CHAPTER EIGHT

Network Performance Optimization

Objectives

4.5 Explain different methods and rationales for network performance optimization

- Methods:
  - QoS
  - Traffic shaping
  - Load balancing
  - High availability
  - Caching engines
  - Fault tolerance

- Reasons:
  - Latency sensitivity
  - High bandwidth applications
    - VoIP
    - Video applications
  - Uptime

5.1 Given a scenario, select the appropriate command line interface tool and interpret the output to verify functionality

- Traceroute
- Ipconfig
- Iftconfig
- Ping
- ARP ping
- ARP
- Nslookup
Chapter 8: Network Performance Optimization

- Host
- Dig
- Mtr
- Route
- Nbtstat
- Netstat

What You Need to Know

- Understand the importance of data redundancy.
- Explain how the various RAID levels function.
- Understand the difference between fault tolerance and disaster recovery.
- Understand the various backup strategies.
- Identify tape rotation strategies.
- Understand QoS and traffic shaping.
- Use various TCP/IP troubleshooting tools, including ping, tracert, traceroute, arp, netstat, nbtstat, ipconfig, ifconfig, nslookup, dig, mtr, host, and route. Interpret the output from these tools.
- Interpret visual indicators such as LEDs on network devices to help troubleshoot connectivity problems.
- Understand the most common causes of remote connectivity issues, including troubleshooting of Internet access mechanisms such as cable, DSL, and dialup.
- Identify the cause of and remedy for common network client connectivity issues such as authentication failure, permissions issues, and incorrect protocol configurations.

Introduction

As far as network administration goes, nothing is more important than fault tolerance and disaster recovery. First and foremost, it is the responsibility of the network administrator to safeguard the data held on the servers and to ensure that, when requested, this data is ready to go.

Because both fault tolerance and disaster recovery are such an important part of network administration, they are well represented on the CompTIA Network+ exam. In that light, this chapter is important in terms of both real-world application and the exam itself.
What Is Uptime?

All devices on the network, from routers to cabling, and especially servers, must have one prime underlying trait: availability. Networks play such a vital role in the operation of businesses that their availability must be measured in dollars. The failure of a single desktop PC affects the productivity of a single user. The failure of an entire network affects the productivity of the entire company and potentially the company’s clients as well. A network failure might have an even larger impact than that as new e-commerce customers look somewhere else for products, and existing customers start to wonder about the site’s reliability.

Every minute that a network is not running can potentially cost an organization money. The exact amount depends on the role that the server performs and how long it is unavailable. For example, if a small departmental server supporting 10 people goes down for one hour, this might not be a big deal. If the server that runs the company’s e-commerce website goes down for even 10 minutes, it can cost hundreds of thousands of dollars in lost orders.

The importance of data availability varies between networks, but it dictates to what extent a server/network implements fault tolerance measures. The projected capability for a network or network component to weather failure is defined as a number or percentage. The fact that no solution is labeled as providing 100 percent availability indicates that no matter how well we protect our networks, some aspect of the configuration will fail sooner or later.

So how expensive is failure? In terms of equipment replacement costs, it’s not that high. In terms of how much it costs to actually fix the problem, it is a little more expensive. The actual cost of downtime is the biggest factor. For businesses, downtime impacts functionality and productivity of operations. The longer the downtime, the greater the business loss.

Assuming that you know you can never really obtain 100% uptime, what should you aim for? Consider this. If you were responsible for a server system that was available 99.5% of the time, you might be satisfied. But if you realized that you would also have 43.8 hours of downtime each year—that’s one full workweek and a little overtime—you might not be so smug. Table 8.1 compares various levels of downtime.
Chapter 8: Network Performance Optimization

Table 8.1 Levels of Availability and Related Downtime

<table>
<thead>
<tr>
<th>Level of Availability</th>
<th>Availability %</th>
<th>Downtime Per Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commercial availability</td>
<td>99.5%</td>
<td>43.8 hours</td>
</tr>
<tr>
<td>High availability</td>
<td>99.9%</td>
<td>8.8 hours</td>
</tr>
<tr>
<td>Fault-resilient clusters</td>
<td>99.99%</td>
<td>53 minutes</td>
</tr>
<tr>
<td>Fault-tolerant</td>
<td>99.999%</td>
<td>5 minutes</td>
</tr>
<tr>
<td>Continuous</td>
<td>100%</td>
<td>0</td>
</tr>
</tbody>
</table>

These figures make it simple to justify spending money on implementing fault tolerance measures. Just remember that even to reach the definition of commercial availability, you will need to have a range of measures in place. After the commercial availability level, the strategies that take you to each subsequent level are likely to be increasingly expensive, even though they might be easy to justify.

For example, if you estimate that each hour of server downtime will cost the company $1,000, the elimination of 35 hours of downtime—from 43.8 hours for commercial availability to 8.8 hours for high availability—justifies some serious expenditure on technology. Although this first jump is an easily justifiable one, subsequent levels might not be so easy to sell. Working on the same basis, moving from high availability to fault-resilient clusters equates to less than $10,000, but the equipment, software, and skills required to move to the next level will far exceed this figure. In other words, increasing fault tolerance is a law of diminishing returns. As your need to reduce the possibility of downtime increases, so does the investment required to achieve this goal.

The role played by the network administrator in all of this can be somewhat challenging. In some respects, you must function as if you are selling insurance. Informing management of the risks and potential outcomes of downtime can seem a little sensational, but the reality is that the information must be provided if you are to avoid post-event questions about why management was not made aware of the risks. At the same time, a realistic evaluation of exactly the risks presented is needed, along with a realistic evaluation of the amount of downtime each failure might bring.
The Risks

Having established that you need to guard against equipment failure, you can now look at which pieces of equipment are more liable to fail than others. In terms of component failure, the hard disk is responsible for 50 percent of all system downtime. With this in mind, it should come as no surprise that hard disks have garnered the most attention when it comes to fault tolerance. Redundant array of inexpensive disks (RAID), which is discussed in detail in this chapter, is a set of standards that allows servers to cope with the failure of one or more hard disks.

NOTE

Backup Although this chapter discusses various methods of fault tolerance designed to reduce the susceptibility to server failure and downtime, none of these methods is a substitute for a complete and robust backup strategy. No matter how many of these measures are in place, backing up data to an external medium is still the most essential fault tolerance measure.

In fault tolerance, RAID is only half the story. Measures are in place to cope with failures of most other components as well. In some cases, fault tolerance is an elegant solution, and in others, it is a simple case of duplication. We'll start our discussion by looking at RAID, and then we'll move on to other fault tolerance measures.

Fault Tolerance

As far as computers are concerned, fault tolerance refers to the capability of the computer system or network to provide continued data availability in the event of hardware failure. Every component within a server, from the CPU fan to the power supply, has a chance of failure. Some components such as processors rarely fail, whereas hard disk failures are well documented.

Almost every component has fault tolerance measures. These measures typically require redundant hardware components that can easily or automatically take over when a hardware failure occurs.

Of all the components inside computer systems, the hard disks require the most redundancy. Not only are hard disk failures more common than for any other component, but they also maintain the data, without which there would be little need for a network.
Chapter 8: Network Performance Optimization

**Hard Disks Are Half the Problem**

The failure of any component in a computer system is undesirable, but the failure of the component that holds our data is particularly troubling. In fact, hard disks are far more likely to fail than any other component, followed closely by system fans. Why these two? They both have moving parts that can wear out. This is why we spend so much effort ensuring that there are fault tolerance measures for hard disks.

**EXAM ALERT**

For the Network+ exam, remember that the primary function of fault tolerance measures is to allow a system or network to continue operating in the event of unexpected hardware or software errors.

**Disk-Level Fault Tolerance**

Deciding to have *hard disk fault tolerance* on the server is the first step; the second is deciding which fault tolerance strategy to use. Hard disk fault tolerance is implemented according to different RAID levels. Each RAID level offers differing amounts of data protection and performance. The RAID level appropriate for a given situation depends on the importance placed on the data, the difficulty of replacing that data, and the associated costs of a respective RAID implementation. Often, the costs of data loss and replacement outweigh the costs associated with implementing a strong RAID fault tolerance solution.

RAID can be deployed through dedicated hardware, which is more costly, or can be software-based. Today’s network operating systems, such as UNIX and Windows server products, have built-in support for RAID.

**RAID 0: Stripe Set Without Parity**

Although it’s given RAID status, RAID 0 does not actually provide any fault tolerance. In fact, using RAID 0 might even be less fault-tolerant than storing all your data on a single hard disk.

RAID 0 combines unused disk space on two or more hard drives into a single logical volume, with data written to equally sized stripes across all the disks. Using multiple disks, reads and writes are performed simultaneously across all drives. This means that disk access is faster, making the performance of RAID 0 better than other RAID solutions and significantly better than a single hard disk. The downside of RAID 0 is that if any disk in the array fails, the data is lost and must be restored from backup.
Because of its lack of fault tolerance, RAID 0 is rarely implemented. Figure 8.1 shows an example of RAID 0 striping across three hard disks.

**RAID 1**

One of the more common RAID implementations is RAID 1. RAID 1 requires two hard disks and uses disk mirroring to provide fault tolerance. When information is written to the hard disk, it is automatically and simultaneously written to the second hard disk. Both of the hard disks in the mirrored configuration use the same hard disk controller; the partitions used on the hard disk need to be approximately the same size to establish the mirror. In the mirrored configuration, if the primary disk were to fail, the second mirrored disk would contain all the required information, and there would be little disruption to data availability. RAID 1 ensures that the server will continue operating in the case of primary disk failure.

A RAID 1 solution has some key advantages. First, it is cheap in terms of cost per megabyte of storage, because only two hard disks are required to provide fault tolerance. Second, no additional software is required to establish RAID 1, because modern network operating systems have built-in support for it. RAID levels using striping are often incapable of including a boot or system partition in fault tolerance solutions. Finally, RAID 1 offers load balancing over multiple disks, which increases read performance over that of a single disk. Write performance, however, is not improved.

Because of its advantages, RAID 1 is well suited as an entry-level RAID solution, but it has a few significant shortcomings that exclude its use in many environments. It has limited storage capacity—two 100GB hard drives provide only 100GB of storage space. Organizations with large data storage needs can exceed a mirrored solution capacity in very short order. RAID 1 also has a single point of failure, the hard disk controller. If it were to fail, the data would be inaccessible on either drive. Figure 8.2 shows an example of RAID 1 disk mirroring.
An extension of RAID 1 is disk duplexing. Disk duplexing is the same as mirroring, with the exception of one key detail: It places the hard disks on separate hard disk controllers, eliminating the single point of failure.

