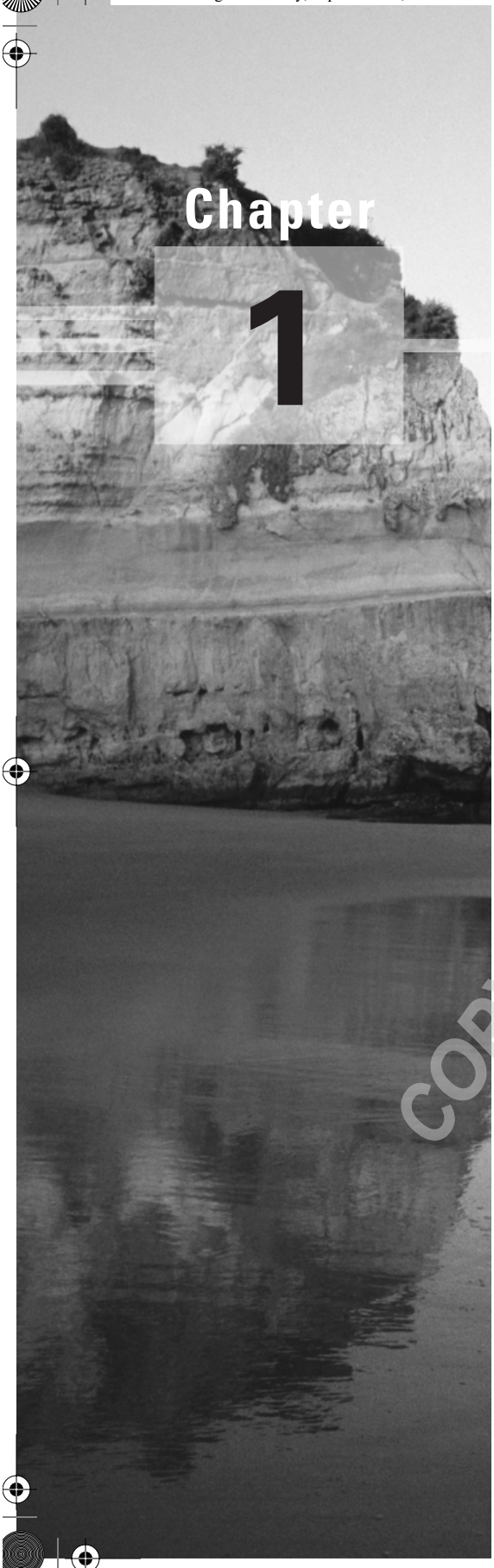


Chapter 1

Troubleshooting Methodology

EXAM TOPICS COVERED IN THIS CHAPTER INCLUDE:

- ✓ **Know troubleshooting methodologies.**

Troubleshooting is a skill that takes time and experience to fully develop. To be successful when diagnosing and repairing network failures, a good set of troubleshooting tools and skills is essential.

While there's no specific exam objective that maps to this chapter, the information presented here is nevertheless important to the exam. This chapter emphasizes the importance of following a specific set of troubleshooting steps when you try to diagnose and solve network problems. An effective troubleshooting methodology is needed because of the complexity of today's network environments. As a Cisco Certified Network Professional, you need to understand and know how to apply an efficient and systematic troubleshooting methodology. Otherwise, you would be required to have a very intimate understanding of the network you are troubleshooting. It is imperative that you learn troubleshooting skills and understand the information available to you while solving network problems.

The Complexity of Internetworks

When a network failure occurs, time is of the essence. When a production network goes down, several things are affected. The most important of these is the bottom line—network failures cost money. A good example is a call-center network. The company relies on the network to be available for its employees so that they can take phone orders, answer inquiries, or perform other business transactions that generate income. A failure in this environment needs to be diagnosed and repaired in a timely manner. The longer the network is down, the more money the company loses.

To minimize monetary and productivity losses, network failures must be resolved quickly. Troubleshooting is an integral part of getting this done. Intimate knowledge of a network also facilitates rapid resolution. Armed with a few troubleshooting skills and intimate knowledge of the network, you can solve most problems rather quickly, thus saving money.

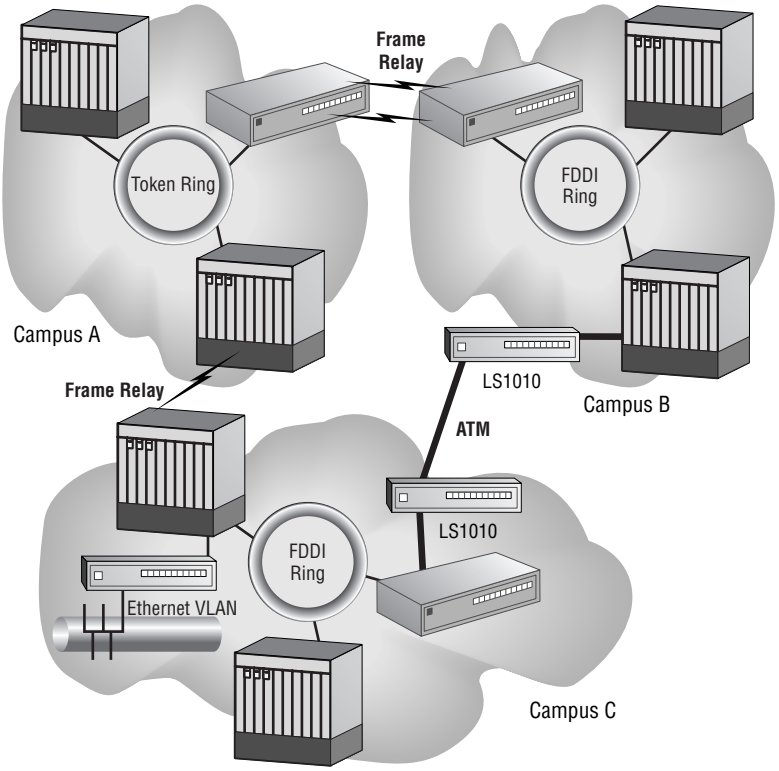
Hold on a minute. What if you're new on the job and you don't yet have an intimate knowledge of the network? You can probably get up to speed quickly enough, right? Although that may have been the case in the past, getting up to speed becomes an overwhelming challenge in today's complex networks. These networks consist of many facets of routing, dial-up, switching, video, WAN (ISDN, Frame Relay, ATM, and others), LAN, and VLAN technologies. Refer to Figure 1.1 to get an idea of how these technologies intertwine. Notice that ATM, Frame Relay, Token Ring, Ethernet, and FDDI all are present. Each technology has its own properties and commands to allow for troubleshooting. Various protocols are used for each of these technologies. In addition, different applications require specific network resources. (At least the

seven-layer OSI model, which you will review in Chapter 4, is used to maintain a common template when designing new technologies and protocols.) It would take you a long time to master all of the technologies implemented in the network and to be able to solve network problems, based on your knowledge of the network alone. All of these factors contribute to today's complex network environments.

