CHAPTER 1

Troubleshooting Methodology and Approach

It's 5:30 a.m. on a Monday and your pager goes off. You recognize the phone number—it's your CEO's administrative assistant. As the administrator of the company's 8000-phone IP Telephony network, you assume there's a big problem. You rush into work and find the CEO's administrative assistant, who states that several calls for the CEO have been disconnected in the middle of the call, including a call from a very important customer. Where do you start?

Troubleshooting a Cisco IP Telephony network can be a daunting task. Rather than describing step-by-step how to solve specific problems (subsequent chapters provide that information), this chapter focuses on teaching a good troubleshooting methodology: learning how to find clues and track down your "suspect" by breaking the problem into smaller pieces and tackling each piece individually.

A typical IP Telephony network consists of—at the very least—one or more of the following components:

- Cisco CallManager servers
- IP phones
- Voice gateways

These components are in addition to the data network infrastructure that supports voice over IP (VoIP) traffic. More-complex installations can have dozens of servers for different services and redundancy, each server running a variety of applications, as well as hundreds or thousands of IP phones and a large number of voice gateways.

Before exploring the myriad of tools, traces, and techniques available to you that aid in troubleshooting, you must develop a systematic method by which you can focus on the problem and narrow it down until you determine the root cause.

In addition to the information in this book, you should become familiar with the various standard protocols that are used in an IP Telephony network, such as the following:

- H.323
- Media Gateway Control Protocol (MGCP)
- Telephony Application Programming Interface/Java Telephony Application Programming Interface (TAPI/JTAPI)

You should also become familiar with the protocols used when interfacing with the traditional time-division multiplexing (TDM)-based Public Switched Telephone Network (PSTN), such as the following:

- Q.931 (an ISDN protocol)
- T1- or E1-Channel Associated Signaling (T1-CAS or E1-CAS)
- Foreign Exchange Office (FXO)
- Foreign Exchange Station (FXS)

Additionally, because an IP Telephony network runs over a data network, it is important to understand the protocols that transport VoIP data, such as the following:

- Internet Protocol (IP)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)
- Real-Time Transport Protocol (RTP)

Later chapters cover some of these concepts. However, each of the mentioned protocols could take up an entire book on its own, so you should refer to the specifications and RFCs or to other materials that go into detail about these protocols. Appendix A, "Cisco IP Telephony Protocol and Codec Information and References," provides references to where you can find additional information for each protocol discussed in this book.

On the other hand, because the Skinny Client Control Protocol (SCCP or Skinny protocol, the Cisco-developed protocol that Cisco IP Phones use) is not the product of an industry-wide standards body, this book goes into additional detail about how this protocol works. Understanding the Skinny protocol is essential to understanding how the phone operates and how to troubleshoot problems with it. The Skinny protocol is covered in greater detail in Chapter 5, "IP Phones."

Developing a Troubleshooting Methodology or Approach

To track down a problem and resolve it quickly, you must assume the role of detective. First, you need to look for as many clues as you can find. Some clues lead you to additional clues, and others lead you to a dead end. As soon as you've got all the clues, you need to try to make sense of them and come up with a solution. This book shows you where to look for these clues and track down the problem while trying to avoid as many dead ends as possible.

Troubleshooting a problem can be broken down into two stages: data gathering and data analysis, although your analysis might lead you to collect additional data. The following list is a general guide for steps to take when troubleshooting an IP Telephony problem:

Step 1 Gather data about the problem:

(a) Identify and isolate the problem.

(b) Use topology information to isolate the problem.

(c) Gather information from the end users.

(d) Determine the problem's timeframe.

Step 2 Analyze the data you collected about the problem:

(a) Use deductive reasoning to narrow the list of possible causes.

(b) Verify IP network integrity.

(c) Determine the proper troubleshooting tool(s), and use them to find the root cause.

Production Versus Nonproduction Outages

Troubleshooting a problem can occur in one of two timeframes:

- During a scheduled outage window, such as when you're installing a new system, adding components, or upgrading for new features or functionality
- During production hours when the problem affects end users or service

Although the methodology to troubleshoot problems in either of these two situations is similar, the focus on how to resolve the problem should be different. In the case of a service-affecting problem during production hours, the focus should be to quickly restore service by either resolving the problem or finding a suitable workaround.

In contrast, when a problem is found during a new install or scheduled outage window, the focus should be on determining the root cause to ensure the problem is completely diagnosed and resolved so that it does not have the potential to become service-affecting.

For example, if users are encountering a delayed dial tone or sluggish behavior on their phones, you might discover that a high-level process on CallManager is consuming 100 percent of the CPU on one of the servers. During a new install or scheduled outage window, it's a good idea to investigate what is causing the CPU consumption to ensure that the problem does not return during production hours.

