power a simple analog phone, but not always. Most traditional PBX-type phone systems use telephones that require two pairs of copper and a digital signal bus—these are called, simply enough, digital phones.

Wireless IP phones can solve the challenges of wiring, but they introduce some challenges of their own. Since wireless Ethernet does not yet offer quality-of-service mechanisms or a high yield for simultaneous use, the abundance ceiling of wireless IP phones is much lower than their wired cousins. Wiring aside, using traditional cordless phones is one way to maintain mobility without the restrictions of wireless Ethernet.

If you choose to use wired IP phones, then your Ethernet wiring system must be sufficient to support them. 100BaseT Ethernet and Category 5e cabling are considered the minimum for connecting IP phones. For this reason, exclusively using IP may not

always be possible. Not all sites where you want to use IP phones have Ethernet wiring in place.

If you had an old 10Base2 Ethernet segment at a certain site, you wouldn't be able to connect any IP phones at that site, because there are no IP phones on the market that support the physical layer interfaces (BNC coax connectors) required by 10Base2 Ethernet. Moreover, 100BaseT is the only Ethernet spec that allows for the right quality-of-service required to support a large rollout of IP phones.

Project 2.1. Configure an IP Hardphone and the VoIP Test Network

To complete this project, an IP hardphone is needed. We'll use the Grandstream Budgetone 100 series phones in this example.

What you need for this project:

- Grandstream IP phone
- LAN

IP phones are really just software applications that speak VoIP's protocols: SIP, SCCP, H.323, or MGCP for call signaling; RTP for audio transmission; and sometimes LDAP for directory integration. They may also include XML or Java services so that their displays and buttons can be used to further enhance the end user's telephony experience. When bundled together, this suite of protocol software that comprises an IP phone can run either on a PC, what we call a softphone, or on a specialized chassis whose enclosure looks like a traditional telephone, what we call a hardphone (see the previous section for more details).

Hard or soft, IP phones all require a TCP/IP stack to support data networking as well as a physical interface to the network. In a softphone, these are provided by the PC's operating system and networking hardware. In a hardphone, they are embedded more tightly and are less visible to the user.

Like a PC with a 10/100 Ethernet interface, an IP hardphone has an RJ45-compatible jack, so the first step in getting a hardphone online is connecting a patch cable between that jack and an Ethernet switch. This step will be largely the same regardless of the make and model of IP phone you're using.

Next, the IP phone must be given a TCP/IP host configuration that is workable on the network to which it's connected. To configure an IP phone for the network, you'll need:

- A DHCP-assigned or statically assigned IP address
- A DHCP-assigned or administrator-designated IP subnet mask

- A default gateway address (optionally assigned by DHCP)
- The address of a DNS (domain name service) server that serves this network

The IP address used by the phone can be static, or it can be dynamically assigned using DHCP (Dynamic Host Configuration Protocol) if you have a DHCP server on this Ethernet segment. DHCP is not necessary in a small environment with very few IP phones to keep track of. It becomes a necessity in larger environments where an administrator mistakenly assigning the same address to two different phones can cause a disruption, just as in PC networking. For now, we'll use static addresses.

The specific steps required to configure each make of IP phone varies depending on the administrator features and firmware of each. Most permit rudimentary network configuration using the buttons on the phone itself. The Grandstream Budgetone 101 telephone is an entry-level SIP-based IP phone, and its initial configuration is done in this manner.

SIP is the Session Initiation Protocol, a standard for call signaling and capabilities negotiation. It is covered more extensively in Chapter 7.

Configuring a Grandstream Budgetone 101 IP phone

The Budgetone 101 phone has a Menu key, two arrow keys, and an LCD display, which are used to navigate its configuration menu options: DHCP, IP Address, Subnet Mask, Router Address, DNS Server Address, TFTP Server Address, Codec Selection Order, SIP Server Address, and Firmware Versions (called *Code Rel* on the phone's screen). When you get to the option you want, you press the Menu key to select it, and then enter the numeric data required for each option using the keypad. Use this menu only to set up the IP address, subnet mask, and router (default gateway) address.