There must be an easier, more logical way to efficiently and successfully troubleshoot without having to become intimately familiar with every network environment. Well, you'll be happy to know that there is an easier option—following a troubleshooting model—and it is discussed in detail in this chapter. By following a troubleshooting model, the need for intimate knowledge of the network is reduced. A troubleshooting model should be adopted to help resolve network malfunctions and reduce downtime.

Let's move on to discuss Cisco's model in detail.

FIGURE 1.1 Today's complex enterprise network



The Problem-Solving Model

Imagine trying to solve a network failure by using a different approach every time. With today's complex networks, the possible scenarios would be innumerable. Because so many different things can go wrong within a network, it would be possible to start from many different points. Not only is this an ineffective method of troubleshooting, but it is also time-consuming, and time is very valuable in a "network down" situation.

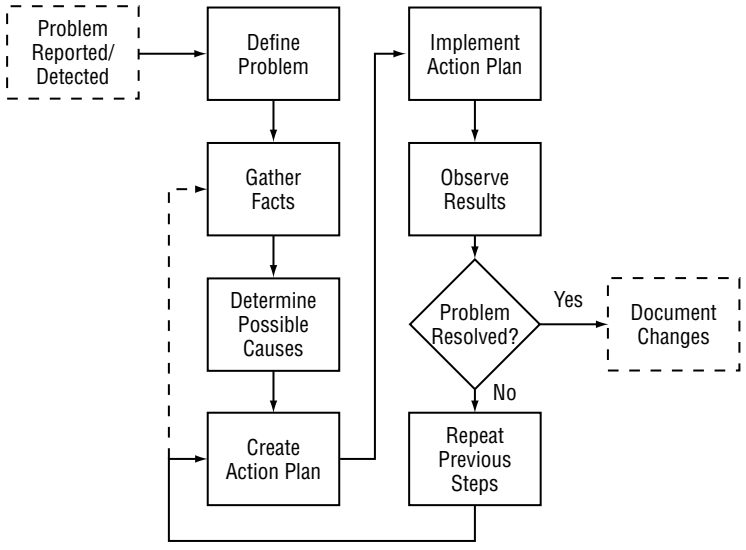
Cisco has designed an effective *troubleshooting model* that contains seven steps. A troubleshooting model is a list of troubleshooting steps or processes that can be followed to provide an efficient manner of resolving network problems. The headings in this section contain information specific to each step of the troubleshooting model. (Steps 4 and 5 are combined into one section of the chapter—creating and implementing the action plan.) After the seven steps are completed and the problem is resolved, a few more actions follow, such as completing documentation of the problem-solving events.

To be effective when troubleshooting and to achieve faster resolution times, follow the model outlined in Figure 1.2. This flow chart shows the seven steps.

The process begins when a network failure is reported to you. Following are brief descriptions of the steps to take:

1. Define the problem. At this point in the process it is important to make a determination of the issue, identifying sets of symptoms and potential causes.

FIGURE 1.2 Cisco's troubleshooting model



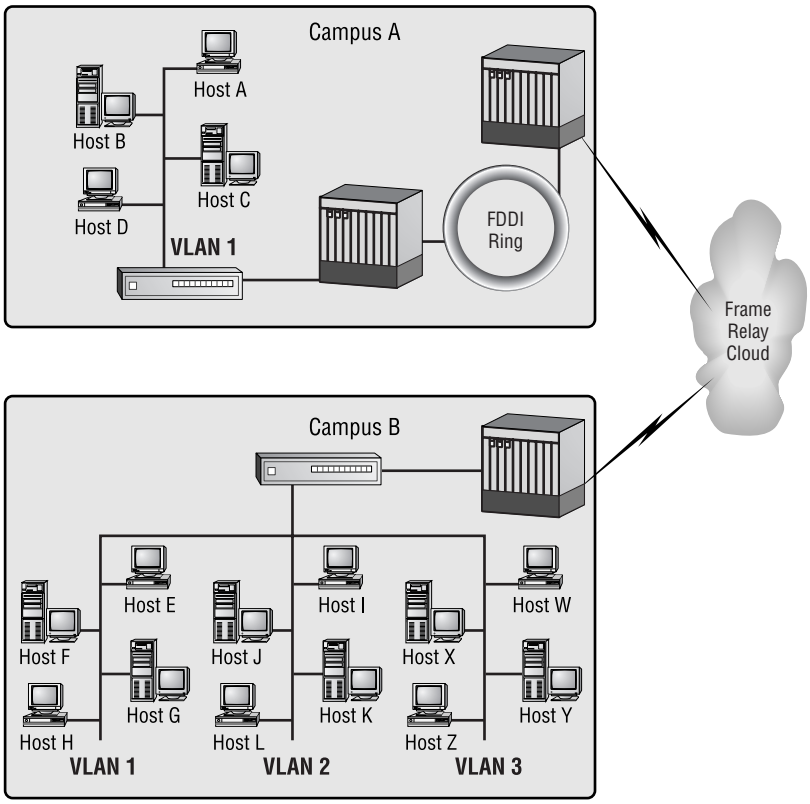
2. Gather detailed information. These facts about the problem can be obtained from a number of sources, including key users, network management systems, output from router and switch diagnostic commands, and protocol analyzer traces.

3. Consider possible scenarios. Brainstorm and come up with several possible or probable causes of the failure. Also, when developing this list, eliminate items that are definitely not the cause of the problem.

4. Create an action plan. Begin with the most likely source of the trouble and devise a plan to correct this issue, changing only one variable at a time. If you change multiple items simultaneously, it is possible that the problem will be resolved without your identifying the root cause. This then leaves the potential for the problem to repeat itself in the future.

5. Implement the action plan. As you implement each step of the action plan, carefully check to see if the problem has been resolved.

FIGURE 1.3 Example campus network



6. Observe the results of implementing the action plan. In many instances it will be clear when the problem is resolved; however, in those cases where the problem is subtler, a more structured observation technique must be used. This technique involves many of the same tools used in the fact-gathering portion of the process, such as talking to users, employing network management tools, and checking router and switch output.