**EXAM ALERT**

Be aware of the differences between disk duplexing and mirroring for the exam.

**RAID 5**

RAID 5, also known as disk striping with parity, uses distributed parity to write information across all disks in the array. Unlike the striping used in RAID 0, RAID 5 includes parity information in the striping, which provides fault tolerance. This parity information is used to re-create the data in the event of a failure. RAID 5 requires a minimum of three disks, with the equivalent of a single disk being used for the parity information. This means that if you have three 40GB hard disks, you have 80GB of storage space, with the other 40GB used for parity. To increase storage space in a RAID 5 array, you need only add another disk to the array. Depending on the sophistication of the RAID setup you are using, the RAID controller will be able to incorporate the new drive into the array automatically, or you will need to rebuild the array and restore the data from backup.

Many factors have made RAID 5 a very popular fault-tolerant design. RAID 5 can continue to function in the event of a single drive failure. If a hard disk in the array were to fail, the parity would re-create the missing data and continue to function with the remaining drives. The read performance of RAID 5 is improved over a single disk.

**EXAM ALERT**

RAID 5 can continue operation if a hard disk fails. However, administrators have to remember to replace the failed hard disk and restore the RAID 5 configuration. If a second hard disk fails before the failed one is replaced, data loss could occur.
The RAID 5 solution has only a few drawbacks:

- The costs of implementing RAID 5 are initially higher than other fault tolerance measures requiring a minimum of three hard disks. Given the costs of hard disks today, this is a minor concern. However, when it comes to implementing a RAID 5 solution, hardware RAID 5 is more expensive than a software-based RAID 5 solution.

- RAID 5 suffers from poor write performance because the parity has to be calculated and then written across several disks. The performance lag is minimal and doesn’t have a noticeable difference on the network.

- When a new disk is placed in a failed RAID 5 array, there is a regeneration time when the data is being rebuilt on the new drive. This process requires extensive resources from the server.

Figure 8.3 shows an example of RAID 5 striping with parity.

RAID 10

Sometimes RAID levels are combined to take advantage of the best of each. One such strategy is RAID 10, which combines RAID levels 1 and 0. In this configuration, four disks are required. As you might expect, the configuration consists of a mirrored stripe set. To some extent, RAID 10 takes advantage of the performance capability of a stripe set while offering the fault tolerance of a mirrored solution. As well as having the benefits of each, though, RAID 10 inherits the shortcomings of each strategy. In this case, the high overhead and decreased write performance are the disadvantages. Figure 8.4 shows an example of a RAID 10 configuration. Table 8.2 summarizes the various RAID levels.
Chapter 8: Network Performance Optimization

**FIGURE 8.4** Disks in a RAID 10 configuration.

![Disks in a RAID 10 configuration](image)

Data is striped (RAID 0)

Data from stripe set is mirrored (RAID 1)

**NOTE**

RAID levels 2, 3, and 4 are omitted from this discussion because they are used infrequently and rarely, if ever, are seen in modern network environments.

**Table 8.2  Summary of RAID Levels**

<table>
<thead>
<tr>
<th>RAID Level</th>
<th>Description</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Required Disks</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAID 0</td>
<td>Disk striping</td>
<td>Increased read and write performance. RAID 0 can be implemented with two or more disks.</td>
<td>Does not offer any fault tolerance.</td>
<td>Two or more</td>
</tr>
<tr>
<td>RAID 1</td>
<td>Disk mirroring</td>
<td>Provides fault tolerance. Can also be used with separate disk controllers, reducing the single point of failure. This is called disk duplexing.</td>
<td>RAID 1 has 50 percent overhead and suffers from poor write performance.</td>
<td>Two</td>
</tr>
</tbody>
</table>
Table 8.2  Summary of RAID Levels  Continued

<table>
<thead>
<tr>
<th>RAID Level</th>
<th>Description</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Required Disks</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAID 5</td>
<td>Disk striping with distributed parity</td>
<td>Can recover from a single disk failure. Increased read performance over a poor-write single disk. Disks can be added to the array to increase storage capacity.</td>
<td>May slow down the network during regeneration time, and performance may suffer.</td>
<td>Minimum of three</td>
</tr>
<tr>
<td>RAID 10</td>
<td>Striping with mirrored volumes</td>
<td>Increased performance with striping. Offers mirrored fault tolerance.</td>
<td>High overhead, as with mirroring.</td>
<td>Four</td>
</tr>
</tbody>
</table>

Server and Services Fault Tolerance

In addition to providing fault tolerance for individual hardware components, some organizations go the extra mile and include the entire server in the fault-tolerant design. Such a design keeps servers and the services they provide up and running. When it comes to server fault tolerance, two key strategies are commonly employed: standby servers and server clustering.

Standby Servers

Standby servers are a fault tolerance measure in which a second server is configured identically to the first one. The second server can be stored remotely or locally and set up in a failover configuration. In a failover configuration, the secondary server is connected to the primary and is ready to take over the server functions at a moment’s notice. If the secondary server detects that the primary has failed, it automatically cuts in. Network users will not notice the transition, because little or no disruption in data availability occurs.

The primary server communicates with the secondary server by issuing special notification notices called heartbeats. If the secondary server stops receiving the heartbeat messages, it assumes that the primary has died and therefore assumes the primary server configuration.
Chapter 8: Network Performance Optimization

Server Clustering

Companies that want maximum data availability and that have the funds to pay for it can choose to use server clustering. As the name suggests, server clustering involves grouping servers for the purposes of fault tolerance and load balancing. In this configuration, other servers in the cluster can compensate for the failure of a single server. The failed server has no impact on the network, and the end users have no idea that a server has failed.

The clear advantage of server clusters is that they offer the highest level of fault tolerance and data availability. The disadvantage is equally clear—cost. The cost of buying a single server can be a huge investment for many organizations; having to buy duplicate servers is far too costly. In addition to just hardware costs, additional costs can be associated with recruiting administrators who have the skills to configure and maintain complex server clusters. Clustering provides the following advantages:

- **Increased performance**: More servers equals more processing power. The servers in a cluster can provide levels of performance beyond the scope of a single system by combining resources and processing power.

- **Load balancing**: Rather than having individual servers perform specific roles, a cluster can perform a number of roles, assigning the appropriate resources in the best places. This approach maximizes the power of the systems by allocating tasks based on which server in the cluster can best service the request.

- **Failover**: Because the servers in the cluster are in constant contact with each other, they can detect and cope with the failure of an individual system. How transparent the failover is to users depends on the clustering software, the type of failure, and the capability of the application software being used to cope with the failure.

- **Scalability**: The capability to add servers to the cluster offers a degree of scalability that is simply not possible in a single-server scenario. It is worth mentioning, though, that clustering on PC platforms is still in its relative infancy, and the number of machines that can be included in a cluster is still limited.

To make server clustering happen, you need certain ingredients—servers, storage devices, network links, and software that makes the cluster work.
Fault Tolerance

Link Redundancy

Although a failed network card might not actually stop the server or a system, it might as well. A network server that cannot be used on the network results in server downtime. Although the chances of a failed network card are relatively low, attempts to reduce the occurrence of downtime have led to the development of a strategy that provides fault tolerance for network connections.

Through a process called adapter teaming, groups of network cards are configured to act as a single unit. The teaming capability is achieved through software, either as a function of the network card driver or through specific application software. The process of adapter teaming is not widely implemented; although the benefits it offers are many, so it’s likely to become a more common sight. The result of adapter teaming is increased bandwidth, fault tolerance, and the ability to manage network traffic more effectively. These features are broken into three sections:

- **Adapter fault tolerance**: The basic configuration enables one network card to be configured as the primary device and others as secondary. If the primary adapter fails, one of the other cards can take its place without the need for intervention. When the original card is replaced, it resumes the role of primary controller.

- **Adapter load balancing**: Because software controls the network adapters, workloads can be distributed evenly among the cards so that each link is used to a similar degree. This distribution allows for a more responsive server because one card is not overworked while another is underworked.

- **Link aggregation**: This provides vastly improved performance by allowing more than one network card's bandwidth to be aggregated—combined into a single connection. For example, through link aggregation, four 100Mbps network cards can provide a total of 400Mbps of bandwidth. Link aggregation requires that both the network adapters and the switch being used support it. In 1999, the IEEE ratified the 802.3ad standard for link aggregation, allowing compatible products to be produced.
Chapter 8: Network Performance Optimization

Using Uninterruptible Power Supplies

No discussion of fault tolerance can be complete without a look at power-related issues and the mechanisms used to combat them. When you’re designing a fault-tolerant system, your planning should definitely include uninterruptible power supplies (UPSs). A UPS serves many functions and is a major part of server consideration and implementation.

On a basic level, a UPS is a box that holds a battery and built-in charging circuit. During times of good power, the battery is recharged; when the UPS is needed, it’s ready to provide power to the server. Most often, the UPS is required to provide enough power to give the administrator time to shut down the server in an orderly fashion, preventing any potential data loss from a dirty shutdown.

Why Use a UPS?

Organizations of all shapes and sizes need UPSs as part of their fault tolerance strategies. A UPS is as important as any other fault tolerance measure. Three key reasons make a UPS necessary:

- **Data availability**: The goal of any fault tolerance measure is data availability. A UPS ensures access to the server in the event of a power failure—or at least as long as it takes to save a file.

- **Protection from data loss**: Fluctuations in power or a sudden power-down can damage the data on the server system. In addition, many servers take full advantage of caching, and a sudden loss of power could cause the loss of all information held in cache.

- **Protection from hardware damage**: Constant power fluctuations or sudden power-downs can damage hardware components within a computer. Damaged hardware can lead to reduced data availability while the hardware is being repaired.

Power Threats

In addition to keeping a server functioning long enough to safely shut it down, a UPS safeguards a server from inconsistent power. This inconsistent power can take many forms. A UPS protects a system from the following power-related threats:

- **Blackout**: A total failure of the power supplied to the server.