To get the phone enabled for the next configuration step, turn DHCP off, and assign an IP address, subnet mask, and router address.

More advanced configuration is performed using the Budgetone's built-in web configuration tool. When you access the IP address you assigned to the phone using your web browser, you'll be prompted to log in to the phone, as in Figure 2-3. The default password is "admin."

Then, you'll be confronted with a big page of configuration options like the one in Figure 2-4. Many of the options are available only through this interface, not from the phone's keypad menu. For this project, the only settings we're concerned about are the codec selection ones. Configure the first (highest-priority) codec to be "PCMU" if you're in North America or "PCMA" if you're elsewhere in the world. That's all we're going to cover about codecs for now. After applying any configuration changes, the Budgetone needs to be power cycled.

Figure 2-3. The Budgetone's web configuration login page

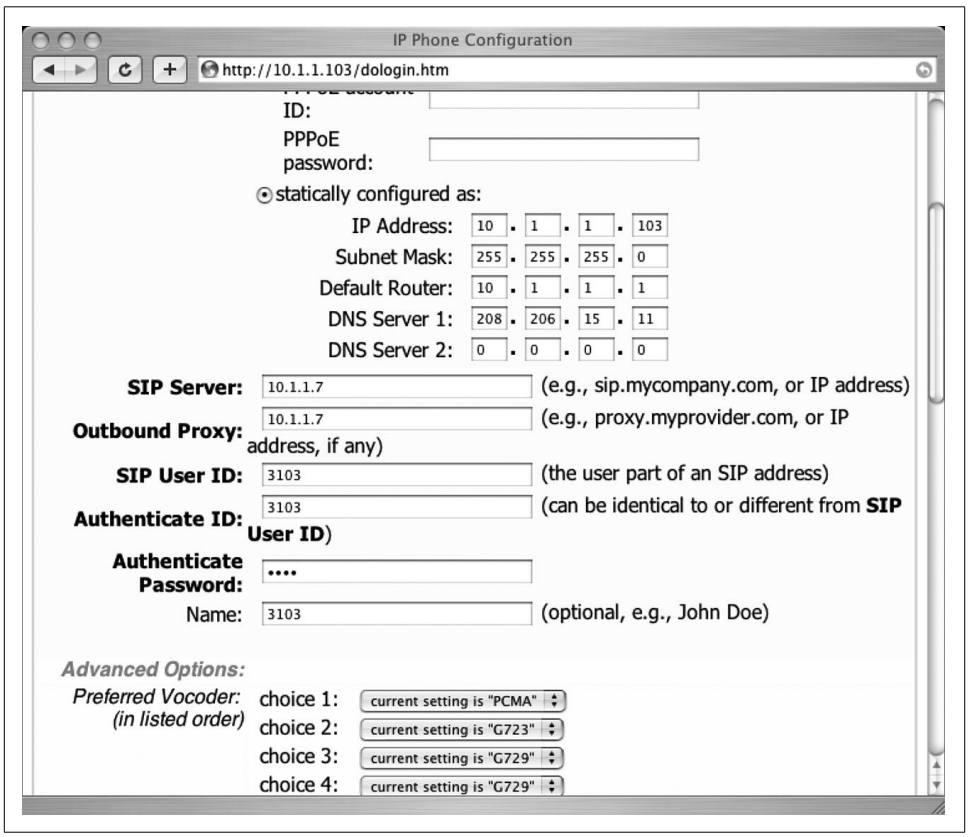


Figure 2-4. The Budgetone's main configuration page

Some IP phones offer a Telnet interface rather than a web-based one. To use these tools, one must connect to the phone with a Telnet client instead of a web browser. In any event, once the network configuration is set on the IP phone, ping its address from another host on the same network subnet to make sure it's speaking TCP/IP.