7. Repeat the process if the action plan doesn't fix the problem. Revise your action plan to address the next most likely source of the trouble. Be sure to undo the changes that were attempted in the previous attempt. Then repeat the process starting with step 4. If there are no more potential causes for which to create an action plan, start with step 2 and repeat the process.

The best way to understand how Cisco's model works and how you should use it is by looking at an example. For this example, assume you are in charge of operational support of the network pictured in Figure 1.3. There are two campus networks, connected via a Frame Relay cloud. Within each network, VLANs are connected to a Catalyst 6500 switch and then to a core router that has a connection to the Frame Relay cloud in one way or another.

The fun begins when you get a call from a user who "can't get to Host Z." Based on this information, let's apply Cisco's troubleshooting model to solve the user's difficulty and fix the problem in the network.

Step 1: Define the Problem

As you can see, the user's problem is vague; you need more information if you are to solve the problem any time soon. This is where *problem definition* comes in. Problem definition is the step in the troubleshooting model when details are used to define what the most likely cause of a problem is. Now, while you still have the user on the line, the first step is to ask him what he means when he says he can't "get to" Host Z. The user then defines the situation by telling you that he can't FTP to Host Z. Ask the user if he experiences any other difficulties or if this is the only one. Verify where the user is currently located. After these preliminary questions, you'll have a basic idea of what is and isn't working. Unfortunately, you can't simply assume that the FTP is broken, because there are many other pieces of the network that can contribute to this problem.

It is also important to realize that you may want or need to gather facts before you actually form your problem statement. By gathering facts to help define the issue, the diagnosis of the problem or problems will be more accurate and will help you solve the trouble more quickly in the end. Problem definition and fact gathering should be used in tandem for a quick and accurate resolution.

Once you have enough information to define the problem, you should create a problem statement that is specific, concise, and an accurate description of what needs to be solved. In this case, you might have a statement that says *User A from Campus A cannot FTP to Host Z on Campus B*. With a good statement of the problem, it is easier to focus on the problem itself and not try to troubleshoot issues that do not fall within the problem definition.

Step back for a moment before you actually form your final problem statement. You need to gather more information before you can form an accurate problem statement. It's time to move on to the fact-gathering step. Keep in mind, however, that after you accumulate all the information, you have to come back and create your problem statement.

Step 2: Gather Facts

At this point, the problem is still pretty vague and needs more definition. This is where the fact-gathering step of the troubleshooting model is employed. *Fact gathering* is the process of using diagnostic tools to collect information specific to the network and network devices that are involved in a problem. Additional information should include data that excludes other possibilities and helps pinpoint the actual problem. An example of fact gathering in the case we're discussing is to verify whether you can ping, Traceroute, or Telnet to Host Z, thus reducing the number of possible causes.

Depending on the user and situation, you may or may not be able to get more detailed information. It is up to you as a network engineer or administrator to solve the problem, which means that you may have to get the information yourself.

It is important that you gain as much information as possible to actually define the problem while in the problem-definition phase of the troubleshooting model. Without a proper and specific definition of the problem, it will be much harder to isolate and resolve. Information that is useful for defining a problem is listed in Table 1.1.

TABLE 1.1 Useful Information for Defining a Problem

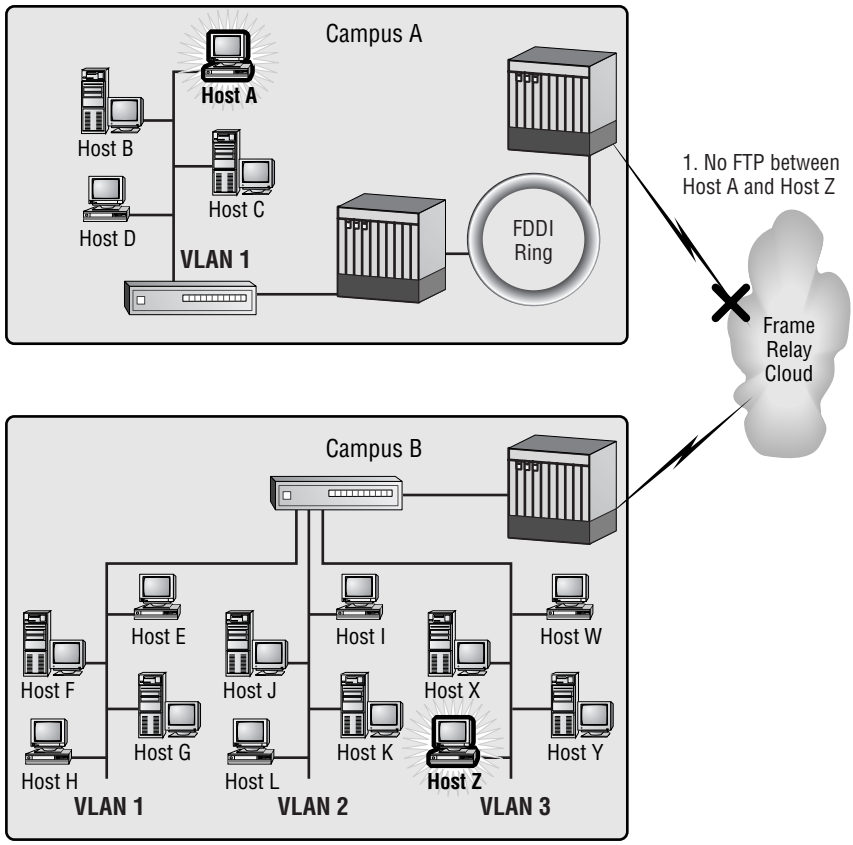
Information	Example
Symptoms	Can't Telnet, FTP, or get to the WWW.
Reproducibility	Is this a one-time occurrence, or does it always happen?
Timeline	When did it start? How long did it last? How often does it occur? Has the current configuration ever worked properly?
Scope	What are you able to access successfully via Telnet or FTP? Which WWW sites can you reach, if any? Who else does this affect?
Baseline Info	Were any recent changes made to the network configurations?

All of this information can be used to guide you to the actual problem and to create the problem statement. Use your network topology diagram and check each item in Table 1.1.

Identify Symptoms

First, you need to define what is working and what isn't. You can do this by identifying the symptom and defining the scope. Figure 1.4 is a picture of your network. Although the large X on the Frame Relay cloud represents that there is an FTP connectivity issue, it does not indicate the location of the failure. Right now, all you know is that a single user could not FTP to Host Z.

FIGURE 1.4 Host A cannot FTP to Host Z.



Reproduce the Problem

Before spending time and effort trying to solve this problem, verify that it is still a problem. Troubleshooting is a waste of time and resources if the problem can't be reproduced. It's just like a dog chasing its tail. If the issue is intermittent, further steps should be taken to capture as much information as possible about the event the next time it does occur. This will help narrow down the scope of items you will look at.