However, if this problem occurs during production hours, the best approach is to stop or restart the offending process and let the redundant systems take over to quickly restore service. After you restore service, perform a root-cause analysis to try to determine why that process was consuming the CPU. The downside of this approach is that you might not be able to further troubleshoot the problem when the process is restarted. Fortunately, CallManager provides many diagnostic traces (if they are enabled prior to the problem) that you can reference after a problem has occurred to see what was happening on CallManager at the time of the problem.

Note that although 100 percent CPU of a high-level process can cause sluggish behavior or delayed dial tone, do not infer from this that 100 percent CPU is necessarily always a bad thing. As of CallManager 3.3(1), low priority tasks (such as phone registrations) can consume 100 percent CPU without causing adverse effects to the ability to place or receive calls. Look at the 100 percent CPU as a possible symptom but not necessarily the root cause. In this case, you observe the symptoms of sluggish or delayed dial tone and 100 percent CPU utilization and make a correlation between the two.

If you encounter an event where you are unable to determine the root cause due to insufficient information, it is a good idea to turn on the appropriate traces to ensure that if the problem reoccurs, you will have enough data to identify the root cause.

Sometimes, several service-affecting problems occur simultaneously. In fact, this is not uncommon, because multiple problems often manifest themselves as symptoms of the same root cause. When multiple problems occur simultaneously, focus on the problem that has the greatest impact on users. For example, if some users are reporting dropped calls and others are reporting occasional echo, the two problems are probably unrelated. Troubleshoot the dropped-call problem first because keeping calls connected is more critical than removing the occasional echo on an active call.

Step 1: Gathering Data About the Problem

So you've just installed a new IP Telephony network, or you've been given the task of maintaining one—or maybe you've taken your first CallManager out of the box and are having problems getting it to run. You've encountered a problem. The first thing to do is gather as much information about the problem as possible.

Identifying and Isolating the Problem

Half the battle in troubleshooting a problem is determining which piece of the puzzle is the source of the problem. With so many different pieces composing an IP Telephony network, the first step is to isolate the problem and, if multiple problems are being reported, determine which of the problems might be related to each other and which should be identified as separate problems.

You must also determine which parts of the problem are symptoms and which are the root cause of the problem. For example, if a user complains of a phone resetting itself, it might seem logical to first assume that something is wrong with the phone. However, the problem might lie with CallManager or one of the many routers and switches that make up the underlying data network. So although the symptom is a phone reset, the root cause could be a WAN network outage or CallManager failure. You must always remember to look at the big picture when searching for the root cause and not let the symptoms of the problem lead you in the wrong direction. To help you visualize the big picture, detailed topology information is essential.

Using Topology Information to Isolate the Problem

You can take many proactive steps to help make the troubleshooting process easier. One of the first lines of defense is possessing current topology information. One of the most important pieces of topology information is a detailed network diagram (usually created using Microsoft Visio or a similar application). The network diagram should include network addressing information and the names of all the devices. It should also clearly show how the devices are interconnected and the port numbers being used for these interconnections. This information will prove invaluable when you try to isolate which components are involved in a particular problem.

For medium- to larger-sized networks, you should have a high-level overview topology that gives you a general idea of how things are connected and then several more-detailed diagrams for each piece of the network that drill down to the interface level on your network devices.