A simple VoIP test network

Throughout this book, a TCP/IP network is used to illustrate Voice over IP concepts through projects and hacks. This network, illustrated in Figure 2-5, is structured as follows:

- IP hardphones have an IP address of 10.1.1.100–150.
- IP softphones and ATA devices have IP addresses of 10.1.1.200–250.
- VoIP servers and nonendpoint devices, like proxies, have an IP address range of 10.1.1.10–29. The Asterisk server we use will always be 10.1.1.10.
- The default gateway router's address is 10.1.1.1.
- The subnet mask for all devices is 255.255.255.0, giving our test network a maximum size of 254 possible devices or an 8-bit subnet.
- DHCP will not be used, except as noted in specific projects.
- The test network will always use wired, switched Ethernet, unless specifically noted.
- It will consist of one segment, or one Ethernet LAN, unless specifically noted.
- This test network requires access to the Internet for many projects. To accomplish this, use a NAT firewall or Internet access appliance.

Many VoIP devices need access to a time clock. The NTP (network time protocol) server we've chosen is `time.nist.gov`. More NTP servers are available from the list at *http://www.nist.gov*.

Project 2.2. Make an IP-to-IP Phone Call

For this project, you'll need two IP phones. Our scenario uses two Grandstream Budgetone 100 series phones configured as directed in Project 2.1. Most IP phones permit a type of IP-to-IP calling similar to what's described here, so you can replicate an IP-to-IP call using a different make of IP phone.

What you need for this project:

- Two Grandstream IP phones
- LAN

With both IP phones connected to the same Ethernet switch or directly connected (to each other) using a crossover patch cable, make a note of the IP address you've

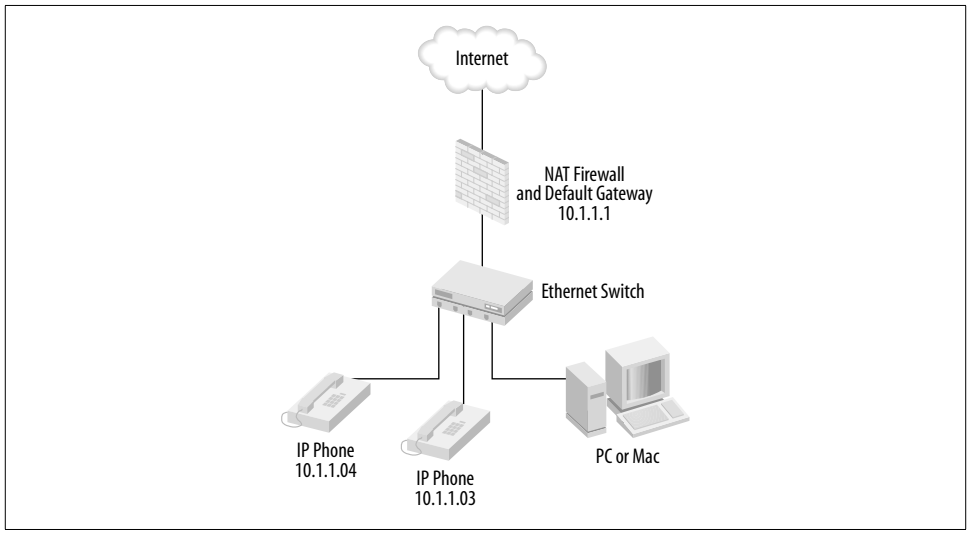


Figure 2-5. The VoIP test network at the end of Project 2.1

established for each. In this example, we'll use 10.1.1.103 for the receiver and 10.1.1.104 for the caller. If you have your phones configured for DHCP, give them this static configuration instead.

The Budgetone can place IP telephone calls from one IP endpoint directly to another without the need for a VoIP call-management server. This is called *IP-to-IP calling*. Since each IP phone has a uniquely identifying characteristic within the scope of the network—an IP address—one phone can call the other by IP address as if it was a phone number.

To do this, first make sure there is nothing set for User Name or SIP User in the Budgetone's configuration page. That is, make sure they are both blank, apply the changes if necessary, and then power cycle the phone.