Understand the Timeline

In addition to verifying whether the problem is reproducible, it is important to investigate the frequency of the problem. For instance, maybe it happens only once or twice a day. By establishing a timeframe you can more readily identify any possible causes. In addition, you need to know whether this is the first time the user has attempted this function. There is a different set of variables involved with an item that worked yesterday but not today than there is with something that fails during first-time use. Obviously, if it worked yesterday, you can look at what changed overnight as well as looking for something that is broken. If the user has never used this

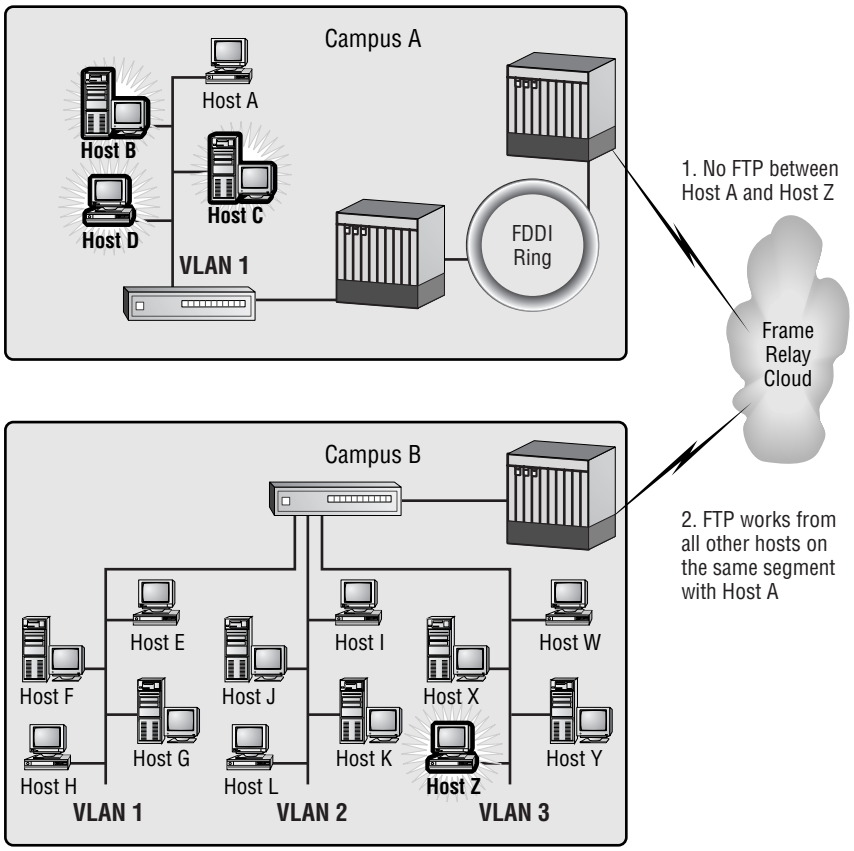
feature before, there may be an existing access list or other security device that has only now been activated by the user's initial use of this application.

Determine the Scope of a Problem

Next, you need to find out whether anyone else is unable to FTP to Host Z. If others can FTP to Host Z (for the sake of this example, assume that they can), you can be pretty sure that the problem is specific to the user, either on their station or on the destination host. This step determines the scope of the problem and helps to differentiate between a user-specific problem and a more widely spread problem. Figure 1.5 shows that other hosts can FTP to Host Z without any problems.

Now that you have the problem narrowed down to a single user, you need to define the *boundary of dysfunctionality*. The boundary of dysfunctionality is the limit or scope of the network problem. For example, a distinction can be made between where nodes are functioning properly and where they are not. To define this boundary in our example, you need to know whether the user can successfully FTP anywhere.

FIGURE 1.5 Other hosts can FTP to Host Z

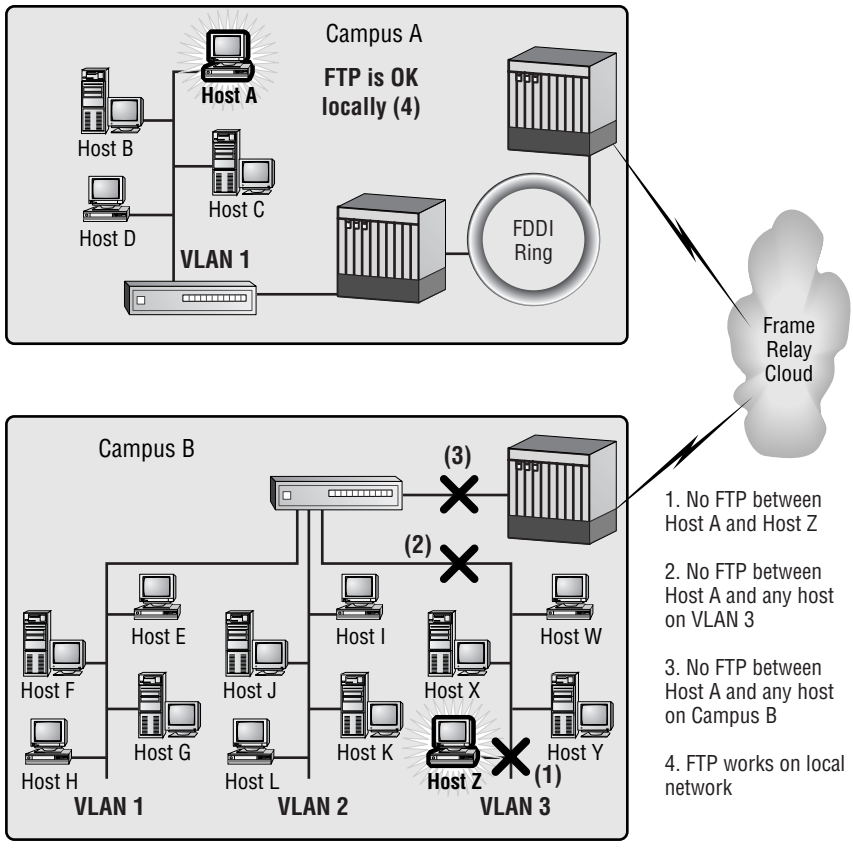


There are three methods for establishing the boundary of dysfunctionality: outside-in troubleshooting, inside-out troubleshooting, and divide-by-half troubleshooting. Each of these techniques has its own advantages and disadvantages based on the situation. The methods are explained in the following sections.