- **Spike**: A spike is a very short (usually less than a second) but very intense increase in voltage. Spikes can do irreparable damage to any kind of equipment, especially computers.
Disaster Recovery

- **Surge**: Compared to a spike, a surge is a considerably longer (sometimes many seconds) but usually less intense increase in power. Surges can also damage your computer equipment.
- **Sag**: A sag is a short-term voltage drop (the opposite of a spike). This type of voltage drop can cause a server to reboot.
- **Brownout**: A brownout is a drop in voltage that usually lasts more than a few minutes.

Many of these power-related threats can occur without your knowledge; if you don’t have a UPS, you cannot prepare for them. For the cost, it is worth buying a UPS, if for no other reason than to sleep better at night.

**Disaster Recovery**

Even the most fault-tolerant networks will fail, which is an unfortunate fact. When those costly and carefully implemented fault tolerance strategies fail, you are left with disaster recovery.

Disaster recovery can take many forms. In addition to disasters such as fire, flood, and theft, many other potential business disruptions can fall under the banner of disaster recovery. For example, the failure of the electrical supply to your city block might interrupt the business functions. Such an event, although not a disaster per se, might invoke the disaster recovery methods.

The cornerstone of every disaster recovery strategy is the preservation and recoverability of data. When talking about preservation and recoverability, we are talking about backups. When we are talking about backups, we are likely talking about tape backups. Implementing a regular backup schedule can save you a lot of grief when fault tolerance fails or when you need to recover a file that has been accidentally deleted. When it comes time to design a backup schedule, three key types of backups are used—full, differential, and incremental.

**Full Backup**

The preferred method of backup is the full backup method, which copies all files and directories from the hard disk to the backup media. There are a few reasons why doing a full backup is not always possible. First among them is likely the time involved in performing a full backup.
Chapter 8: Network Performance Optimization

Depending on the amount of data to be backed up, full backups can take an extremely long time and can use extensive system resources. Depending on the configuration of the backup hardware, this can slow down the network considerably. In addition, some environments have more data than can fit on a single tape. This makes doing a full backup awkward, because someone might need to be there to change the tapes.

The main advantage of full backups is that a single tape or tape set holds all the data you need backed up. In the event of a failure, a single tape might be all that is needed to get all data and system information back. The upshot of all this is that any disruption to the network is greatly reduced.

Unfortunately, its strength can also be its weakness. A single tape holding an organization’s data can be a security risk. If the tape were to fall into the wrong hands, all the data could be restored on another computer. Using passwords on tape backups and using a secure offsite and onsite location can minimize the security risk.

**Differential Backup**

Companies that just don’t have enough time to complete a full backup daily can make use of the *differential backup*. Differential backups are faster than a full backup, because they back up only the data that has changed since the last full backup. This means that if you do a full backup on a Saturday and a differential backup on the following Wednesday, only the data that has changed since Saturday is backed up. Restoring the differential backup requires the last full backup and the latest differential backup.

Differential backups know what files have changed since the last full backup because they use a setting called the *archive bit*. The archive bit flags files that have changed or have been created and identifies them as ones that need to be backed up. Full backups do not concern themselves with the archive bit, because all files are backed up, regardless of date. A full backup, however, does clear the archive bit after data has been backed up to avoid future confusion. Differential backups take notice of the archive bit and use it to determine which files have changed. The differential backup does not reset the archive bit information.
The faster backup time of incremental backups comes at a price—the amount of time required to restore. Recovering from a failure with incremental backups requires numerous tapes—all the incremental tapes and the most recent full backup. For example, if you had a full backup from Sunday and an incremental for Monday, Tuesday, and Wednesday, you would need four tapes to restore the data. Keep in mind that each tape in the rotation is an additional step in the restore process and an additional failure point. One damaged incremental tape, and you will be unable to restore the data. Table 8.3 summarizes the various backup strategies.

### Table 8.3 Backup Strategies

<table>
<thead>
<tr>
<th>Backup Type</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Data Backed Up</th>
<th>Archive Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>Backs up all data on a single tape or tape set. Restoring data requires the fewest tapes.</td>
<td>Depending on the amount of data, full backups can take a long time.</td>
<td>All files and directories are backed up.</td>
<td>Does not use the archive bit, but resets it after data has been backed up.</td>
</tr>
<tr>
<td>Incremental</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CAUTION

If you experience trouble with any type of backup, you should clean the tape drive and then try the backup again. Also visually inspect the tape for physical damage.

CAUTION

Both full and incremental backups clear the archive bit after files have been backed up.

Incremental Backup

Some companies have a finite amount of time they can allocate to backup procedures. Such organizations are likely to use incremental backups in their backup strategy. Incremental backups save only the files that have changed since the last full or incremental backup. Like differential backups, incremental backups use the archive bit to determine which files have changed since the last full or incremental backup. Unlike differentials, however, incremental backups clear the archive bit, so files that have not changed are not backed up.

The faster backup time of incremental backups comes at a price—the amount of time required to restore. Recovering from a failure with incremental backups requires numerous tapes—all the incremental tapes and the most recent full backup. For example, if you had a full backup from Sunday and an incremental for Monday, Tuesday, and Wednesday, you would need four tapes to restore the data. Keep in mind that each tape in the rotation is an additional step in the restore process and an additional failure point. One damaged incremental tape, and you will be unable to restore the data. Table 8.3 summarizes the various backup strategies.
Table 8.3 Backup Strategies  
Continued

<table>
<thead>
<tr>
<th>Backup Type</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Data Backed Up</th>
<th>Archive Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Differential</td>
<td>Faster backups than a full backup.</td>
<td>Uses more tapes than a full backup. The restore process takes longer than a full backup.</td>
<td>All files and directories that have changed since the last full or differential backup.</td>
<td>Uses the archive bit to determine the files that have changed, but does not reset the archive bit.</td>
</tr>
<tr>
<td>Incremental</td>
<td>Faster backup times.</td>
<td>Requires multiple disks; restoring data takes more time than the other backup methods.</td>
<td>The files and directories that have changed since the last full or incremental backup.</td>
<td>Uses the archive bit to determine the files that have changed, and resets the archive bit.</td>
</tr>
</tbody>
</table>

**Tape Rotations**

After you select a backup type, you are ready to choose a backup rotation. Several backup rotation strategies are in use—some good, some bad, and some really bad. The most common, and perhaps the best, rotation strategy is grand-father, father, son (GFS).

The GFS backup rotation is the most widely used—and for good reason. For example, a GFS rotation may require 12 tapes: four tapes for daily backups (son), five tapes for weekly backups (father), and three tapes for monthly backups (grandfather).

Using this rotation schedule, you can recover data from days, weeks, or months earlier. Some network administrators choose to add tapes to the monthly rotation so that they can retrieve data even further back, sometimes up to a year. In most organizations, however, data that is a week old is out of date, let alone six months or a year.

**Backup Best Practices**

Many details go into making a backup strategy a success. The following are issues to consider as part of your backup plan:

- **Offsite storage:** Consider storing backup tapes offsite so that in the event of a disaster in a building, a current set of tapes is still available offsite. The offsite tapes should be as current as any onsite and should be secure.
Label tapes: The goal is to restore the data as quickly as possible, and trying to find the tape you need can be difficult if it isn’t marked. Furthermore, this can prevent you from recording over a tape you need.

New tapes: Like old cassette tapes, the tape cartridges used for the backups wear out over time. One strategy used to prevent this from becoming a problem is to periodically introduce new tapes into the rotation schedule.

Verify backups: Never assume that the backup was successful. Seasoned administrators know that checking backup logs and performing periodic test restores are part of the backup process.

Cleaning: You need to clean the tape drive occasionally. If the inside gets dirty, backups can fail.

CAUTION
A backup strategy must include offsite storage to account for theft, fire, flood, or other disasters.

Hot and Cold Spares

The impact that a failed component has on a system or network depends largely on the predisaster preparation and on the recovery strategies used. Hot and cold spares represent a strategy for recovering from failed components.

Hot Spares and Hot Swapping

Hot spares allow system administrators to quickly recover from component failure. In a common use, a hot spare enables a RAID system to automatically fail over to a spare hard drive should one of the other drives in the RAID array fail.

A hot spare does not require any manual intervention. Instead, a redundant drive resides in the system at all times, just waiting to take over if another drive fails. The hot spare drive takes over automatically, leaving the failed drive to be removed later. Even though hot-spare technology adds an extra level of protection to your system, after a drive has failed and the hot spare has been used, the situation should be remedied as soon as possible.

Hot swapping is the ability to replace a failed component while the system is running. Perhaps the most commonly identified hot-swap component is the hard drive. In certain RAID configurations, when a hard drive crashes, hot swapping allows you to simply take the failed drive out of the server and install a new one.
Chapter 8: Network Performance Optimization

The benefits of hot swapping are very clear in that it allows a failed component to be recognized and replaced without compromising system availability. Depending on the system's configuration, the new hardware normally is recognized automatically by both the current hardware and the operating system. Nowadays, most internal and external RAID subsystems support the hot-swapping feature. Some hot-swappable components include power supplies and hard disks.

**Cold Spares and Cold Swapping**

The term *cold spare* refers to a component, such as a hard disk, that resides within a computer system but requires manual intervention in case of component failure. A hot spare engages automatically, but a cold spare might require configuration settings or some other action to engage it. A cold spare configuration typically requires a reboot of the system.

The term cold spare has also been used to refer to a redundant component that is stored outside the actual system but is kept in case of component failure. To replace the failed component with a cold spare, you need to power down the system.

Cold swapping refers to replacing components only after the system is completely powered off. This strategy is by far the least attractive for servers, because the services provided by the server are unavailable for the duration of the cold-swap procedure. Modern systems have come a long way to ensure that cold swapping is a rare occurrence. For some situations and for some components, however, cold swapping is the only method to replace a failed component. The only real defense against having to shut down the server is to have redundant components reside in the system.

**NOTE**

The term "warm swap" is applied to a device that can be replaced while the system is still running but that requires some kind of manual intervention to disable the device before it can be removed. Using a PCI hot plug technically is a warm-swap strategy, because it requires that the individual PCI slot be powered down before the PCI card is replaced. Of course, a warm swap is not as efficient as a hot swap, but it is far and away better than a cold swap.

**Hot, Warm, and Cold Sites**

A disaster recovery plan might include the provision for a recovery site that can be brought into play quickly. These sites fall into three categories: hot, warm,
Disaster recovery sites represent the ultimate in precautions for organizations that really need them. As a result, they don’t come cheap.

The basic concept of a disaster recovery site is that it can provide a base from which the company can be operated during a disaster. The disaster recovery site normally is not intended to provide a desk for every employee. It’s intended more as a means to allow key personnel to continue the core business functions.