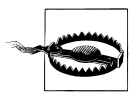
Now, to dial by IP address. All IP addresses are 12 decimal digits long, even if leading zeros aren't written out. Conversely, the dots normally included in an IP address are *not* dialed. So, on the Budgetone phones, 10.1.1.103 is dialed as:

```
010 001 001 103
```

To dial, take the phone off the hook so you hear a dial-tone. Then, press the Menu key, dial the address of your second phone according to the convention shown earlier, and press the Send or Redial button. Of course, nobody would want to dial 12-digit IP addresses in order to place phone calls all the time; call-management servers, like SIP registrars, provide more elegant dialing conventions. Dialing by IP address, in this case, allows you to circumvent call management and make a direct VoIP connection between two endpoints.

When the receiving phone rings, have somebody answer the call. If you can hear the other person talk through your IP phone's handset, you've just made your first successful VoIP phone call—sort of the IP equivalent of Bell's and Watson's first phone call back in 1876.

If the receiving phone doesn't ring, then you should check the IP address you dialed, check the phone's configuration to make sure it is listening on the default port for SIP (5060), and make sure SIP registration is turned off. These options, which are accessed in the Budgetone's web configuration page, will be covered in greater detail later.



Dialing by IP address isn't user-friendly, and it isn't practical at all in a DHCP environment, let alone an enterprise or home phone system. Outside your test lab, you'll use it only for troubleshooting.

Distributed Versus Mainframe

In the world of traditional telephony, endpoints and PBXs interact in a manner similar to dumb terminals and mainframe computers. That is, the PBX (or mainframe) has all of the application functionality built in, and the user interface functions of the endpoints (or terminals) are dictated by the PBX.

With IP telephony, voice endpoints are far more programmable, lessening the requirement for centralization. VoIP endpoints don't always have their functions dictated by a particular VoIP server. In fact, VoIP endpoints may interact with many services on many different physical servers: DNS, LDAP, SIP, and RTP are all VoIP-related application protocols that may be facilitated by separate servers or by no servers at all (some operate between two endpoints and don't require a server in between). The IP-to-IP call placed in Project 2.2 is a good example of that.

Compared to a traditional telephone call, which must always be routed through a telephone switch such as a PBX, this is a significant difference. A traditional telephone call is set up, torn down, and accounted for using the same piece of hardware—the PBX. Moreover, the sounds of the conversation are routed through the PBX, because the PBX is the circuit-switching mechanism that provides the voice loop between caller and receiver. This is illustrated in Figure 2-6.

But in a VoIP network, the call-management functions are separated much more from the voice transmission functions. This allows each function to be enabled through separate network resources, as shown in Figure 2-7. Call management could occur over a wide area link, while the voice transmission could occur directly between two endpoints on the same local area link, in order to preserve capacity on the wide area link. The net result is that a single, powerful call-management server could work on behalf of many remote sites, increasing the value of the WAN and possibly saving money that might ordinarily be spent on remote PBX systems to support each site.

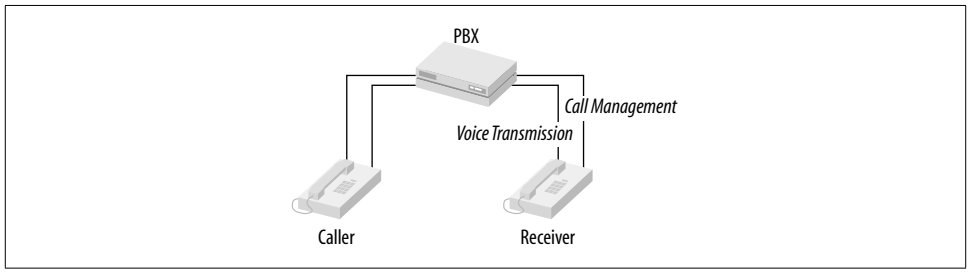


Figure 2-6. With a traditional PBX, voice transmission and call management are dependent upon a route through the voice switch