Outside-In Troubleshooting

The first method, *outside-in troubleshooting*, consists of starting the troubleshooting process at the opposite end of the connection. In this case, you would start at Campus B, VLAN 3, and work back toward the user's system (see Figure 1.6). The corresponding test would be for the user to try to FTP to another host on the same VLAN as Host Z, indicated by the X (2) on the diagram. If the result of that test is negative, then you need to come back one step. By coming back one step, you would try to FTP to a host on a different VLAN, indicated by the X (3) on the diagram. If that test failed, the only thing left to try would be to FTP to another host on the user's segment. In the example, assume that the user can FTP to other hosts that are directly connected to the same Ethernet segment. In general, outside-in troubleshooting is a good method to use when there are many hosts that cannot connect to a server or subset of servers.

FIGURE 1.6 Starting from the outside and working in

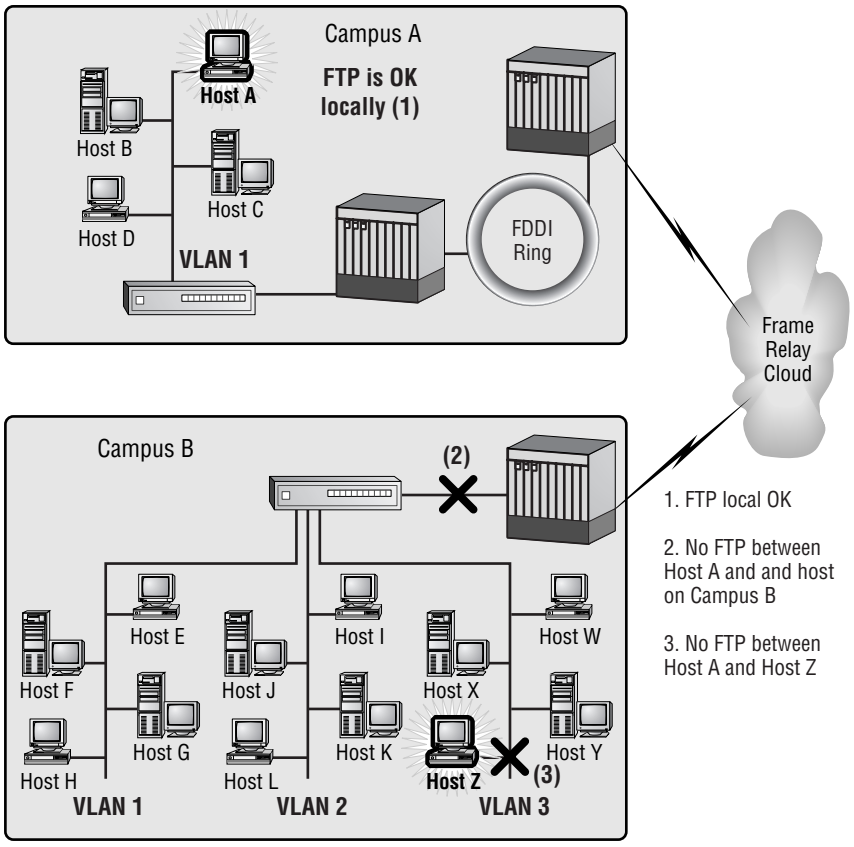


Inside-Out Troubleshooting

The second method of fixing the boundary of dysfuncionality is to start near the user and work your way toward the destination, Host Z in this case. This is referred to as the *inside-out troubleshooting* method. Figure 1.7 contains a diagram that describes this testing method. You see that the user can FTP to hosts within the same network, but can't FTP to any host on the Campus B network. The steps are marked by the Xs, with the step number in parentheses.

Using the second method saved you one step—three instead of four. Statistically, however, you isolate the boundary with fewer steps by using the first method. The important thing is that the boundary be established.

FIGURE 1.7 Starting from the inside and working out

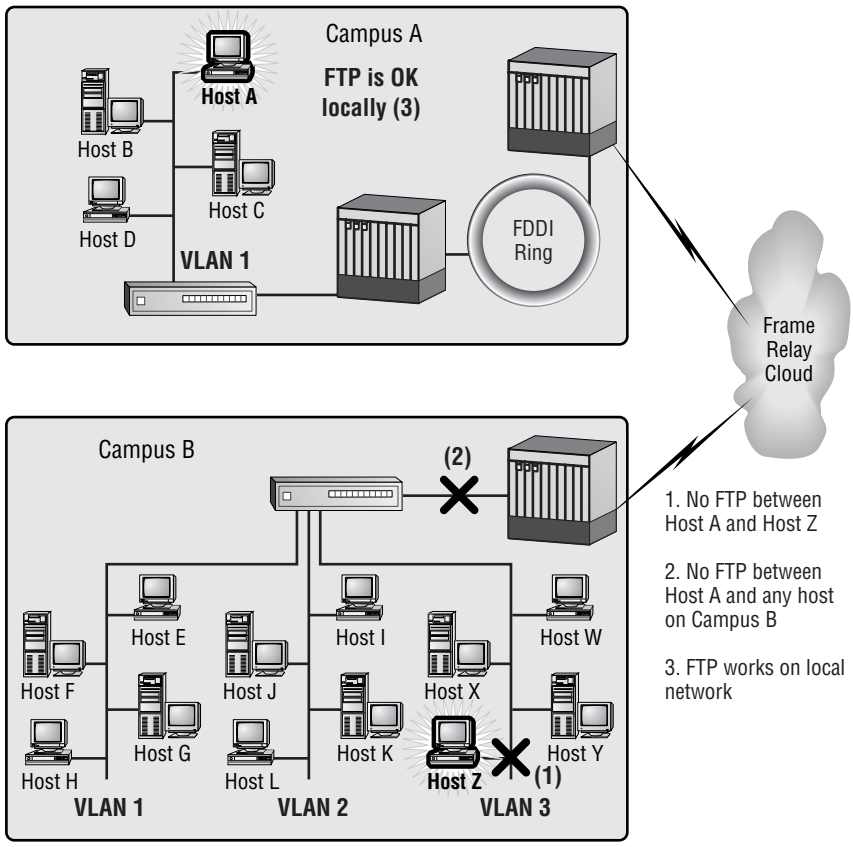


Divide-by-Half Troubleshooting

The third and final method is *divide-by-half troubleshooting,* which is depicted in Figure 1.8. Divide-by-half indicates that a point between two ends of a network problem is used as a troubleshooting reference point. Either half of the problem's scope may be investigated first. In this example, you start by trying to FTP to any host within Campus B. Depending on the results, you can divide in half again and test. If the test results in a successful FTP to any host on the Campus B network, then the new point to test is another host on VLAN 3. If the test fails, the new testing point is to try to FTP to a local host. In this case, the divide-by-half method takes three steps, just as the inside-out method does.