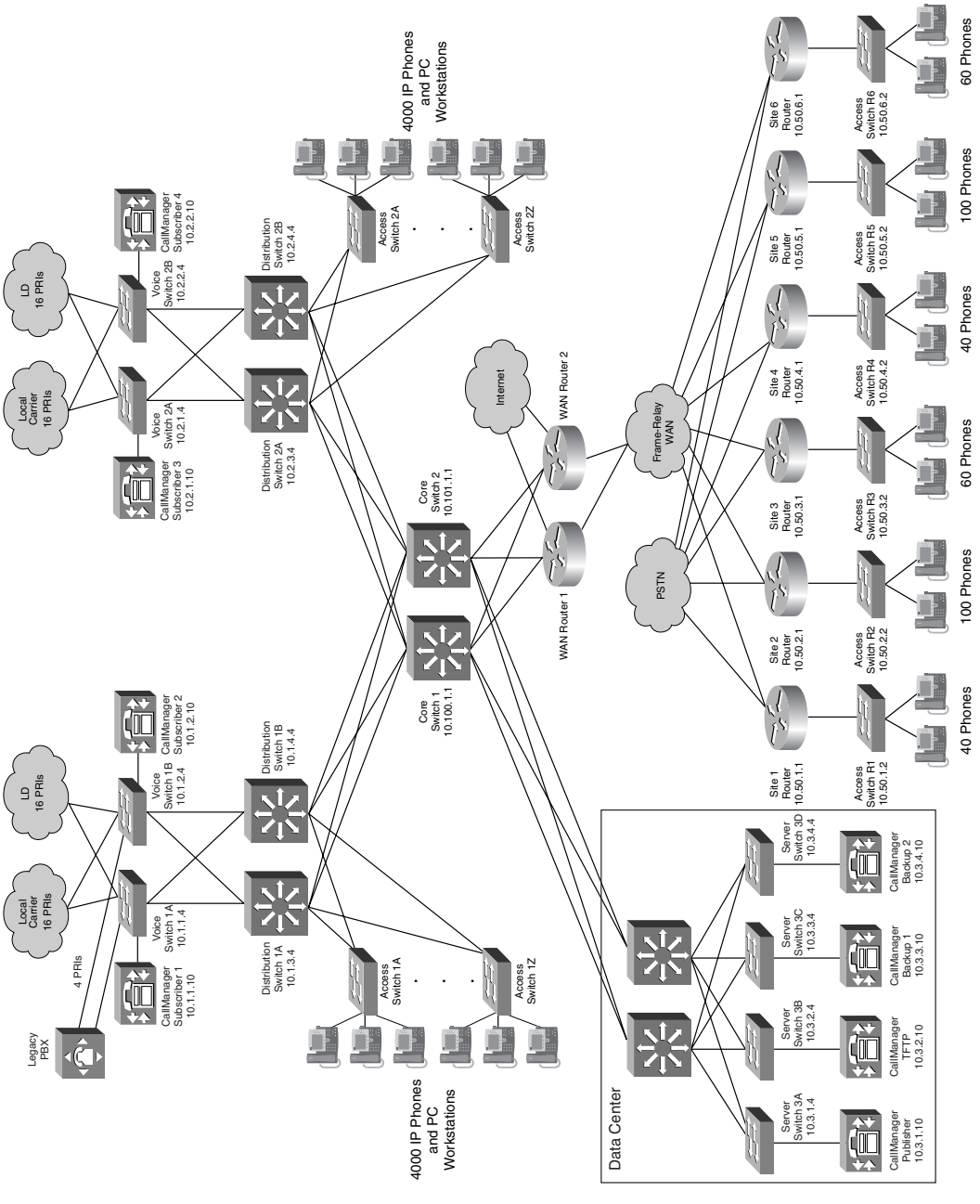
Figure 1-1 shows a typical high-level topology diagram for a large enterprise IP Telephony network. Notice that device names and IP addresses are listed in the diagram. This makes troubleshooting easier by allowing you to quickly look up devices to access them. Because Figure 1-1 is a high-level diagram, it does not get down to the interface level of each device.

Most networks are not as large as the one shown in Figure 1-1. However, no matter the size of your network, a similar topology diagram is very useful for quickly sharing information about your network with others who might be assisting you in troubleshooting.

In addition to the network diagram, you should use some method to store information such as IP address assignments, device names, password information, and so on. For a small network, you can use something as simple as a spreadsheet or even a plain text file. For larger deployments, some kind of database or network management application such as CiscoWorks is recommended. Many customers keep all this topology information on a web server as well, making it quickly and easily accessible to others when it is needed the most. Be sure to keep this information in a secure location.

You also need documentation of your dial plan. Some deployments, especially those heavily utilizing toll-bypass, have very complex dial plans. Knowing where a call is supposed to go just by knowing the phone number and from where it is dialed helps you quickly understand a problem.

Figure 1-1 *Sample High-Level Topology Diagram*



When your topology information is complete, it should include all the following information:

- Interconnection information for all devices, including device names and port numbers. If any patch panels exist between devices, the port numbers should be listed.
- IP addressing for all network devices (routers, switches, and so on)
- IP addressing for all telephony and application servers and voice gateways (including data application servers)
- IP addressing for endpoints (that is, scopes of a DHCP pool)
- WAN and PSTN service provider names and Circuit IDs for each circuit
- Spanning-tree topology, including root bridges for all VLANs and which ports should be forwarding and blocking
- Dial plan information
- Software version information for all devices

If you are troubleshooting a network you didn't design, topology is one of the first pieces of information you should obtain, if it's available. If a topology drawing is not available, it is a good idea to spend time obtaining this information from someone who is familiar with the network and then making a quick sketch. A general topological understanding of the network or at least the piece of the network in question helps when you're trying to differentiate the problem from its symptoms. It's necessary when you're trying to isolate the problem to a particular part of the network.

For example, if a user reports hearing choppy audio when making a conference call, it is essential to know exactly where in the network the conference bridge device is located in relation to the user's phone, including all the intermediate network devices. Without a network diagram, finding this information could waste precious time. Assume that the network you are troubleshooting looks like Figure 1-1. If the user's phone is connected to Access Switch 1A, the other conference participants are on Access Switch 1Z, and the conference bridge device is on Voice Switch 1A, you can see that the number of devices is greatly reduced from 100 or more switches and routers to four or five.

What is worse than not having topology information? Having *incorrect* topology information can lead to countless hours heading down the wrong path. If you're going to keep topology information (highly recommended), make sure you keep it current.

Use all the topology information you have to narrow down which pieces of the network might be involved in the problem you are trying to troubleshoot. To further isolate the problem, interview the end users who reported the problem to gather additional information.