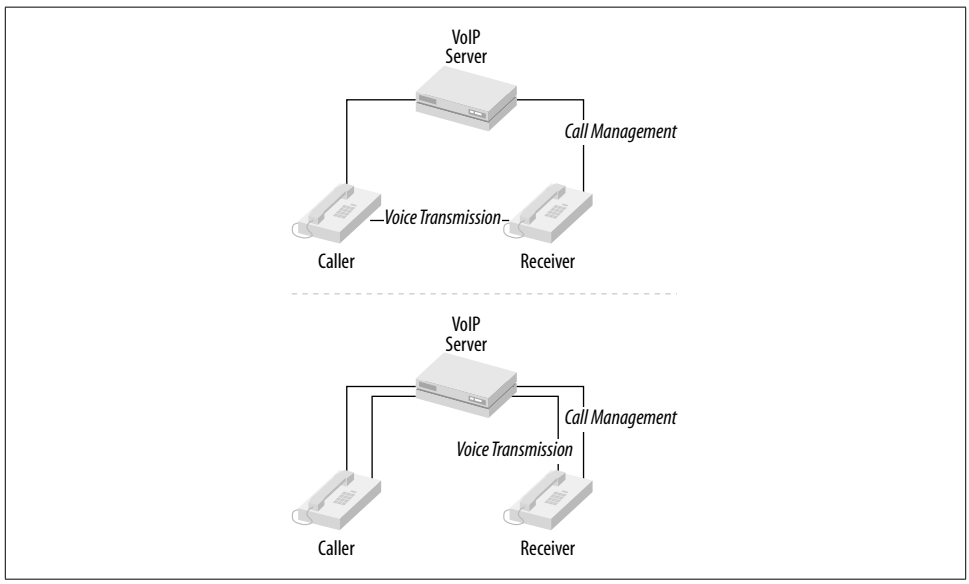


Figure 2-7. In IP telephony, call management and signaling can be separated from voice transmission

The distributed nature of VoIP applications makes them preferable to traditional telephony on a wide area network—but that's not the only advantage of VoIP on a WAN. The other great benefit of VoIP—especially in a bandwidth-conservative WAN—is compression.

The Core and the Edge

At the heart of a network resides the *core*, or network backbone. In modern IP networks, the core serves the purpose of transporting high levels of aggregate traffic between nodes that are probably not endpoints—that is, they aren't the hosts where the traffic originated or the hosts where it is headed, but rather hosts whose purpose is to forward that traffic along the core network until it needs an exit along the route to its destination.

The core is kind of like the 10-lane interstate highway: a lot of people drive on it, but nobody's driveway is an entrance ramp to it. So, while billions of hosts may send and receive data that crosses the Internet core (backbone), almost none of those hosts are directly connected to the core.

Instead, IP network endpoints connect to disparate network links that share high-capacity aggregate connections to the core. These links are collectively known as the *edge*. The edge is like the surface streets that surround the 10-lane interstate highway. Most traffic that ends up on the big highway originates from the surface streets.

A key difference between distributed and mainframe computing follows this analogy: in a mainframe environment, such as the PSTN, all the endpoints have a direct connection to a core—the central office switch. Likewise, in a PBX system, all the endpoints have a direct connection to a core—the PBX switch. So, all the driveways in a mainframe town are actually entrance ramps right onto the big highway.

VoIP facilitates the build-out of the networking smarts that normally exist at the traditional PSTN core, so that application functionality gets closer and closer to the edge of the network. This is similar to the way distributed PC applications have been displacing mainframe client/server applications over the past 20 years.

With VoIP, the core network is still there, and very much required, but it serves a different purpose than the core network of the PSTN. In a VoIP environment, the core is mainly used to move data back and forth, and the programmatic functionality of voice applications exists in a distributed model of peers: VoIP servers and endpoints. These peers can reside anywhere on the edge and offer new and changing features, without requiring changes at the core.