You now have isolated the problem to something outside the immediate network. Upon further inspection and fact gathering, you find that the user can't ping external hosts, either. With all this information in hand, you can now start to contemplate possible causes of the failure and move on to the following Consider Possibilities step.

FIGURE 1.8 Divide-by-half method



Step 3: Consider Possibilities

This step within the troubleshooting model is used to contemplate the possible causes of the failure. Obviously, it is quite easy to create a very long list of possible causes. That is why it is so important to gather as much relevant information as you can and to create an accurate problem statement. By defining the problem and assigning the corresponding boundaries, the resulting list of possible causes diminishes because the entries in the list will be focused on the actual problem and not on "possible" problems.

First, review what you know about your sample problem:

- Host A can't FTP to Host Z.
- Host A can't FTP to any host on Campus B.
- Host A can't ping to anywhere outside its own network.
- Host A can FTP to any host on its own network.
- All other hosts on Host A's network can FTP to Host Z, as well as to other hosts.

Based on what you know, you now need to list possible causes. These possible causes are as follows:

- No default gateway is configured on Host A.
- The wrong subnet mask is configured.
- There is a misconfigured access list on the router connected to the switch on Campus A.

If you had not gathered such specific information in step 2, the list could have included all possible problems with any piece of equipment between Host A and Host Z. That would have been a long list, and it would take a lot of time to eliminate all of the possible causes.

Remember that because these are only *possible* causes, you still have to create an action plan, implement it, and observe to see whether the changes made were effective. When the list of possible problems is long, it may require more iterations of the problem-solving steps to actually solve the problem. In this example, you have only four possible causes, so this is a much more manageable list. Although there may be other possible causes that you can think of (and it's great that you can do that), for this example and in the interest of simplicity, only these three are listed.

Here's where it gets interesting. You now have to check each of these possibilities and fix them if they are the cause of the problem. To do this, move on to the next step, which is to create an action plan.

Steps 4 and 5: Create and Implement the Action Plan

Creating an *action plan* is actually very easy. It entails the documentation of steps that will be taken to remedy the cause of the network problem. Most of the hard work was gathering information about the problem. The investigation gave you three leads about the source of the problem. Now it is a matter of checking out each possibility and determining which one is most likely the source of the issue.

The majority of the possibilities point directly at the host machine, so start there. The first two causes are host configuration issues. Now, assume that you've checked the TCP/IP configuration on the host and everything is configured properly. You can eliminate the host machine as the culprit.

You then move on to the remaining possible cause, which is an access list on the router. While looking at the configuration on the router, you see that an access list is applied to the Ethernet interface directly connected to the host segment. After reviewing the syntax of the access list, you determine that it is the cause of the failure.

Great—you've found the problem. Now what? Once you find the problem, you must decide what is needed to fix it. In this case, it is an access-list problem, so there are some special considerations about how to restore functionality. You must be careful in your actions here, because that list may contain other entries that provide security or other network administrative functionality. You can't just remove the list—you could cause new problems as you fix the original one.

The best thing to do in this situation is to make a copy of the access list in a text editor, and then make changes that are specific to your problem. When editing the access list, change its number. After all of the changes are made in your text editor, ensure that you have a current backup of the configuration on the router in case you need to restore the original configuration. Then paste the modified access list back into the router. Finally, go to the interface and apply the new access list. By following this procedure, the access list is never removed from the interface.

Obviously, you have now changed the access-list number that is applied to the interface, so any documentation that refers to the original number will need to be updated. If the access list that was causing the problem was applied only to Ethernet 0, you can now safely remove the old list, update this list with the corrections to address your problem, and put it back on the router. Then reapply this list to Ethernet 0. As was the case before, the access list is never removed from the interface.

When you create and implement action plans, it is important that you don't fix one problem and cause another. Before implementing an action plan, think it through or discuss it with coworkers to pick it apart, and make sure that your solution will fix the problem without doing anything to create adverse side effects.

Another good practice, when creating and implementing action plans, is to change only one thing at a time, if possible. If multiple changes must be made, it is best to make the changes in small sets. This way it is easier to keep track of what was done, what worked, and what didn't. The observation step (step 6) becomes much more effective if only a few changes are made at one time; ideally, make only one change at a time. There is nothing worse than troubleshooting your self-induced errors in addition to the original difficulties!

To summarize, follow these practices and guidelines to create a good action plan:

- Make one change or a set of related changes at a time, and then observe the results.
- Make nonimpacting changes—this means trying not to cause other problems while implementing the changes. The more transparent the change, the better.
- Do not create security holes when changing access lists, TACACS+, RADIUS, or other security-oriented configurations.
- Most importantly, make sure you can revert to the original configuration if unforeseen problems occur as a result of the change. Always have a backup or copy of the configuration.

Now that you have reviewed the process of creating and implementing changes, you need to be able to monitor the network and interpret the information to verify whether the changes implemented were effective.

Step 6: Observe Results

Observing results consists of using the exact same methods and commands that were used to obtain information to define the problem—to see whether the changes you implemented had the results you want. By making a change and then testing its effectiveness, you move toward the correct solution.

It may take one or more changes to fix the problem, but you should observe each change separately to monitor progress and to make sure that the alteration doesn't create any adverse effects. After the first change is made, you should be able to gather enough information to learn whether or not the modification was effective, even if it doesn't entirely solve the problem.

Real World Scenario

Looks Can Be Deceiving

One common mistake when observing the results of a change is seeing symptoms go away and interpreting this as the problem's having been solved. For example, assume that users are complaining about slow response time while accessing the Internet. In the course of troubleshooting, you find and correct some non-optimally-configured interface settings on the router on the users' segment. You then go back to the user who originally reported the problem. She reports that everything is running fine now; however, she neglects to mention the fact that there was a shift change and now only two people are connecting to the Internet where there used to be 50. The next day, when all of the users are back online, the problem repeats itself. If an analysis of the observations had been done, it would have demonstrated that the traffic flow to the Internet had dropped off and that this could be a contributing factor to the improvement in response times.

As is demonstrated in this example, failure to analyze your observations creates the risk that important information can be overlooked and the problem will recur. To avoid this possibility, ensure that you look at the entire scope of the problem. Use your network management tools to help you determine whether the problem is really resolved. You can also look at your network baseline information to find out what the "normal" traffic pattern looks like. In this example, it should show a sharp drop-off in utilization when the shift changes. This would tell you that the improvement in connection speed may not be due to the interface changes you've made, but rather are due to a lower volume of traffic—and that more verification is needed.