Gathering Information from the User

Information the user provides can be vital to your ability to correct a problem. Try to gather as much detail as possible on exactly what the problem is. Often when troubleshooting a problem, you might realize that what you've been troubleshooting for hours is not really the problem the user encountered. The more detail about the problem you can gather before you begin troubleshooting, the easier it is to find a resolution—and that means less frustration for you. Here is some general information to collect from users:

- Details about exactly what the user experienced when the problem occurred.
- Phone numbers for all parties involved in the problematic call or calls. You can use this as search criteria if you need to look through traces.
- Actions performed by the user when the problem occurred. This includes what buttons were pressed and in what order.
- User observations. This includes text messages displayed on the phone or recorded announcements.
- Information about the user's device. For example, if the user experienced a problem while using a 7960 phone, get the phone's MAC address and IP address, along with registration information and any other statistics available from the phone.

Sometimes the information provided by an end user is not enough to even begin troubleshooting. For example, if a user has trouble transferring calls, you should ask what steps the user took when the problem happened and, if possible, when the problem occurred so that you can examine traces. Sometimes the proper diagnostic tools are not enabled when the problem occurs, forcing you to ask the user to inform you the next time the problem occurs. Be sure to turn on tracing or debugs before making the request so that when the problem occurs again, you will have captured the data. Users can get quite irritated if you have to ask them for the same piece of information two or three times. Also point out to the user the importance of letting you know immediately after a problem occurs, as many of the diagnostic trace files overwrite themselves within several hours or days (depending on the amount of traffic on your system).

Determining the Problem's Timeframe

In addition to *what* the problem is, you should try to determine *when* the problem occurred. Determining the problem's earliest occurrence can help correlate the problem with other changes that might have been made to the system or other events that occurred around the same time. For example, assume that a regular workday begins at 9 a.m. and ends around 6 p.m. Many users report that they get a busy signal when dialing into their voice mail. It is important to know whether they are attempting to do this at 9:10 a.m., a time when the voice mail system is likely under attack from many users all trying to access the system at once. This might change the problem from a troubleshooting issue to a load-balancing or equipment-expansion issue. You check the voice mail system and notice that at the time the

problem was reported, all the voice mail ports were in use. Clearly in this example you need more voice mail ports or servers to handle call volume. However, if the problem occurs at 10:30 p.m., capacity is likely not the problem, so it's time to start troubleshooting your network and voice mail system. As another example, if a user reports that her phone was not working for 10 minutes and you know there was a network outage in her part of the building at that time, you can be relatively sure that the problem was due to the network outage.

When relying on end users to give "when" information about a problem, ask them to note the time on their phone when the problem occurred. The phone's time is synchronized with the clock on the CallManager to which the phone is registered. As long as you have the time on your CallManagers and network devices synchronized, having a phone-based time from the user makes finding the proper trace files very easy.

In some cases, the information about when a problem occurred might be the only piece of information you have other than a limited description of the problem at hand. If you have information about when, you might be able to look through trace files during that timeframe to search for anything abnormal.

TIP Although it is important to use information about *when* the problem started happening, it is equally important to not assume that the problem was a direct result of an event. For example, if a user reports a problem the day after an upgrade was performed on CallManager, you might give some credence to the notion that the upgrade might have caused the problem, but don't automatically assume that this is the root cause.

Step 2: Analyzing the Data Collected About the Problem

Now that you have collected data from a variety of sources, you must analyze it to find the root cause and/or workaround for your problem.

Using Deductive Reasoning to Narrow the List of Possible Causes

The next part of your fact-finding mission is to identify the various components that might be involved and to eliminate as many components as possible. The more you can isolate the problem, the easier it is to find the root cause. For example, if a user complains about choppy voice quality, consider some of the following questions to help isolate the real problem, and think about how the answer will help narrow your focus:

- Does the problem happen on only one phone? If so, you can probably eliminate hundreds or thousands of other phones as suspects. However, keep in mind a single user's perspective. He might think the problem happens only on his phone, so you'll have to ask other users to see if the problem is more widespread than a single phone.

- What numbers are being called when the problem occurs? The answer to this question helps determine which parts of the system are being used when the problem occurs. For example, if the user never experiences poor audio quality when calling certain numbers but always experiences it when calling other numbers, this is a big clue.
- Does the problem happen only between IP phones, only through one or more voice gateways, or both? The user probably won't know the answer, but you'll be able to answer this question yourself after you answer the preceding question about which numbers are being called when the problem happens.

You will find more detailed questions similar to these throughout this book when troubleshooting particular problems.

Although not all of the following apply to every problem, where applicable, you must check all of the following pieces involved in the call. Use your topology information to help obtain this information.

- CallManager nodes involved in the signaling
- Network devices that signaling and/or voice traffic traverse
- Gateways or phones involved in the call
- Other devices involved, such as conference bridges or transcoders

Concentrate your energy on the smallest subset of devices possible. For example, if all the users on a particular floor are having the same problem, concentrate on the problem a particular user is having. If you fix the problem for that one user, in most cases you fix it for all the affected users.