In traditional telephony, that isn't the case. The PSTN's core is itself responsible for all of the features available to you as a telephone company customer or enterprise PBX user, and offering new features can require the phone company or enterprise PBX administrator to alter the core network.

VoIP in Enterprise Networks

VoIP can be used to connect IP phones on an Ethernet segment to a VoIP server that is used for call management, and that VoIP server can be used to provide a connection for those phones to the PSTN, as in Figure 2-8.

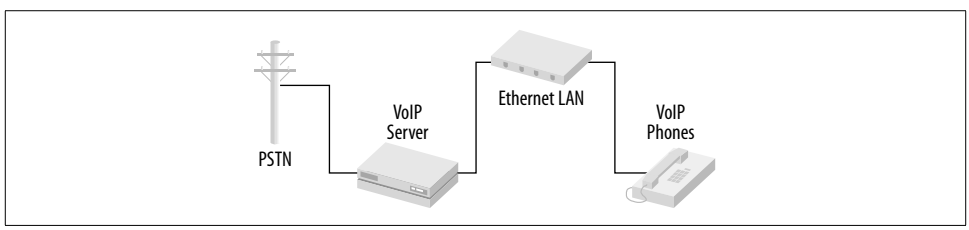


Figure 2-8. A VoIP server can be a PSTN gateway for IP phones connected via Ethernet

A single VoIP server can act as a PSTN gateway for IP phones on Ethernet segments located at remote offices, as long as WAN connectivity exists between them. This way, the IP phones at all the sites can call one another, and the VoIP server routes calls between the offices and to the PSTN, as in Figure 2-9.

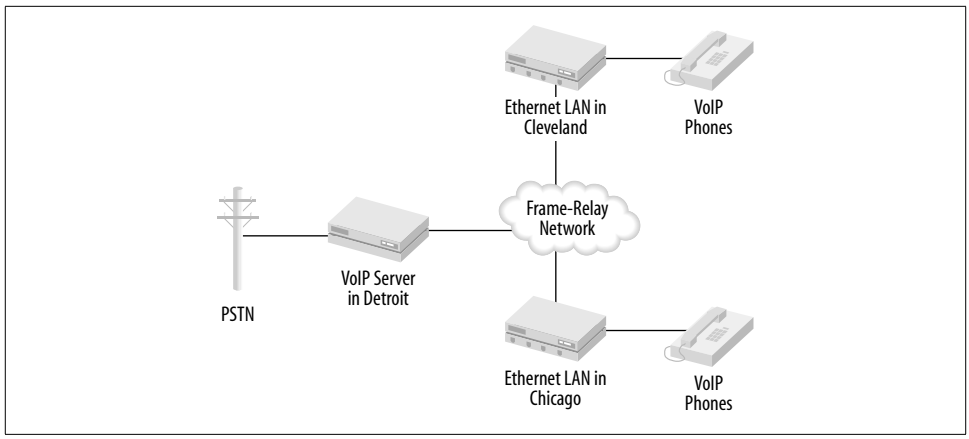


Figure 2-9. A VoIP server can be a PSTN gateway for many IP phones on a wide area network

If a large company uses a conventional PBX at every site around the country, all can be linked together using VoIP over a WAN. This way, each PBX can connect calls within its local network of traditional phones, as well as calls between them and the PSTN, but calls placed between phones on opposing PBXs can be routed over the WAN using VoIP, as in Figure 2-10.