Not until all of the changes from the action plan are implemented and the results are observed and analyzed can you verify whether the action plan has solved the problem. If it has, move on and document the modifications that were made to the network. If the changes did not work, you need to go back and either gather more information or create a new action plan. These options are explained in more detail in the next section.

Step 7: Iterate as Needed

Iterations, or repetitions of certain steps within the troubleshooting model, are simply ways of whittling away at a larger problem. By implementing action plans and monitoring the results, you can move toward solving the overall problem.

Iterations of the troubleshooting process allow you to focus with more and more detail on the possible causes of the failure. The result of focusing on the problem is your ability to identify more-specific possibilities for the failure.

The iteration process has its own set of steps: While working through the action-plan process, you might get more ideas of possible sources of the trouble. Write them down; if the current action plan doesn't work, you will have notes about some other options. If you feel that you have exhausted all of the possible causes, you should probably go back and gather more information. You will probably find additional clues.

This is also the time to undo any changes that had adverse effects or that did not fix the problem. Make sure to document what was done, so it will be easier to undo the any configuration modifications.

Document the Changes

The network problem has been officially resolved after you've implemented a change, observed that the symptoms have disappeared, and can successfully execute the tests that were used to aid in gathering information about the problem. In this example, the way to verify that the problem is solved is for Host A to try to FTP to Host Z. If this test is successful, then the problem is resolved.

In the previous sections, we have suggested that documentation is an integral part of troubleshooting. When you keep track of the alterations that were made, the routers, switches, or hosts that were changed, and when the changes occurred, you have valuable information for future reference. There is always the possibility that something you changed will have affected something else and you didn't notice it. If this happens, you will have documentation to refer to, so you can undo the changes. Or if a similar problem occurs in the future, you can refer to these documents to resolve the new problem, based on what was done the last time. More on documentation and establishing baseline information will be covered in the next chapters.

Summary

With the complexity of today's networks, it is important to adhere to a troubleshooting model to aid in efficiently and effectively isolating and resolving network problems.

Various methods of problem isolation and the troubleshooting method itself help administrators pinpoint problem areas and foresee future trouble. Troubleshooting skills are gained through experience. It is unreasonable to expect that you can jump in on your first network failure and be able to solve it quickly. Experience is the best teacher. Following a problem-solving model helps you to reach a timely solution to network failures. It helps to know your network, but the "shooting-from-the-hip" style of troubleshooting is nowhere near as effective as a methodical and logical process.

Using the seven steps of the Cisco troubleshooting model in order is a clear, calculated, and logical way to make a network run more smoothly. The three methods of problem isolation (outside-in, inside-out, and divide-by-half) are more subjective, and it is up to each individual to use the method they are comfortable with. It is important to document changes so you have a "trail" of what was done on the network. Finally, it's important to reverse any network alterations that did not correct the problem.

Exam Essentials

Know the seven steps to the Cisco troubleshooting model, as well as the function that each performs. The seven steps to the Cisco troubleshooting model are define the problem, gather facts, determine possible causes, develop an action plan, implement the action plan, observe results, and repeat if necessary. Once a problem is resolved, documentation should be updated.

Know the troubleshooting methodologies and how to use them. These methodologies are outside-in troubleshooting, inside-out troubleshooting, and divide-by-half troubleshooting. In addition to understanding them, know when is most appropriate to use each method.

Be able to apply the Cisco troubleshooting methodology to example situations. Know how to apply each step of the model in real-life scenarios. You should be able to determine what step in a troubleshooting scenario is next in the series, and to correlate a task with the correct step in the process.

Key Terms

Before you take the exam, be certain you are familiar with the following terms:

action plan
boundary of dysfunctionality
divide-by-half troubleshooting
fact gathering
inside-out troubleshooting
iterations
observing results
outside-in troubleshooting
problem definition
troubleshooting model

Review Questions

1. What are valid reasons for using a troubleshooting model? (Choose all that apply.)
   - **A.** Networks are complex and require thorough troubleshooting.
   - **B.** Difficult problems require a systematic and logical method.
   - **C.** Problems are always resolved more quickly by using a systematic model.
   - **D.** Cisco equipment requires diagnostic commands to be entered in a systematic manner.
2. What are the seven steps of the Cisco troubleshooting model? (Choose all that apply.)
   - **A.** Document the changes.
   - **B.** Create a baseline.
   - **C.** Create an action plan.
   - **D.** Undo the wrong changes.
   - **E.** Define the problem.
   - **F.** Observe changes.
   - **G.** Observe results.
   - **H.** Implement an action plan.
   - **I.** Gather facts.
   - **J.** Consider solutions.
   - **K.** Consider possibilities.
   - **L.** Define the problem boundary.
   - **M.** Iterate the process.
3. Place the seven steps of the troubleshooting model in the correct order.
   - **A.** Define Problem, Gather Facts, Consider Possibilities, Create and Implement Action Plan, Observe Results, Iterate Process
   - **B.** Define Problem, Gather Facts, Create and Implement Action Plan, Observe Results, Iterate Process, Consider Possibilities
   - **C.** Consider Possibilities, Gather Facts, Define Problem, Create and Implement Action Plan, Observe Results, Iterate Process
   - **D.** Define Problem, Create and Implement Action Plan, Gather Facts, Consider Possibilities, Observe Results, Iterate Process
4. What is the main purpose of the Define Problem step in the problem-solving model?
   - **A.** To consider the possible causes of the problem
   - **B.** To establish the correct troubleshooting method to be used
   - **C.** To form a specific and concise problem statement that directs the focus of the troubleshooting effort
   - **D.** To diagnose the problem exactly

5. What are the two major reasons for gathering facts when troubleshooting? (Choose two answers.)
   - **A.** To isolate the possible causes of the failure.
   - **B.** To isolate the boundary of the problem.
   - **C.** To isolate the layer 2 from layer 3.
   - **D.** It is required as part of the troubleshooting model.

6. Which of the following types of information are relevant while gathering facts for troubleshooting? (Choose all that apply.)
   - **A.** Network baseline info
   - **B.** The scope of the failure
   - **C.** Whether the trouble is reproducible
   - **D.** The timeline of the failure
   - **E.** Symptoms of the failure

7. In which troubleshooting method do you start near the user and work toward the destination?
   - **A.** Outside-in
   - **B.** Inside-out
   - **C.** Divide-by-half
   - **D.** None of the above

8. Why is establishing the failure boundary important? (Choose all that apply.)
   - **A.** Establishing the failure boundary focuses on the portion of the network or application that is failing.
   - **B.** You know whether you can assign the task to someone else.
   - **C.** It focuses on the relevant information.
   - **D.** It narrows the possibilities for causes of the failure.