Verifying IP Network Integrity

One thing that people often forget is that your IP Telephony network is only as good as your IP network. A degraded network or a network outage can cause a wide range of problems, ranging from slight voice quality problems to a total inability to make or receive calls on one or more phones. The network is always a consideration when you encounter certain problems, so network health issues are covered throughout this book. Network health is especially important during the discussion of voice quality problems in Chapter 7, "Voice Quality," because most voice quality problems stem from packet delay and/or loss.

Always remember to keep the IP network in mind and look at every layer in the OSI model, starting from Layer 1. Check your physical layer connectivity (cables, patch panels, fiber connectors, and so on). Then make sure you have Layer 2 connectivity by checking for errors on ports, ensuring that Layer 2 switches are functioning properly, and so forth. Continue working your way up the stack until you reach the application layer (Layer 7). As an example, two of the most common reasons for one-way audio (where one side of the conversation cannot hear the other) are the lack of an IP route from one phone to another and the lack of a default gateway being configured on a phone. Taking the layered approach,

you would first check the cabling and switches to make sure that there are no errors on the ports. You would then check Layer 3, the network layer, by ensuring that IP routing is working correctly. When you reach this layer, you discover that for some reason the IP packets from one phone are unable to reach the other phone. Upon further investigation, you might discover that there was a missing IP route on one of the routers in the network or a missing default gateway on one of the end devices (such as an IP phone or voice gateway).

Determining the Proper Troubleshooting Tool

After you narrow down the appropriate component(s) causing a problem and have detailed information from the user(s) experiencing the problem, you must select the proper tool(s) to troubleshoot the problem. Most components have multiple troubleshooting tools available to help you. Chapter 3, "Understanding the Troubleshooting Tools," provides more details about some of the tools available for troubleshooting CallManager. You should use the tracing and debugging facilities available in CallManager and other devices to determine exactly what is happening. Additional tools and traces are covered in the chapter associated with diagnosing certain types of problems. For example, Chapter 6, "Voice Gateways," covers debugging Cisco IOS Software voice gateways. Because CallManager is central to almost all problems, information about various portions of the CCM trace facilities appears throughout this book.

This step is the most demanding on your troubleshooting skills because you analyze the detailed information provided in the various tools and use it to search for additional clues using other tools. Sometimes the problem description you have is not detailed enough to determine which tool to use. In this case, you should try various tools in search of anything that looks out of the ordinary.

The following case study shows how this troubleshooting methodology works in a real-world scenario.

Case Study: Resolving a Problem Using Proper Troubleshooting Methodology

It is 6 a.m., and you have arrived at work to resolve your CEO's problem. The only data you have is the page you received at 5:30 a.m. that says "CEO's calls keep dropping. Please help ASAP!" You need a bit more information than that to fix the problem.

This case study applies the methodology previously described. You must gather the data before you can begin the analysis.

Gathering the Data

As part of the data-gathering stage, you should do the following:

- Identify and isolate the problem
- Use topology information to isolate the problem
- Gather data from the end users
- Determine the problem's timeframe

You find the CEO's administrative assistant and begin your fact-finding mission. He states that at various times during the previous day and one time this morning, the CEO is on the phone when, all of the sudden, the call is disconnected. Eager to resolve the problem, you ask the administrative assistant for the following information:

- The exact date and times the problem occurred
- Whether the dropped calls were incoming or outgoing
- What number was dialed if it was an outbound call or what number the call came from if it was an inbound call