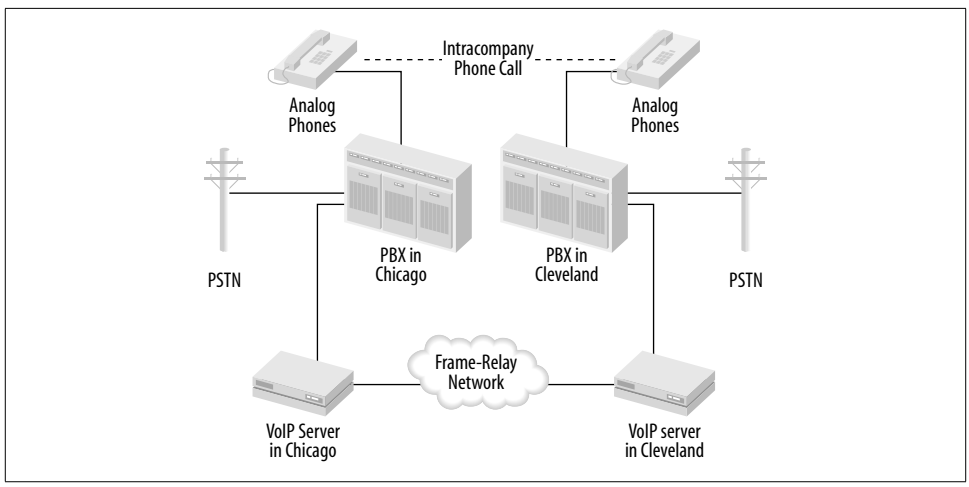


Figure 2-10. VoIP servers can use a WAN to connect calls between PBXs in different offices (switch-to-switch trunking)

At a minimum, at least two VoIP devices (such as an IP telephone and a VoIP server or two VoIP servers) and at least one form of connectivity are required by all VoIP solutions.

Like the network, VoIP is a conversation-oriented technology. Its protocols are simply rules that devices and software must follow in order to carry on the conversations required to make VoIP applications work—that is, to carry each human speech conversation. Each VoIP protocol set (H.323 and SIP are the two big sets) has its own rules that enforce proper conversation, just as Congress has a parliamentary procedure that enforces its debates. The biggest rule is the definition of VoIP's minimum requirements: two or more TCP/IP hosts using one common protocol and connected data links.

Network Convergence

When you support only one transport (in VoIP's case, TCP/IP) for all networked applications, including telecommunications, you've achieved complete convergence. The more you leverage your TCP/IP network to support voice and multimedia telecom apps, the more you converge. Theory tells us that convergence increases administrator productivity, and experience tells us that support costs drop the more voice and data networks are converged.

Convergence isn't something that has to happen overnight. There may be plenty of reasons you don't want total convergence: capital that is tied up in perfectly good legacy hardware is one; network readiness is another. As with many past paradigm shifts in networking, a migration path exists to get you from partial to total convergence. One of these paths is the "hybrid" voice switch.

Pure IP or IP Enabled

Pure IP voice switches can't make direct use of traditional circuit-switched telephones and trunks. Vendors that refer to the VoIP solutions as *pure IP* mean that the phones and trunks connected to their switch are totally packet-based. Connections to outside non-IP systems, like the PSTN, are accomplished through outboard hardware that facilitates transmission of call signals to the call-switching server using IP. In this fashion, vendors whose switching servers support only IP endpoints and not traditional endpoints tend to use the pure IP moniker. Cisco's CallManager 4.0 is a good example of what pure IP means—it's a completely software-based switch that requires outboard hardware, called a media gateway, in order to support non-IP endpoints. As you can see in Figure 2-11, any devices that communicate with a pure IP PBX do so using the TCP/IP Protocol trunked over Ethernet.

IP-enabled voice switches, unlike pure IP systems, offer support for all kinds of voice endpoints and make no bones about connecting to analog phones and trunks like those from the PSTN. Analog, digital, and IP devices can all connect, as shown in