In general, a cold recovery site is a site that can be up and operational in a relatively short amount of time, such as a day or two. Provision of services, such as telephone lines and power, is taken care of, and the basic office furniture might be in place. But there is unlikely to be any computer equipment, even though the building might well have a network infrastructure and a room ready to act as a server room. In most cases, cold sites provide the physical location and basic services.

Cold sites are useful if you have some forewarning of a potential problem. Generally speaking, cold sites are used by organizations that can weather the storm for a day or two before they get back up and running. If you are the regional office of a major company, it might be possible to have one of the other divisions take care of business until you are ready to go. But if you are the one and only office in the company, you might need something a little hotter.

For organizations with the dollars and the desire, hot recovery sites represent the ultimate in fault tolerance strategies. Like cold recovery sites, hot sites are designed to provide only enough facilities to continue the core business function, but hot recovery sites are set up to be ready to go at a moment’s notice.

A hot recovery site includes phone systems with the phone lines already connected. Data networks also are in place, with any necessary routers and switches plugged in and turned on. Desks have desktop PCs installed and waiting, and server areas are replete with the necessary hardware to support business-critical functions. In other words, within a few hours, the hot site can become a fully functioning element of an organization.

The issue that confronts potential hot-recovery site users is simply that of cost. Office space is expensive in the best of times, but having space sitting idle 99.9 percent of the time can seem like a tremendously poor use of money. A very popular strategy to get around this problem is to use space provided in a disaster recovery facility, which is basically a building, maintained by a third-party company, in which various businesses rent space. Space is usually apportioned according to how much each company pays.
Chapter 8: Network Performance Optimization

Sitting between the hot and cold recovery sites is the warm site. A warm site typically has computers but is not configured ready to go. This means that data might need to be upgraded or other manual interventions might need to be performed before the network is again operational. The time it takes to get a warm site operational lands right in the middle of the other two options, as does the cost.

**EXAM ALERT**

A hot site that mirrors the organization’s production network can assume network operations at a moment’s notice. Warm sites have the equipment needed to bring the network to an operational state but require configuration and potential database updates. A cold site has the space available with basic service but typically requires equipment delivery.

**Network Optimization Strategies**

Today’s networks are all about speed. Network users expect data and application delivery quickly. Just look at how impatient many of us get waiting for web pages to load. Networks, however, are saturated and congested with traffic, making it necessary to have strategies to ensure that we are using bandwidth in the best possible way. These strategies are collectively referred to as quality of service (QoS).

**QoS**

QoS describes the strategies used to manage and increase the flow of network traffic. QoS features allow administrators to predict bandwidth use, monitor that use, and control it to ensure that bandwidth is available to the applications that need it. These applications generally can be broken into two categories:

- **Latency-sensitive**: These applications need bandwidth for quick delivery where network lag time impacts their effectiveness. This includes voice and video transfer. For example, voice over IP (VoIP) would be difficult to use if there were a significant lag time in the conversation.

- **Latency-insensitive**: Controlling bandwidth also involves managing latency-insensitive applications. This includes bulk data transfers such as huge backup procedures and File Transfer Protocol (FTP) transfers.
With bandwidth being limited, and networks becoming increasingly congested, it becomes more difficult to deliver latency-sensitive traffic. If network traffic continues to increase and we can't always increase bandwidth, the choice is to prioritize traffic to ensure timely delivery. This is where QoS really comes into play. QoS ensures the delivery of applications, such as videoconferencing and VoIP telephony, without adversely affecting network throughput. QoS achieves more efficient use of network resources by differentiating between latency-insensitive traffic such as fax data and latency-sensitive streaming media.

One important strategy for QoS is priority queuing. Essentially what happens is that traffic is placed in order based on its importance on delivery time. All data is given access, but the more important and latency-sensitive data is given higher priority.

Traffic Shaping

Traffic shaping is a QoS strategy that is designed to enforce prioritization policies on the transmission of data throughout the network. It is intended to reduce latency by controlling the amount of data that flows into and out of the network. Traffic is categorized, queued, and directed according to network policies.

Methods Used for Traffic Shaping

You can shape and limit network traffic using several different strategies. Which one you choose depends on the network's needs and the amount of traffic. Here are some common traffic-shaping methods:

- **Shaping by application:** Administrators can configure a traffic shaper by categorizing specific types of network traffic and assigning that category a bandwidth limit. For example, traffic can be categorized using FTP. The rule can specify that no more than 4Mbps be dedicated for FTP traffic. This same principle can apply to Telnet sessions, streaming audio, or any other application coming through the network.

- **Shaping network traffic per user:** Every network has users who use more bandwidth than others. Some of this might be work-related, but more often than not, it is personal use. In such a case, it may be necessary to establish traffic shaping on a per-user basis. Traffic shapers allow...
administrators to delegate a certain bandwidth to a user. For instance, Bob from accounting is allowed no more than 256Kbps. This doesn’t limit what the user can access, just the speed at which that content can be accessed.

- **Priority shaping:** One important consideration when looking at traffic shaping is determining which traffic is mission-critical and which is less so. In addition to setting hard or burstable traffic limits on a per-application or per-user basis, traffic shaping devices can also be used to define the relative importance, or priority, of different types of traffic. For example, in an academic network where teaching and research are most important, recreational uses of the network (such as network games or peer-to-peer file-sharing application traffic) can be allowed bandwidth only when higher-priority applications don’t need it.

---

**EXAM ALERT**

For the Network+ exam, remember that traffic shaping involves delaying the flow of data traffic that is designated as less important compared to other traffic streams.

---

**Caching Engines**

Caching is an important consideration when optimizing network traffic. For example, as discussed in Chapter 3, “Networking Components and Devices,” proxy servers use caching to limit the number of client requests that go to the Internet. Instead, the requests are filled from the proxy server's cache. Recall from Chapter 3 that when a caching proxy server has answered a request for a web page, the server makes a copy of all or part of that page in its cache. Then, when the page is requested again, the proxy server answers the request from the cache rather than going back out to the Internet. For example, if a client on a network requests the web page www.comptia.org, the proxy server can cache the contents of that web page. When a second client computer on the network attempts to access the same site, that client can grab it from the proxy server cache; accessing the Internet is not necessary. This greatly reduces the network traffic that has to be filtered to the Internet, a significant gain in terms of network optimization.

When it comes to determining what to cache, an administrator can establish many rules:

- What websites to cache
- How long the information is cached
When cached information is updated
- The size of cached information
- What type of content is cached
- Who can access the cache

The rules for caching vary from network to network, depending on the network’s needs. In networks where a large number of people access similar websites, caching can greatly increase network performance. The advantages of properly configured caching are clear—reduced bandwidth, and latency and increased throughput. One possible disadvantage of caching is receiving out-of-date files, because you are obtaining content from the caching engine and not the website itself.

NOTE

Caching improves network performance by caching content locally, thereby limiting surges in traffic.

**Working with Command-Line Utilities**

For anyone working with TCP/IP networks, troubleshooting connectivity is something that simply must be done. This section describes the tools that are used in the troubleshooting process and identifies scenarios in which they can be used.

**Troubleshooting with Diagnostic Utilities**

You can use many utilities when troubleshooting TCP/IP. Although the actual utilities available vary from platform to platform, the functionality between platforms is quite similar. Table 8.4 lists the TCP/IP troubleshooting tools covered on the Network+ exam, along with their purpose.
Table 8.4 Common TCP/IP Troubleshooting Tools and Their Purposes

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tracert/traceroute</td>
<td>Used to track the path a packet takes as it travels across a network. tracert is used on Windows systems; traceroute is used on UNIX, Linux, and Macintosh systems.</td>
</tr>
<tr>
<td>ping</td>
<td>Used to test connectivity between two devices on a network.</td>
</tr>
<tr>
<td>arp</td>
<td>Used to view and work with the IP address to MAC address resolution cache.</td>
</tr>
<tr>
<td>arp ping</td>
<td>Uses ARP to test connectivity between systems rather than using Internet Control Message Protocol (ICMP), as done with a regular ping.</td>
</tr>
<tr>
<td>netstat</td>
<td>Used to view the current TCP/IP connections on a system.</td>
</tr>
<tr>
<td>nbtstat</td>
<td>Used to view statistics related to NetBIOS name resolution and to see information about current NetBIOS over TCP/IP connections.</td>
</tr>
<tr>
<td>ipconfig</td>
<td>Used to view and renew TCP/IP configuration on a Windows system.</td>
</tr>
<tr>
<td>ifconfig</td>
<td>Used to view TCP/IP configuration on a UNIX, Linux, or Macintosh system.</td>
</tr>
<tr>
<td>nslookup/dig</td>
<td>Used to perform manual DNS lookups. nslookup can be used on Windows, UNIX, Macintosh, and Linux systems. dig can be used on UNIX, Linux, and Macintosh systems.</td>
</tr>
<tr>
<td>mtr</td>
<td>Used on Linux/UNIX systems. Essentially combines the functionality of ping with that of traceroute.</td>
</tr>
<tr>
<td>host</td>
<td>Used on Linux/UNIX systems to perform a reverse lookup on an IP address.</td>
</tr>
<tr>
<td>route</td>
<td>Used to view and configure the routes in the routing table.</td>
</tr>
</tbody>
</table>

The following sections look in more detail at these utilities and the output they produce.

**CAUTION**

Many of the utilities discussed in this chapter have a help facility that you can access by typing the command followed by `/?` or `-?`. On a Windows system, for example, you can get help on the `netstat` utility by typing `netstat /?`. Sometimes, using a utility with an invalid switch also brings up the help screen.
The Trace Route Utility (tracert/traceroute)

The trace route utility does exactly what its name implies—it traces the route between two hosts. It does this by using ICMP echo packets to report information at every step in the journey. Each of the common network operating systems provides a trace route utility, but the name of the command and the output vary slightly on each. However, for the purposes of the Network+ exam, you should not concern yourself with the minor differences in the output format. Table 8.5 shows the trace route command syntax used in various operating systems.