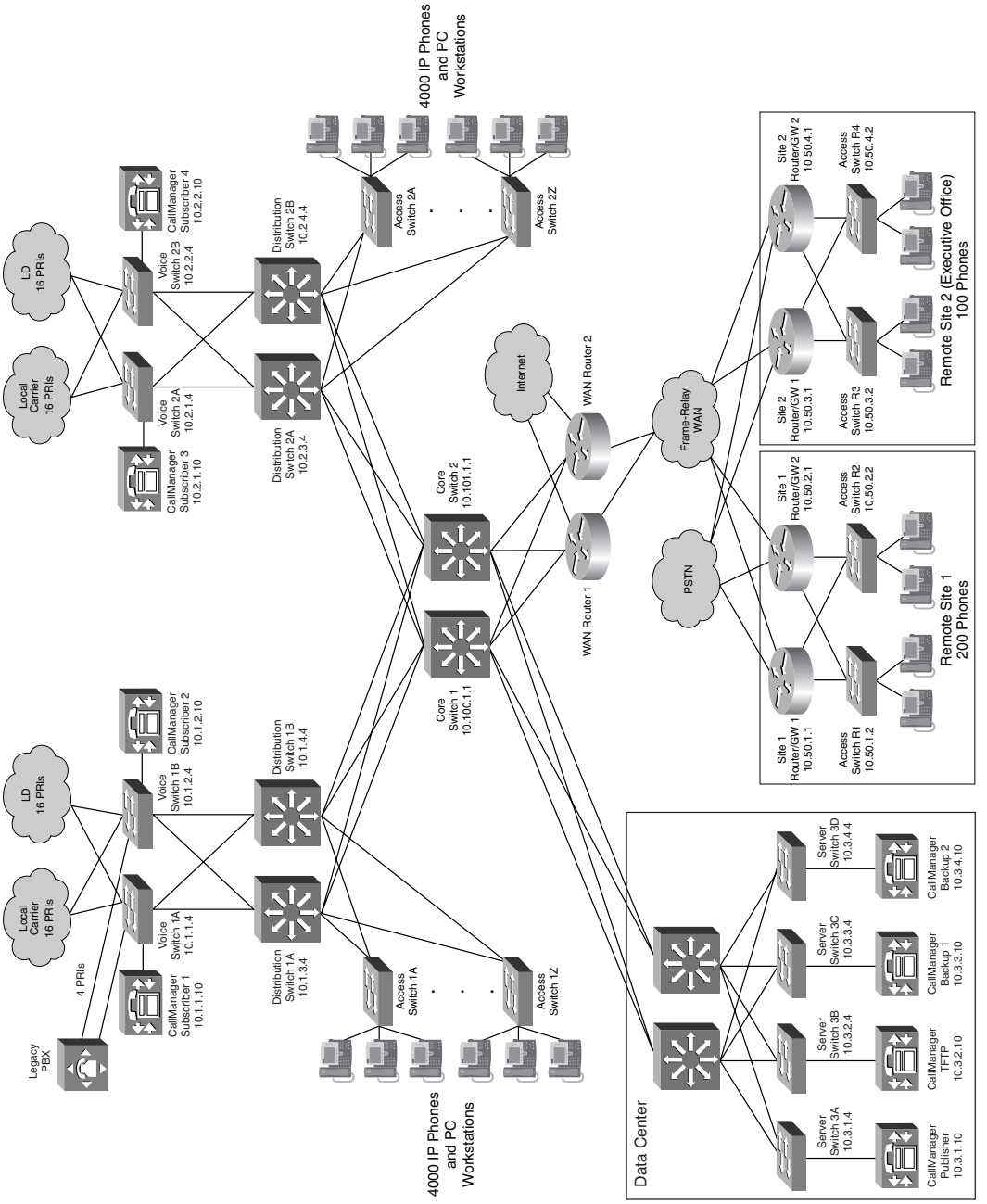
The assistant states that the call was dropped around 5:15 a.m. because the CEO was in early to prepare for the stockholders meeting. This is the extent of the information he remembers. Most users do not pay attention to specifics like this unless they have been instructed to, but all is not lost. The CEO has a 7960 phone that stores information locally about missed calls, received calls, and placed calls. You head into the CEO's office and look at the list of received calls and placed calls for the morning. You notice that a call was received at 5:05 a.m. and a call placed at 5:25 a.m. You notice that the second call was placed to the same area code and prefix as the call that was received.

You ask the CEO about the two calls. She remembers that she was on the phone with a customer for about 15 minutes when the call was disconnected. She immediately called the customer back. She also confirms that the first call that was received was the dropped call. Now you know that the problematic call was received at approximately 5:05 a.m. and was dropped just before 5:25 a.m.

While you are looking at the CEO's phone, you also go into the Settings menu (press the **settings** button **> Network Configuration > CallManager 1**) to see which CallManager the CEO's phone is registered to. This lets you isolate which CallManager in the cluster is involved in the signaling for this phone.

Armed with this information, you can begin the task of isolating the problem. You refer to your topology diagram to isolate the components that are involved. Figure 1-2 shows a high-level diagram of the network topology.

Figure 1-2 *High-Level Topology Diagram*



Reinforcing the topology in Figure 1-2, assume the following setup:

- A cluster with eight CallManager nodes
- 32 voice gateway connections to the PSTN for outgoing calls at your main site—16 for local calls and 16 for international and long distance
- 32 more voice gateways at your main campus where all your inbound calls come in. The telephone company has set up the inbound calls so that the 32 gateways are redundant whereby if one of the gateways is down, all your incoming calls can still use any of the other remaining gateways.
- Two gateways at each remote site used for both inbound and outbound calls. All outbound calls prefer the first gateway, and inbound calls prefer the second gateway, although each can handle both inbound and outbound calls should one fail.

As shown in Figure 1-2, the executive offices are at a remote site across the WAN. With just the information you have so far, you can eliminate a large portion of the network. So far you know that the problematic call was to the CEO. You also know that the problematic call was an inbound call. You ask the CEO and her admin if all the dropped calls were inbound calls. As far as they can remember, they were.

You know that the call this morning was during a time of day where there is little phone activity. Remember that all inbound calls to the remote site come in through Primary Rate Interfaces (PRIs) connected to the remote voice gateways and that inbound calls to the site prefer the second gateway. It is unlikely that all the channels on the first PRI were in use during a time of low call volume, so you assume that the call probably came in through the second gateway, although you still keep it in the back of your mind that the call might have come in through the first gateway at the remote site.