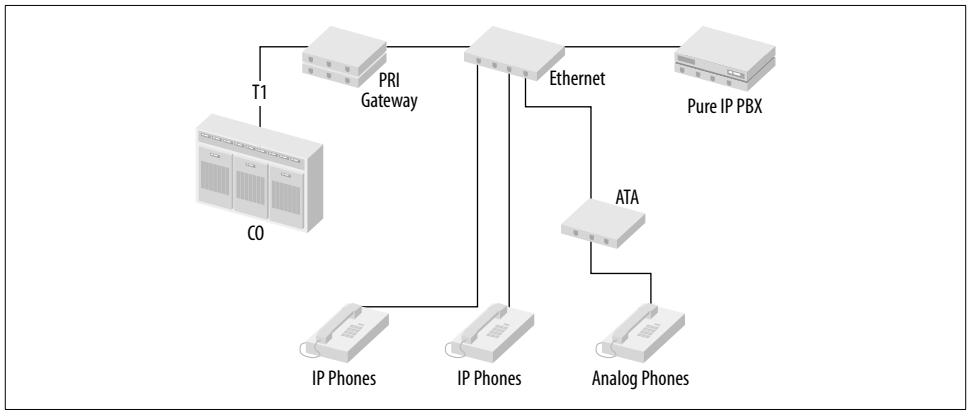


Figure 2-11. A pure IP switch has only IP-based trunks; all trunks that feed the same switch are carried by TCP/IP

Figure 2-12. The media interfacing required to use traditional telephony devices with an IP-enabled switch is all contained within the switch chassis, often using a single digital bus and microprocessor, much like a conventional PBX. Good examples of software-based IP-enabled switches are Avaya's Communication Manager 2.0 and Digium's Asterisk (an open source solution), both of which run on Linux. Sometimes VoIP implementers refer to IP-enabled switches as *hybrid* switches.

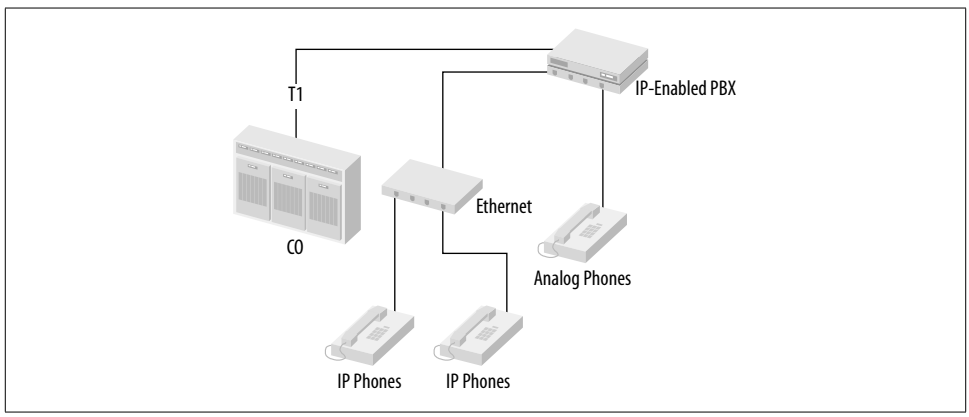


Figure 2-12. An IP-enabled voice switch supports IP-based, digital connections like T1 and analog connections

Key Issues: Voice over Data: Many Conversations, One Network

- VoIP can replace traditional telephony, but quality-of-service measures are required in order to make it as reliable as old-school gear.
- The OSI network model breaks down VoIP in terms of layers. The networking aspects run at the lower layers, and the application aspects run at the higher layers.
- VoIP media streams are delivered by connectionless UDP datagrams, and not TCP packets. This is because, in telephony and other real-time media applications, there's no point in error correction. VoIP administrators would rather strive for full error *abatement*. This means designing an IP network to carry voice, not just data.
- Most IP phones allow simple calls to be made directly to each other, dialed by IP address, without the need for a VoIP PBX server as an intermediary. The job of the server, among other things, is to provide a human-friendly addressing scheme and other features that the phones alone can't provide.
- Traditional telephony networking is characterized by client/server or mainframe-like tendencies. VoIP networks are characterized by distributed or fat-client tendencies.
- Most IP endpoints sit at the proverbial "edge" of the network, where PCs and printers also reside.
- Pure IP voice systems don't use any legacy interfacing or protocols—such as POTS or T1. Rather, they support only VoIP protocols and offload the media conversion required for such interfacing to other devices.
- IP-enabled, or hybrid IP, voice systems offer server-based interfacing for legacy links while also providing VoIP signaling, usually in one server chassis.