Table 8.5 Trace Route Utility Commands

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Trace Route Command Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows systems</td>
<td>tracert IP address</td>
</tr>
<tr>
<td>Novell NetWare</td>
<td>iptrace</td>
</tr>
<tr>
<td>Linux/UNIX</td>
<td>traceroute IP address</td>
</tr>
<tr>
<td>Macintosh</td>
<td>traceroute IP address</td>
</tr>
</tbody>
</table>
Trace route provides a lot of useful information, including the IP address of every router connection it passes through and, in many cases, the name of the router (although this depends on the router’s configuration). Trace route also reports the length, in milliseconds, of the round-trip the packet made from the source location to the router and back. This information can help identify where network bottlenecks or breakdowns might be. The following is an example of a successful tracert command on a Windows Server system:

```
C:\>tracert 24.7.70.37
Tracing route to c1-p4.sttlwa1.home.net [24.7.70.37]
over a maximum of 30 hops:
  1  30 ms  20 ms  20 ms  24.67.184.1
  2  20 ms  20 ms  30 ms  rd1ht-ge3-0.ok.shawcable.net [24.67.224.7]
  3  50 ms  30 ms  30 ms  rc1wh-atm0-2-1.vc.shawcable.net [204.209.214.193]
  4  50 ms  30 ms  30 ms  rc2wh-pos15-0.vc.shawcable.net [204.209.214.90]
  5  30 ms  40 ms  30 ms  rc2wt-pos2-0.wa.shawcable.net [66.163.76.37]
  6  30 ms  40 ms  30 ms  c1-pos6-3.sttlwa1.home.net [24.7.70.37]
Trace complete.
```

Similar to the other common operating systems covered on the Network+ exam, the tracert display on a Windows-based system includes several columns of information. The first column represents the hop number. You may recall that hop is the term used to describe a step in the path a packet takes as it crosses the network. The next three columns indicate the round-trip time, in milliseconds, that a packet takes in its attempts to reach the destination. The last column is the hostname and the IP address of the responding device.

Of course, not all trace route attempts are successful. The following is the output from a tracert command on a Windows Server system that doesn’t manage to get to the remote host:

```
C:\>tracert comptia.org
Tracing route to comptia.org [216.119.103.72]
over a maximum of 30 hops:
  1  27 ms  28 ms  14 ms  24.67.179.1
  2  55 ms  13 ms  14 ms  rd1ht-ge3-0.ok.shawcable.net [24.67.224.7]
  3  27 ms  27 ms  28 ms  rc1wh-atm0-2-1.shawcable.net [204.209.214.19]
```
Working with Command-Line Utilities

In this example, the trace route request gets to only the seventh hop, at which point it fails. This failure indicates that the problem lies on the far side of the device in step 7 or on the near side of the device in step 8. In other words, the device at step 7 is functioning but might not be able to make the next hop. The cause of the problem could be a range of things, such as an error in the routing table or a faulty connection. Alternatively, the seventh device might be operating at 100 percent, but device 8 might not be functioning at all. In any case, you can isolate the problem to just one or two devices.

NOTE

In some cases, the owner of a router might configure it to not return ICMP traffic like that generated by ping or traceroute. If this is the case, the ping or traceroute will fail just as if the router did not exist or was not operating.

CAUTION

Although we have used the Windows tracert command to provide sample output in these sections, the output from traceroute on a UNIX, Linux, or Macintosh system is extremely similar.

The trace route utility can also help you isolate a heavily congested network. In the following example, the trace route packets fail in the midst of the tracert from a Windows Server system, but subsequently they continue. This behavior can be an indicator of network congestion:

C:\> tracert comptia.org
Tracing route to comptia.org [216.119.103.72] over a maximum of 30 hops:
1    96 ms  96 ms  55 ms  24.67.179.1
2    14 ms  13 ms  28 ms  rd1ht-g3-0.ok.shawcable.net [24.67.224.7]
3    28 ms  27 ms  41 ms  rc1wh-atm0-2-1.shawcable.net [204.209.214.19]
4    28 ms  41 ms  27 ms  rc1wt-pos2-0.wa.shawcable.net [66.163.76.65]
5    41 ms  27 ms  27 ms  rc2wt-pos1-0.wa.shawcable.net [66.163.68.2]
6    55 ms  41 ms  27 ms  c1-pos6-3.sttlwa1.home.net [24.7.70.37]
7    54 ms  42 ms  27 ms  home-gw.st6wa.ip.att.net [192.205.32.249]
8    55 ms  41 ms  28 ms  gbr3-p40.st6wa.ip.att.net [12.123.44.138]
9    *    *    *    Request timed out.
10   *    *    *    Request timed out.
11   *    *    *    Request timed out.
12   *    *    *    Request timed out.
13   69 ms  68 ms  69 ms  gbr2-p28.sd2ca.ip.att.net [12.122.11.254]
14   55 ms  68 ms  69 ms  gbr1-p68.sd2ca.ip.att.net [12.122.1.109]
15   82 ms  69 ms  82 ms  gbr1-p38.phmaz.ip.att.net [12.122.2.142]
16   68 ms  69 ms  82 ms  gar2-p360.phmaz.ip.att.net [12.123.142.45]
17   110 ms 96 ms  96 ms  12.125.99.70
18   124 ms 96 ms  96 ms  light.crystaltech.com [216.119.107.1]
19   82 ms  96 ms  96 ms  216.119.103.72
Trace complete.

Generally speaking, trace route utilities allow you to identify the location of a problem in the connectivity between two devices. After you have determined this location, you might need to use a utility such as ping to continue troubleshooting. In many cases, as in the examples provided in this chapter, the routers might be on a network such as the Internet and therefore not within your control. In that case, you can do little except inform your ISP of the problem.

ping

Most network administrators are very familiar with the ping utility and are likely to use it on an almost daily basis. The basic function of the ping command is to test the connectivity between the two devices on a network. All the command is designed to do is determine whether the two computers can see each other and to notify you of how long the round-trip takes to complete.

Although ping is most often used on its own, a number of switches can be used to assist in the troubleshooting process. Table 8.6 shows some of the commonly used switches with ping on a Windows system.
Table 8.6 ping Command Switches

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ping -t</td>
<td>Pings a device on the network until stopped.</td>
</tr>
<tr>
<td>ping -a</td>
<td>Resolves addresses to hostnames.</td>
</tr>
<tr>
<td>ping -n count</td>
<td>Specifies the number of echo requests to send.</td>
</tr>
<tr>
<td>ping -r count</td>
<td>Records the route for count hops.</td>
</tr>
<tr>
<td>ping -s count</td>
<td>Timestamp for count hops.</td>
</tr>
<tr>
<td>ping -w timeout</td>
<td>Timeout in milliseconds to wait for each reply.</td>
</tr>
</tbody>
</table>

EXAM ALERT
The Network+ exam is likely to ask questions about ping, its switches used and how ping can be used in a troubleshooting scenario.

ping works by sending ICMP echo request messages to another device on the network. If the other device on the network hears the ping request, it automatically responds with an ICMP echo reply. By default, the ping command on a Windows-based system sends four data packets; however, using the -t switch, a continuous stream of ping requests can be sent.

ping is perhaps the most widely used of all network tools; it is primarily used to verify connectivity between two network devices. On a good day, the results from the ping command are successful, and the sending device receives a reply from the remote device. Not all ping results are that successful. To be able to use ping effectively, you must be able to interpret the results of a failed ping command.

The Destination Host Unreachable Message
The Destination host unreachable error message means that a route to the destination computer system cannot be found. To remedy this problem, you might need to examine the routing information on the local host to confirm that the local host is correctly configured, or you might need to make sure that the default gateway information is correct. The following is an example of a ping failure that gives the Destination host unreachable message:

Pinging 24.67.54.233 with 32 bytes of data:
Destination host unreachable.
Destination host unreachable.
Destination host unreachable.
Destination host unreachable.
Ping statistics for 24.67.54.233:
Chapter 8: Network Performance Optimization

Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Approximate round trip times in milliseconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

The Request Timed Out Message
The request timed out error message is very common when you use the ping command. Essentially, this error message indicates that your host did not receive the ping message back from the destination device within the designated time period. Assuming that the network connectivity is okay on your system, this typically indicates that the destination device is not connected to the network, is powered off, or is not configured correctly. It could also mean that some intermediate device is not operating correctly. In some rare cases, it can also indicate that the network has so much congestion that timely delivery of the ping message could not be completed. It might also mean that the ping is being sent to an invalid IP address or that the system is not on the same network as the remote host, and an intermediary device is not configured correctly. In any of these cases, the failed ping should initiate a troubleshooting process that might involve other tools, manual inspection, and possibly reconfiguration. The following example shows the output from a ping to an invalid IP address:

```
C:\> ping 169.76.54.3
Pinging 169.76.54.3 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 169.76.54.3:
Packets: Sent = 4, Received = 0, Lost = 4 (100%)
Approximate round trip times in milliseconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

During the ping request, you might receive some replies from the remote host that are intermixed with request timed out errors. This is often the result of a congested network. An example follows; notice that this example, which was run on a Windows Vista system, uses the -t switch to generate continuous pings:

```
C:\> ping -t 24.67.184.65
Pinging 24.67.184.65 with 32 bytes of data:
Reply from 24.67.184.65: bytes=32 time=55ms TTL=127
Reply from 24.67.184.65: bytes=32 time=54ms TTL=127
Reply from 24.67.184.65: bytes=32 time=27ms TTL=127
Request timed out.
Request timed out.
```
Request timed out.
Reply from 24.67.184.65: bytes=32 time=69ms TTL=127
Reply from 24.67.184.65: bytes=32 time=28ms TTL=127
Reply from 24.67.184.65: bytes=32 time=28ms TTL=127
Reply from 24.67.184.65: bytes=32 time=68ms TTL=127
Reply from 24.67.184.65: bytes=32 time=41ms TTL=127

Ping statistics for 24.67.184.65:
  Packets: Sent = 11, Received = 8, Lost = 3 (27% loss),
Approximate round trip times in milli-seconds:
  Minimum = 27ms, Maximum =  69ms, Average =  33ms

In this example, three packets were lost. If this continued on your network, you
would need to troubleshoot to find out why packets were being dropped.

The Unknown Host Message
The Unknown host error message is generated when the hostname of the desti-
nation computer cannot be resolved. This error usually occurs when you ping
an incorrect hostname, as shown in the following example, or try to use ping
with a hostname when hostname resolution (via DNS or a HOSTS text file) is
not configured:
C:\> ping www.comptia.ca
Unknown host www.comptia.ca

If the ping fails, you need to verify that the ping is being sent to the correct
remote host. If it is, and if name resolution is configured, you have to dig a lit-
tle more to find the problem. This error might indicate a problem with the
name resolution process, and you might need to verify that the DNS or WINS
server is available. Other commands, such as nslookup or dig, can help in this
process.

The Expired TTL Message
The Time to Live (TTL) is an important consideration in understanding the
ping command. The function of the TTL is to prevent circular routing, which
occurs when a ping request keeps looping through a series of hosts. The TTL
counts each hop along the way toward its destination device. Each time it counts
one hop, the hop is subtracted from the TTL. If the TTL reaches 0, it has
expired, and you get a message like the following:
Reply from 24.67.180.1: TTL expired in transit

If the TTL is exceeded with ping, you might have a routing problem on the net-
work. You can modify the TTL for ping on a Windows system by using the
ping -i command.
Troubleshooting with **ping**

Although **ping** does not completely isolate problems, you can use it to help identify where a problem lies. When troubleshooting with **ping**, follow these steps:

1. Ping the IP address of your local loopback using the command `ping 127.0.0.1`. If this command is successful, you know that the TCP/IP protocol suite is installed correctly on your system and is functioning. If you are unable to ping the local loopback adapter, TCP/IP might need to be reloaded or reconfigured on the machine you are using.

**EXAM ALERT**

The loopback is a special function within the TCP/IP protocol stack that is supplied for troubleshooting purposes. The Class A IP address 127.X.X.X is reserved for the loopback. Although convention dictates that you use 127.0.0.1, you can use any address in the 127.X.X.X range, except for the network number itself (127.0.0.0) and the broadcast address (127.255.255.255). You can also ping by using the default hostname for the local system, which is called localhost (for example, `ping localhost`).

2. Ping the assigned IP address of your local network interface card (NIC). If the ping is successful, you know that your NIC is functioning on the network and has TCP/IP correctly installed. If you are unable to ping the local NIC, TCP/IP might not be bound to the NIC correctly, or the NIC drivers might be improperly installed.

3. Ping the IP address of another known good system on your local network. By doing so, you can determine whether the computer you are using can see other computers on the network. If you can ping other devices on your local network, you have network connectivity.

   If you cannot ping other devices on your local network, but you could ping the IP address of your system, you might not be connected to the network correctly.

4. After you’ve confirmed that you have network connectivity for the local network, you can verify connectivity to a remote network by sending a ping to the IP address of the default gateway.

5. If you can ping the default gateway, you can verify remote connectivity by sending a ping to the IP address of a system on a remote network.
Using just the ping command in these steps, you can confirm network connectivity on not only the local network, but also on a remote network. The whole process requires as much time as it takes to enter the command, and you can do it all from a single location.

If you are an optimistic person, you can perform step 5 first. If that works, all the other steps will also work, saving you the need to test them. If your step 5 trial fails, you can go to step 1 and start the troubleshooting process from the beginning.

**NOTE**

All but one of the ping examples used in this section show the ping command using the IP address of the remote host. It is also possible to ping the Domain Name Service (DNS) name of the remote host (for example, ping www.comptia.org, ping server1); of course, you can do this only if your network uses a DNS server. On a Windows-based network, you can also ping by using the Network Basic Input/Output System (NetBIOS) computer name.

**ARP**

*Address Resolution Protocol (ARP)* is used to resolve IP addresses to MAC addresses. This is important because on a network, devices find each other using the IP address, but communication between devices requires the MAC address.

**EXAM ALERT**

For the Network+ exam, remember that the function of ARP is to resolve IP addresses to Layer 2 or MAC addresses.
Chapter 8: Network Performance Optimization

When a computer wants to send data to another computer on the network, it must know the MAC address (physical address) of the destination system. To discover this information, ARP sends out a discovery packet to obtain the MAC address. When the destination computer is found, it sends its MAC address to the sending computer. The ARP-resolved MAC addresses are stored temporarily on a computer system in the ARP cache. Inside this ARP cache is a list of matching MAC and IP addresses. This ARP cache is checked before a discovery packet is sent to the network to determine if there is an existing entry.

Entries in the ARP cache are periodically flushed so that the cache doesn’t fill up with unused entries. The following code shows an example of the `arp` command with the output from a Windows server system:

```
C:\>arp -a
Interface: 24.67.179.22 on Interface 0x3
          Internet Address      Physical Address      Type
  24.67.179.1           00-00-77-93-d8-3d     dynamic
```

As you might notice, the type is listed as dynamic. Entries in the ARP cache can be added statically or dynamically. Static entries are added manually and do not expire. The dynamic entries are added automatically when the system accesses another on the network.

As with other command-line utilities, several switches are available for the `arp` command. Table 8.7 shows the available switches for Windows-based systems.

### Table 8.7 `arp` Switches

<table>
<thead>
<tr>
<th>Switch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-a</code> or <code>-g</code></td>
<td>Displays both the IP and MAC addresses and whether they are dynamic or static entries.</td>
</tr>
<tr>
<td><code>-inet_addr</code></td>
<td>Specifies a specific Internet address.</td>
</tr>
<tr>
<td><code>-N if_addr</code></td>
<td>Displays the ARP entries for a specified network interface.</td>
</tr>
<tr>
<td><code>-eth_addr</code></td>
<td>Specifies a MAC address.</td>
</tr>
<tr>
<td><code>-if_addr</code></td>
<td>Specifies an Internet address.</td>
</tr>
<tr>
<td><code>-d</code></td>
<td>Deletes an entry from the ARP cache.</td>
</tr>
<tr>
<td><code>-s</code></td>
<td>Adds a static permanent address to the ARP cache.</td>
</tr>
</tbody>
</table>
arp ping

Earlier in this chapter we talked about the ping command and how it is used to test connectivity between devices on a network. Using the ping command is often an administrator’s first step to test connectivity between network devices. If the ping fails, it is assumed that the device you are pinging is offline. But this may not always be the case.

Most companies now use firewalls or other security measures that may block ICMP requests. This means that a ping request will not work. Blocking ICMP is a security measure; if a would-be hacker cannot hit the target, he may not attack the host.

If ICMP is blocked, you have still another option to test connectivity with a device on the network—the arp ping. As mentioned, the ARP utility is used to resolve IP addresses to MAC addresses. The arp ping utility does not use the ICMP protocol to test connectivity like ping does, rather it uses the ARP protocol. However, ARP is not routable, and the arp ping cannot be routed to work over separate networks. The arp ping works only on the local subnet.

Just like with a regular ping, an arp ping specifies an IP address; however, instead of returning regular ping results, the arp ping responds with the MAC address and name of the computer system. So, when a regular ping using ICMP fails to locate a system, the arp ping uses a different method to find the system. With arp ping, it is possible to ping a MAC address directly. From this, it is possible to determine if duplicate IP addresses are used and, as mentioned, determine if a system is responding.

arp ping is not built into Windows, but you can download a number of programs that allow you to ping using ARP. Linux, on the other hand, has an arp ping utility ready to use. Figure 8.5 shows the results of an arp ping from a shareware Windows utility.
The `netstat` Command

The `netstat` command displays the protocol statistics and current TCP/IP connections on the local system. Used without any switches, the `netstat` command shows the active connections for all outbound TCP/IP connections. In addition, several switches are available that change the type of information `netstat` displays. Table 8.8 shows the various switches available for the `netstat` utility.

<table>
<thead>
<tr>
<th>Switch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a</td>
<td>Displays the current connections and listening ports.</td>
</tr>
<tr>
<td>-e</td>
<td>Displays Ethernet statistics.</td>
</tr>
<tr>
<td>-n</td>
<td>Lists addresses and port numbers in numeric form.</td>
</tr>
<tr>
<td>-p</td>
<td>Shows connections for the specified protocol.</td>
</tr>
<tr>
<td>-r</td>
<td>Shows the routing table.</td>
</tr>
<tr>
<td>-s</td>
<td>Lists per-protocol statistics.</td>
</tr>
<tr>
<td>interval</td>
<td>Specifies how long to wait before redisplaying statistics.</td>
</tr>
</tbody>
</table>
The netstat utility is used to show the port activity for both TCP and UDP connections, showing the inbound and outbound connections. When used without switches, the netstat utility has four information headings.

- **Proto**: Lists the protocol being used, either UDP or TCP.
- **Local address**: Specifies the local address and port being used.
- **Foreign address**: Identifies the destination address and port being used.
- **State**: Specifies whether the connection is established.

In its default usage, the `netstat` command shows outbound connections that have been established by TCP. The following shows sample output from a `netstat` command without using any switches:

```plaintext
C:\> netstat
Active Connections
Proto  Local Address          Foreign Address        State
TCP    laptop:2848            MEDIASERVICES1:1755    ESTABLISHED
TCP    laptop:1833            www.dollarhost.com:80  ESTABLISHED
TCP    laptop:2858            194.70.58.241:80       ESTABLISHED
TCP    laptop:2860            194.70.58.241:80       ESTABLISHED
TCP    laptop:2354            www.dollarhost.com:80  ESTABLISHED
TCP    laptop:2361            www.dollarhost.com:80  ESTABLISHED
TCP    laptop:1114            www.dollarhost.com:80  ESTABLISHED
TCP    laptop:1959            www.dollarhost.com:80  ESTABLISHED
TCP    laptop:1960            www.dollarhost.com:80  ESTABLISHED
TCP    laptop:1963            www.dollarhost.com:80  ESTABLISHED
TCP    laptop:2870            localhost:8431         TIME_WAIT
TCP    laptop:8431            localhost:2862         TIME_WAIT
TCP    laptop:8431            localhost:2863         TIME_WAIT
TCP    laptop:8431            localhost:2867         TIME_WAIT
TCP    laptop:8431            localhost:2872         TIME_WAIT
```

As with any other command-line utility, the netstat utility has a number of switches. The following sections briefly explain the switches and give sample output from each.
netstat -e

The netstat -e command shows the activity for the NIC and displays the number of packets that have been both sent and received. Here’s an example:

C:\WINDOWS\Desktop>netstat -e
Interface Statistics

<table>
<thead>
<tr>
<th>Received</th>
<th>Sent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bytes</td>
<td>17412385</td>
</tr>
<tr>
<td>Unicast packets</td>
<td>79129</td>
</tr>
<tr>
<td>Non-unicast packets</td>
<td>693</td>
</tr>
<tr>
<td>Discards</td>
<td>0</td>
</tr>
<tr>
<td>Errors</td>
<td>0</td>
</tr>
<tr>
<td>Unknown protocols</td>
<td>306</td>
</tr>
</tbody>
</table>

As you can see, the netstat -e command shows more than just the packets that have been sent and received:

- **Bytes**: The number of bytes that the NIC has sent or received since the computer was turned on.
- **Unicast packets**: Packets sent and received directly by this interface.
- **Non-unicast packets**: Broadcast or multicast packets that the NIC picked up.
- **Discards**: The number of packets rejected by the NIC, perhaps because they were damaged.
- **Errors**: The errors that occurred during either the sending or receiving process. As you would expect, this column should be a low number. If it is not, this could indicate a problem with the NIC.
- **Unknown protocols**: The number of packets that the system could not recognize.

netstat -a

The netstat -a command displays statistics for both TCP and User Datagram Protocol (UDP). Here is an example of the netstat -a command:

C:\WINDOWS\Desktop>netstat -a
Active Connections

<table>
<thead>
<tr>
<th>Proto</th>
<th>Local Address</th>
<th>Foreign Address</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>laptop:1027</td>
<td>LAPTOP:0</td>
<td>LISTENING</td>
</tr>
<tr>
<td>TCP</td>
<td>laptop:1030</td>
<td>LAPTOP:0</td>
<td>LISTENING</td>
</tr>
</tbody>
</table>
As you can see, the output includes four columns, which show the protocol, the local address, the foreign address, and the port's state. The TCP connections show the local and foreign destination addresses and the connection's current state. UDP, however, is a little different. It does not list a state status because, as mentioned throughout this book, UDP is a connectionless protocol and does not establish connections. The following list briefly explains the information provided by the `netstat -a` command:

- **Proto**: The protocol used by the connection.
- **Local Address**: The IP address of the local computer system and the port number it is using. If the entry in the local address field is an asterisk (*), the port has not yet been established.
- **Foreign Address**: The IP address of a remote computer system and the associated port. When a port has not been established, as with the UDP connections, `*:*` appears in the column.
- **State**: The current state of the TCP connection. Possible states include established, listening, closed, and waiting.

**netstat -r**

The `netstat -r` command is often used to view a system's routing table. A system uses a routing table to determine routing information for TCP/IP traffic. The following is an example of the `netstat -r` command from a Windows Vista system:

```
C:\WINDOWS\Desktop>netstat -r
Route table
```
Chapter 8: Network Performance Optimization

Active Routes:

<table>
<thead>
<tr>
<th>Network Destination</th>
<th>Netmask</th>
<th>Gateway</th>
<th>Interface</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0.0</td>
<td>0.0.0.0</td>
<td>24.67.179.1</td>
<td>24.67.179.22</td>
<td>1</td>
</tr>
<tr>
<td>24.67.179.0</td>
<td>255.255.255.0</td>
<td>24.67.179.22</td>
<td>24.67.179.22</td>
<td>1</td>
</tr>
<tr>
<td>24.67.179.22</td>
<td>255.255.255.255</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>1</td>
</tr>
<tr>
<td>24.255.255.255</td>
<td>255.255.255.255</td>
<td>24.67.179.22</td>
<td>24.67.179.22</td>
<td>1</td>
</tr>
<tr>
<td>127.0.0.0</td>
<td>255.0.0.0</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>1</td>
</tr>
<tr>
<td>224.0.0.0</td>
<td>224.0.0.0</td>
<td>24.67.179.22</td>
<td>24.67.179.22</td>
<td>1</td>
</tr>
<tr>
<td>255.255.255.255</td>
<td>255.255.255.255</td>
<td>24.67.179.22</td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

Default Gateway: 24.67.179.1

Persistent Routes:

None

**NOTE**
The `netstat -r` command output shows the same information as the output from the `route print` command.

**netstat -s**
The `netstat -s` command displays a number of statistics related to the TCP/IP protocol suite. Understanding the purpose of every field in the output is beyond the scope of the Network+ exam, but for your reference, sample output from the `netstat -s` command is shown here:

```
C:\>netstat -s

IP Statistics

Packets Received = 389938
Received Header Errors = 0
Received Address Errors = 1876
Datagrams Forwarded = 498
Unknown Protocols Received = 0
Received Packets Discarded = 0
```
Received Packets Delivered = 387566
Output Requests = 397334
Routing Discards = 0
Discarded Output Packets = 0
Output Packet No Route = 916
Reassembly Required = 0
Reassembly Successful = 0
Reassembly Failures = 0
Datagrams Successfully Fragmented = 0
Datagrams Failing Fragmentation = 0
Fragments Created = 0

ICMP Statistics

<table>
<thead>
<tr>
<th>Type</th>
<th>Received</th>
<th>Sent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Messages</td>
<td>40641</td>
<td>41111</td>
</tr>
<tr>
<td>Errors</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Destination Unreachable</td>
<td>223</td>
<td>680</td>
</tr>
<tr>
<td>Time Exceeded</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>Parameter Problems</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Source Quenches</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Redirects</td>
<td>0</td>
<td>38</td>
</tr>
<tr>
<td>Echos</td>
<td>20245</td>
<td>20148</td>
</tr>
<tr>
<td>Echo Replies</td>
<td>20149</td>
<td>20245</td>
</tr>
<tr>
<td>Timestamps</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Timestamp Replies</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Address Masks</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Address Mask Replies</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

TCP Statistics

<table>
<thead>
<tr>
<th>Type</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Opens</td>
<td>13538</td>
</tr>
<tr>
<td>Passive Opens</td>
<td>23132</td>
</tr>
<tr>
<td>Failed Connection Attempts</td>
<td>9259</td>
</tr>
<tr>
<td>Reset Connections</td>
<td>254</td>
</tr>
<tr>
<td>Current Connections</td>
<td>15</td>
</tr>
<tr>
<td>Segments Received</td>
<td>330242</td>
</tr>
<tr>
<td>Segments Sent</td>
<td>326935</td>
</tr>
<tr>
<td>Segments Retransmitted</td>
<td>18851</td>
</tr>
</tbody>
</table>

UDP Statistics

<table>
<thead>
<tr>
<th>Type</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Datagrams Received</td>
<td>20402</td>
</tr>
<tr>
<td>No Ports</td>
<td>20694</td>
</tr>
<tr>
<td>Receive Errors</td>
<td>0</td>
</tr>
<tr>
<td>Datagrams Sent</td>
<td>10217</td>
</tr>
</tbody>
</table>
nbtstat

The nbtstat utility is used to view protocol statistics and information for NetBIOS over TCP/IP connections. nbtstat is commonly used to troubleshoot NetBIOS name resolution problems. Because nbtstat resolves NetBIOS names, it's available only on Windows systems.

A number of case-sensitive switches are available for the nbtstat command, as shown in Table 8.9.

<table>
<thead>
<tr>
<th>Switch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nbtstat -a</td>
<td>(Adapter status) Outputs the NetBIOS name table and MAC addresses of the card for the specified computer.</td>
</tr>
<tr>
<td>nbtstat -A (IP address)</td>
<td>(Adapter status) Lists the remote machine's name table given its IP address.</td>
</tr>
<tr>
<td>nbtstat -c (cache)</td>
<td>Lists the contents of the NetBIOS name cache.</td>
</tr>
<tr>
<td>nbtstat -n (names)</td>
<td>Lists local NetBIOS names.</td>
</tr>
<tr>
<td>nbtstat -r (resolved)</td>
<td>Lists names resolved by broadcast or WINS.</td>
</tr>
<tr>
<td>nbtstat -R (Reload)</td>
<td>Purges and reloads the remote cache name table.</td>
</tr>
<tr>
<td>nbtstat -S (Sessions)</td>
<td>Summarizes the current NetBIOS sessions and their status.</td>
</tr>
<tr>
<td>nbtstat -s (sessions)</td>
<td>Lists the sessions table, converting destination IP addresses into computer NetBIOS names.</td>
</tr>
<tr>
<td>nbtstat -RR (ReleaseRefresh)</td>
<td>Sends Name Release packets to WINS and then starts Refresh.</td>
</tr>
<tr>
<td>nbtstat RemoteName</td>
<td>Remote host machine name.</td>
</tr>
<tr>
<td>nbtstat IP address</td>
<td>Dotted-decimal representation of the IP address.</td>
</tr>
<tr>
<td>nbtstat interval</td>
<td>Redisplays selected statistics, pausing \textit{interval/seconds} between each display. Press Ctrl+C to stop redisplaying statistics.</td>
</tr>
</tbody>
</table>

For example, the following is the output from the nbtstat -n command:

```
C:\>nbtstat -n
Lana # 0:
Node IpAddress: [169.254.196.192] Scope Id: []