You then look at the configuration for the two gateways at Remote Site 2 and note that they are both configured to send incoming calls to CallManager Subscriber 3 as their preferred CallManager and CallManager Backup 1 in case CallManager Subscriber 3 fails.

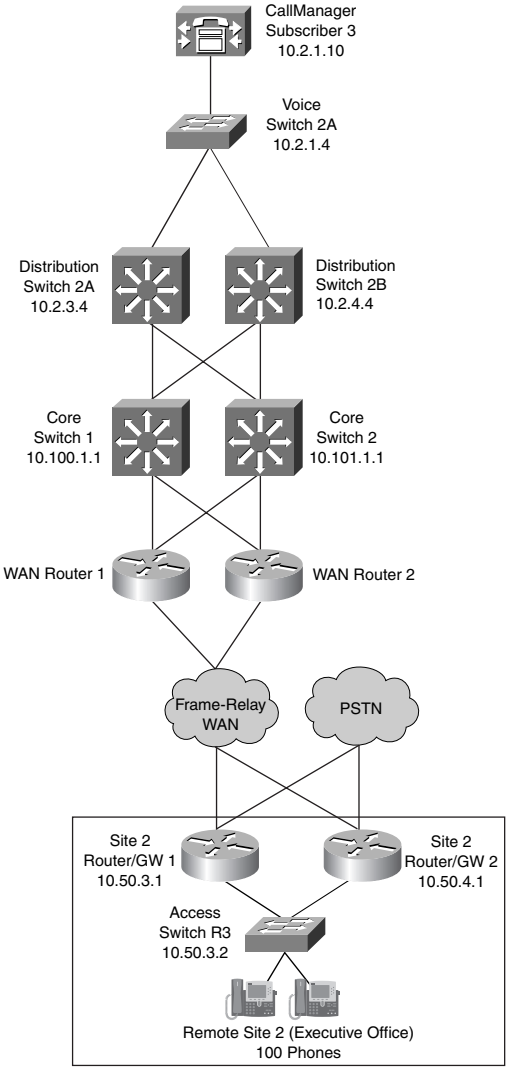
With the information you have so far, you can narrow down the possible suspect devices to the network shown in Figure 1-3.

Armed with this knowledge, you can immediately isolate the problem to the user's phone and the two gateways being used for inbound calls. Keep in mind that you haven't eliminated the possibility that the problem is on CallManager or is network-related.

Now that you know the problem is related to inbound calls, it makes sense to try to understand the call flow for an inbound call to this user. Determine whether these calls all come directly to the user or if the call flow has any intermediate steps, such as Cisco IP Auto Attendant (Cisco IP AA) or an operator who transfers the call to the end user. For the sake of this example, assume that the user has a Direct Inward Dialing (DID) number, so the call comes straight from the PSTN through a gateway to the user, and a Cisco IP AA or operator is not involved. You have now eliminated Cisco IP AA from the picture, as well as the

possibility that other phones or users are involved in this user's problems. This is not to say that other users are not experiencing similar problems, but the focus here is on solving this particular user's problem. If the problem is more widespread than this one user, you will probably find it as you continue to troubleshoot this user's problem.

Figure 1-3 *Network After You Narrow Down the Possible Suspects*



At this point, the problem has been isolated to the following culprits:

- The CEO's phone
- CallManager Subscriber 3
- Site 2 Router/GW 1 and Site 2 Router/GW 2
- The underlying network connecting these devices

It might seem like you haven't made much progress in this example, but in reality you have eliminated a large portion of the system as possible culprits. This concludes the data-gathering piece of your investigation. Now it is time to start analyzing the data. After you isolate the problem, you must break it into smaller pieces.

Analyzing the Data

As soon as you have a clear understanding of the problem you're trying to resolve, and you have isolated the piece or pieces of the network that are involved, the next step is to break the problem into pieces to find the root cause. As part of the data analysis stage, you should do the following:

- Use deductive reasoning to narrow the list of possible causes
- Verify IP network integrity
- Determine the proper troubleshooting tools, and use them to find the root cause

Continuing with the case study example, you now know the pieces involved in the puzzle, but you still don't know why the call is being dropped. For the sake of this example, this chapter keeps things general, but later chapters go into far greater detail on exactly what to look for. In this case, the problem is likely caused by the phone, CallManager, the gateway, the PSTN, or the IP network. So how do you determine which one is causing the problem?

One important distinction to make that will become evident as you read through this book is that many problems can be narrowed down to being either signaling-related or voice packet-related. In this case, you are dealing with a signaling-related problem, because the problematic call is being torn down—a problem that must occur in the signaling path between devices.

Because nearly all signaling for a call must go through one or more CallManager servers, the first tool you decide to use is a trace from CallManager Subscriber 3. You can then analyze the trace files to discover the device that disconnects the call from CallManager's perspective—in other words, "Who hung up first?" Using the information provided by the user, you must find the proper trace file and try to reconstruct the call from beginning to end.