9. Why should you gather specific information before considering the possible causes of the failure?
   - **A.** Gathering specific information is part of the process.
   - **B.** It shortens the list of possible causes of failure.
   - **C.** It provides sufficient documentation.
   - **D.** All of the above.

10. A good action plan should follow which of the following guidelines? (Choose all that apply.)
    - **A.** Make one change at a time.
    - **B.** Make any changes necessary to fix the problem.
    - **C.** Make non-service-impacting changes.
    - **D.** Do not create security holes while implementing changes.
    - **E.** Leave an avenue for retreat available, in case you need to back out of the changes you made.

11. During the implementation of an action plan, which of the following is true?

A. Steps should be taken to ensure that additional problems are not caused.

B. Network diagrams should be drawn.

C. Traffic should be isolated to the problem area.

D. You should gather additional facts to see if the current action plan needs to be altered.

12. What are the benefits of the iteration process? (Choose all that apply.)

A. Iteration allows small steps toward resolving a larger network failure.

B. It takes longer to solve the problem, but it is effective.

C. It allows the troubleshooting process to focus on a problem with more and more detail.

D. It allows for ineffective changes to be removed.

13. In which troubleshooting method do you start near the destination and work toward the user?

A. Outside-in

B. Inside-out

C. Divide-by-half

D. None of the above

14. What should you do after implementing the action plan? (Choose all that apply.)

A. Call the user and tell them the problem is solved.

B. Document the changes.

C. Verify that the changes worked without causing additional problems.

D. Iterate the process.

15. SNMP access to a router is no longer working. After this problem has been defined, what would be the next step in troubleshooting according to the Cisco troubleshooting model?

A. Check the copy of the backed-up configuration.

B. Go through the access-list changes that were implemented last night.

C. Ping the router.

D. Search Cisco web site for SNMP bugs.

16. Which of the following is the correct method of isolating the boundary of dysfunctionality? (Choose all that apply.)

A. Divide-by-half

B. Outside-in

C. Inside-out

D. Step-by-step

E. Oudside-down

F. Network-Application

17. Why should you make only one change at a time? (Choose all that apply.)

- **A.** Making one change at a time further isolates the problem.
- **B.** It makes it easier to back out if the change was ineffective.
- **C.** It eliminates one possible cause at a time.
- **D.** All of the above.

18. Which of these tasks is an important part of the iteration process? (Choose all that apply.)

- **A.** Creating more possible causes
- **B.** Gathering more information
- **C.** Homing in on the cause of the failure
- **D.** Creating a new action plan

19. You have implemented an action plan and observed the results, but the original problem still exists. What should your next step be?

- **A.** Repeat the problem-solving process, continuing to change more items until the problem is resolved.
- **B.** Determine whether there are other possibilities for the cause of the problem.
- **C.** Document the changes that have been made and check for bugs on CCO.
- **D.** Repeat the problem-solving process, undoing the changes that were made in the previous attempt.

20. Over the weekend, changes were made in your network to prepare it for an upcoming migration. On Monday, users complain of slow server-response time. What steps should be followed to correct this problem?

- **A.** Back out the changes one at a time.
- **B.** Verify utilization on the user segment.
- **C.** Start working from the beginning of the troubleshooting model.
- **D.** Verify utilization on the server segment.

Answers to Review Questions

1. A, B. Problems may not always be resolved more quickly with a troubleshooting model, but the models are still very efficient. Cisco equipment has no such requirement for troubleshooting.

2. C, E, G, H, I, K, M. Creating a baseline is a good method for identifying problems when they occur, but this is not part of the troubleshooting method. Undoing wrong changes is part of the iteration process—reversing changes is done when a new action plan is created. You can't observe changes, just the results of changes. Solutions also belong to the action-plan step of the troubleshooting method. Defining the problem boundary is part of the fact-gathering process. In addition, documentation should be updated once the changes are complete, but this is not considered part of the model.

3. A. You must create an action plan before its implementation. You cannot consider possibilities if you do not know what the problem is. You cannot create an action plan without knowing the details of the problem.

4. C. Without forming a specific problem statement, it is more difficult to identify possible solutions.

5. A, B. By isolating the possible causes and the boundary of the problem, the possible solutions can be more accurately drawn.

6. A, B, C, D, E. All of these items are very helpful when gathering information about network problems.

7. B. With inside-out troubleshooting, you start where the user or users are experiencing the problem and work toward the service they are attempting to access.

8. A, C, D. In the case of B, instead of assigning the task to someone else if the failure is outside of your jurisdiction, you should coordinate efforts to solve the problem.

9. B. To consider possible causes of failure without having specific information regarding the problem would lead to a very long list of prospects.

10. A, C, D, E. Any changes made should be included in the action plan and documentation. By doing that, you can make sure you hold to the requirements listed in the other answers.

11. A. One of the important considerations in implementing the action plan is to make sure that other users are not impacted by the changes you are making. You should already have your network diagrams, and observing and analyzing the results of the action plan will tell you whether your action plan needs modification.

12. A, C, D. The iteration process actually allows you to shorten the time it would normally take to fix a problem if you were "shooting from the hip."

13. A. Outside-in troubleshooting is a troubleshooting methodology that starts the troubleshooting process at the server side of the problem. This can be a very useful methodology when many users in various locations are having difficulty accessing one server.

14. B, C. The customer should not be told that a solution has been implemented until it has been verified. Iteration, as well, is useless without your first having observed the results of the changes.

15. C. After the problem is defined, the next step in troubleshooting is to gather facts. Determining whether or not the router responds to ping, Telnet, etc., is part of the fact-gathering process. The other three answer options are not part of the fact-gathering stage, but rather are used to determine possible causes or are tasks in creating an action plan.

16. A, B, C. All three of these methods require step-by-step execution.

17. D. These are all correct answers. When multiple changes are made, you can't be sure whether you caused any observed results or which of the changes solved the problem.

18. B, C, D. If you create more possible causes (option A) as part of the iteration process, you are doing something wrong and you need to go back to the very beginning and gather facts regarding the problem.

19. D. If an action plan does not solve the problem, the changes that were made in order to implement this plan should be backed out before proceeding. This ensures that unnecessary changes are not made to the network.

20. C. Though it is tempting to start backing out changes, the correct answer is to start the Cisco troubleshooting model. The changes from the weekend will indeed need to be looked at as the possible cause of the slowdown; however, the cause could be something totally unrelated.