A call between the CEO's phone and the voice gateway has two distinct signaling connections. One is the communication between CallManager and the voice gateway. The other is the communication between CallManager and the phone. The phone and voice gateway never directly exchange signaling data. All signaling goes through CallManager.

The trace includes all the messaging between CallManager and both the phone and the gateway. Chapter 3 provides more details on where to find these traces and how to read them.

You know that the call in question was set up around 5:05 a.m., so you look through the traces during that timeframe, searching for the phone number you retrieved from the CEO's phone. After combing through the trace file, you determine that the gateway is sending a message to CallManager, telling it to disconnect the call. The CCM traces (discussed in Chapter 3) indicate which gateway the calls are coming from. This eliminates the CEO's phone as a cause of the problem because the disconnect message is coming from the gateway. Because the user indicated that there were three drops, you can now go through the same process of looking through the CCM trace files for each instance of a dropped call and reconstructing those calls to see if the problem is isolated to one gateway. If you don't know the times that the other calls were dropped, you should just concentrate on the one call you do have data for.

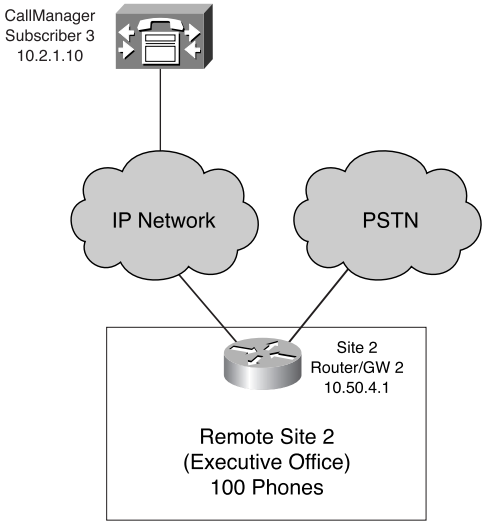
Because CallManager received a message from the gateway telling it to disconnect the call, it is unlikely that a network problem is causing the calls to disconnect. If there were a network problem, you would likely see an indication that there was a problem communicating between CallManager and the gateway. In this case, the gateway had no problem sending the disconnect message to CallManager. It would not hurt to look through the network devices between CallManager and the voice gateway to ensure that there are no network errors, but with a problem like this, the network is an unlikely culprit.

At this point, you have narrowed down the problem to be originating from either the voice gateway or the PSTN. Figure 1-4 shows you've narrowed down the network to only a few devices.

The next step is to go to the suspected gateway and try to determine why one of the calls was dropped. This involves turning on additional debugs on the gateway to determine if the gateway is disconnecting the call or just passing along information from the PSTN about disconnecting the call. Unfortunately, it is unlikely that you had the debugs enabled at the time the problem occurred, so you need to enable the proper debugs and wait for the problem to happen again. This is why it is so important to narrow down the problem to a small subset of devices: You do not want to turn on debugs on dozens of gateways.

Which debugs to use depends on the gateway model and the type of interface to the PSTN. Chapter 6 discusses these considerations in detail. While waiting for the problem to reoccur, you discover that a message to disconnect the call is coming from the PSTN. If you are using an ISDN voice circuit for connectivity to the PSTN, the disconnect message is accompanied by a cause code that provides a general reason why the call was disconnected. Depending on what you discover on the gateway debugs, the next step might be to contact the local service provider or perhaps debug the gateway further to find the root cause.

Figure 1-4 *Network After You Continue Narrowing Down the Possible Suspects*



Conclusions

As this case study has demonstrated, the more information you can obtain about the problem, the easier it is to get to the root cause. For example, without the times the dropped calls occurred, it would have been almost impossible to find them in the trace files on a busy system. When deployed in a large enterprise, it is good to arm your help desk with a list of questions to ask depending on the problem being reported.

The point of this example is not to teach you how to troubleshoot a specific problem or to find out exactly why the user's calls are being dropped. It is to show you how to approach a problem in order to isolate it and break it into more manageable pieces. The same principles can be applied to almost any problem you are troubleshooting.

So remember, first put on your detective hat and gather enough information to isolate the problem to a few pieces of the system. Then dig deeper into each component by breaking the problem into more manageable pieces. Finally, apply your expertise to each of the smaller pieces until you find the resolution to your problem.

Summary

This chapter discussed the methodology you should employ to successfully troubleshoot problems in an IP Telephony network. You should become familiar with the methodologies discussed here. It is vital that you always follow a consistent approach to troubleshooting. Many basic problems can be avoided by using a consistent troubleshooting approach.

Also, be sure that you understand the big picture of IP Telephony architecture. What areas are you unsure about? Are you strong in IP but weak in call processing skills? Are you familiar with the basic protocols that are used? Consider where you are now, and as you move forward, pay particular attention to strengthening your weak areas.

As you begin this journey, hopefully this book can bring some illumination to the sometimes daunting task of troubleshooting an IP Telephony network.