NetBIOS Local Name Table
```
The ipconfig Command

The ipconfig command is a technician’s best friend when it comes to viewing the TCP/IP configuration of a Windows system. Used on its own, the ipconfig command shows basic information such as the name of the local network interface, the IP address, the subnet mask, and the default gateway. Combined with the /all switch, it shows a detailed set of information, as shown in the following example:

C:\>ipconfig /all
Windows 2003 IP Configuration
   Host Name . . . . . . . . . . . . : server
   Primary DNS Suffix . . . . . . : write
   Node Type . . . . . . . . . . . . : Broadcast
   IP Routing Enabled . . . . . . : Yes
   WINS Proxy Enabled. . . . . . : No
   DNS Suffix Search List. . . . . : write
                                 ok.anyotherhost.net
Ethernet adapter Local Area Connection:
   Connection-specific DNS Suffix . : ok.anyotherhost.net
   Description . . . . . . . . . . . : D-Link DFE-530TX PCI Fast Ethernet
   Physical Address . . . . . . . . : 00-80-C8-E3-4C-BD
   DHCP Enabled. . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . : Yes
   IP Address . . . . . . . . . . . : 24.67.184.65
   Subnet Mask . . . . . . . . . . : 255.255.254.0
   Default Gateway . . . . . . . . : 24.67.184.1
   DHCP Server . . . . . . . . . . : 24.67.253.195
   DNS Servers . . . . . . . . . . : 24.67.253.195
                                 24.67.253.212
   Lease Obtained.. . . . : Thursday, February 07, 2008 3:42:00 AM
   Lease Expires .. . . . : Saturday, February 09, 2008 3:42:00 AM

As you can imagine, you can use the output from the ipconfig /all command in a massive range of troubleshooting scenarios. Table 8.10 lists some of the most common troubleshooting symptoms, along with where to look for clues about solving them in the ipconfig /all output.
Chapter 8: Network Performance Optimization

### Table 8.10 Common Troubleshooting Symptoms That `ipconfig` Can Help Solve

<table>
<thead>
<tr>
<th>Symptom</th>
<th>Field to Check in the Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>The user is unable to connect to any other system.</td>
<td>Make sure that the TCP/IP address and subnet mask are correct. If the network uses DHCP, make sure that DHCP is enabled.</td>
</tr>
<tr>
<td>The user can connect to another system on the same subnet but cannot connect to a remote system.</td>
<td>Make sure the default gateway is configured correctly.</td>
</tr>
<tr>
<td>The user is unable to browse the Internet.</td>
<td>Make sure the DNS server parameters are configured correctly.</td>
</tr>
<tr>
<td>The user is unable to browse across remote subnets.</td>
<td>Make sure the WINS or DNS server parameters are configured correctly, if applicable.</td>
</tr>
</tbody>
</table>

### TIP

When looking at `ipconfig` information, you should be sure that all information is present and correct. For example, a missing or incorrect default gateway parameter limits communication to the local segment.

### EXAM ALERT

You should be prepared to identify the output from an `ipconfig` command in relationship to a troubleshooting scenario for the Network+ exam.

Using the `/all` switch might be far and away the most popular, but there are a few others. These include the switches listed in Table 8.11.

### EXAM ALERT

`ipconfig` and its associated switches are widely used by network administrators and therefore should be expected to make an appearance on the exam.
Table 8.11  ipconfig Switches

<table>
<thead>
<tr>
<th>Switch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Displays the ipconfig help screen.</td>
</tr>
<tr>
<td>/all</td>
<td>Displays additional IP configuration information.</td>
</tr>
<tr>
<td>/release</td>
<td>Releases the IPv4 address of the specified adapter.</td>
</tr>
<tr>
<td>/release6</td>
<td>Releases the IPv6 address of the specified adapter.</td>
</tr>
<tr>
<td>/renew</td>
<td>Renews the IPv4 address of a specified adapter.</td>
</tr>
<tr>
<td>/renew6</td>
<td>Renews the IPv6 address of a specified adapter.</td>
</tr>
<tr>
<td>/flushdns</td>
<td>Purges the DNS cache.</td>
</tr>
<tr>
<td>/registerdns</td>
<td>Refreshes the DHCP lease and reregisters the DNS names.</td>
</tr>
<tr>
<td>/displaydns</td>
<td>Used to display the information in the DNS cache.</td>
</tr>
</tbody>
</table>

The ipconfig /release and ipconfig /renew commands work only when your system is using DHCP.

EXAM ALERT

The ipconfig command on the Windows XP/Vista and Windows Server operating systems provides additional switches and functionality geared toward Active Directory and Dynamic DNS. You do not need to be concerned with these switches for the Network+ exam, but you can view information on them by using the ipconfig /? command.

ifconfig

Ifconfig performs the same function as ipconfig, but on a Linux, UNIX, or Macintosh system. Because Linux relies more heavily on command-line utilities than Windows, the Linux and UNIX version of ifconfig provides much more functionality than ipconfig. On a Linux or UNIX system, you can get information about the usage of the ifconfig command by using ifconfig --help. The following output provides an example of the basic ifconfig command run on a Linux system:

```
eth0  Link encap:Ethernet  HWaddr 00:60:08:17:63:A0
     UP BROADCAST RUNNING MTU:1500  Metric:1
     RX packets:911 errors:0 dropped:0 overruns:0 frame:0
     TX packets:804 errors:0 dropped:0 overruns:0 carrier:0
```
Although the `ifconfig` command displays the IP address, subnet mask, and default gateway information for both the installed network adapter and the local loopback adapter, it does not report DHCP lease information. Instead, you can use the `pump -s` command to view detailed information on the DHCP lease, including the assigned IP address, the address of the DHCP server, and the time remaining on the lease. The `pump` command can also be used to release and renew IP addresses assigned via DHCP and to view DNS server information.

**nslookup**

`nslookup` is a utility used to troubleshoot DNS-related problems. Using `nslookup`, you can, for example, run manual name resolution queries against DNS servers, get information about your system’s DNS configuration, or specify what kind of DNS record should be resolved.

When `nslookup` is started, it displays the current hostname and the IP address of the locally configured DNS server. You then see a command prompt that allows you to specify further queries. This is known as interactive mode. Table 8.12 lists the commands you can enter in interactive mode.

**Table 8.12 nslookup Switches**

<table>
<thead>
<tr>
<th>Switch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>Prints options, as well as current server and host information.</td>
</tr>
<tr>
<td>[no]d2</td>
<td>Prints exhaustive debugging information.</td>
</tr>
<tr>
<td>[no]defname</td>
<td>Appends the domain name to each query.</td>
</tr>
<tr>
<td>[no]recurse</td>
<td>Asks for a recursive answer to the query.</td>
</tr>
<tr>
<td>[no]search</td>
<td>Uses the domain search list.</td>
</tr>
<tr>
<td>[no]vc</td>
<td>Always uses a virtual circuit.</td>
</tr>
<tr>
<td>domain=NAME</td>
<td>Sets the default domain name to <em>NAME</em>.</td>
</tr>
</tbody>
</table>
Table 8.12  **nslookup Switches**  Continued

<table>
<thead>
<tr>
<th>Switch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>srchlist=N1[/N2/.../N6]</td>
<td>Sets the domain to N1 and the search list to N1, N2, and so on.</td>
</tr>
<tr>
<td>root=NAME</td>
<td>Sets the root server to NAME.</td>
</tr>
<tr>
<td>retry=X</td>
<td>Sets the number of retries to X.</td>
</tr>
<tr>
<td>timeout=X</td>
<td>Sets the initial timeout interval to X seconds.</td>
</tr>
<tr>
<td>type=X</td>
<td>Sets the query type (for example, A, ANY, CNAME, MX, NS, PTR, SOA, or SRV).</td>
</tr>
<tr>
<td>querytype=X</td>
<td>Same as type.</td>
</tr>
<tr>
<td>class=X</td>
<td>Sets the query class (for example, IN [Internet], ANY).</td>
</tr>
<tr>
<td>[no]msxfr</td>
<td>Uses Microsoft fast zone transfer.</td>
</tr>
<tr>
<td>ixfrver=X</td>
<td>The current version to use in an IXFR transfer request.</td>
</tr>
<tr>
<td>server NAME</td>
<td>Sets the default server to NAME using the current default server.</td>
</tr>
<tr>
<td>exit</td>
<td>Exits the program.</td>
</tr>
</tbody>
</table>

Instead of using interactive mode, you can also execute nslookup requests directly at the command prompt. The following listing shows the output from the nslookup command when a domain name is specified to be resolved:

```
C:\> nslookup comptia.org
Server:  nsc1.ht.ok.shawcable.net
Address:  64.59.168.13

Non-authoritative answer:
Name:  comptia.org
Address:  208.252.144.4
```

As you can see from the output, nslookup shows the hostname and IP address of the DNS server against which the resolution was performed, along with the hostname and IP address of the resolved host.

**dig**

dig is used on a Linux, UNIX, or Macintosh system to perform manual DNS lookups. dig performs the same basic task as nslookup, but with one major distinction: The dig command does not have an interactive mode and instead uses only command-line switches to customize results.

dig generally is considered a more powerful tool than nslookup, but in the course of a typical network administrator's day, the minor limitations of nslookup are unlikely to be too much of a factor. Instead, dig is often simply the
tool of choice for DNS information and troubleshooting on UNIX, Linux, or Macintosh systems. Like nslookup, dig can be used to perform simple name resolution requests. The output from this process is shown in the following listing:

; <<>> DiG 8.2 <<>> examcram.com
;; res options: init recurs defnam dnsrc
;; got answer:
;; >>>HEADER<<: opcode: QUERY, status: NOERROR, id: 4
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 0
;; QUERY SECTION:
;; examcram.com, type = A, class = IN

;; ANSWER SECTION:
examcram.com 7h33m IN A 63.240.93.157

;; AUTHORITY SECTION:
examcram.com 7h33m IN NS usrxdns1.pearsonlive.co.uk.
examcram.com 7h33m IN NS oldtxdns2.pearsonlive.co.uk.

;; Total query time: 78 msec
;; FROM: localhost.localdomain to SERVER: default -- 209.53.4.130
;; WHEN: Sat Oct 16 20:21:24 2004
;; MSG SIZE  sent: 30  rcvd: 103

As you can see, dig provides a number of pieces of information in the basic output—more so than nslookup. Network administrators can gain information from three key areas of the output—ANSWER SECTION, AUTHORITY SECTION, and the last four lines of the output.

The ANSWER SECTION of the output provides the name of the domain or host being resolved, along with its IP address. The A in the results line indicates the record type that is being resolved.

The AUTHORITY SECTION provides information on the authoritative DNS servers for the domain against which the resolution request was performed. This information can be useful in determining whether the correct DNS servers are considered authoritative for a domain.

The last four lines of the output show how long the name resolution request took to process and the IP address of the DNS server that performed the resolution. It also shows the date and time of the request, as well as the size of the packets sent and received.
mtr
The My traceroute (mtr) command is used on Linux/UNIX systems. Essentially it combines the functionality of ping with that of traceroute. After it is issued, the command provides details of the path between two hosts (similar to the traceroute command) and additionally statistics for each node in the path based on samples taken over a period of time (similar to the ping command).

In a Windows environment, a similar command is the pathping command. Pathping has been available since Windows NT and is used with modern Windows versions, including Windows 2000, 2003, XP, and Vista.

The host Command
The host command is used on Linux/UNIX systems to perform a reverse lookup on an IP address. A reverse lookup involves looking up an IP address and resolving the hostname from that. Users running Microsoft Windows operating systems could use the nslookup command to perform a reverse lookup on an IP address, and Linux users can use the host command or nslookup.

Here's an example of the host command:

```
host 24.67.108.119
```

This command returns the hostname associated with the IP address 24.67.108.119.

The route Utility
The route utility is an often-used and very handy tool. With the route command, you display and modify the routing table on your Windows and Linux systems. Figure 8.6 shows the output from a route print command on a Windows Vista system.
As well as displaying the routing table, the Windows version of the route command has a number of other switches, as detailed in Table 8.13. For complete information about all the switches available with the route command on a Windows system, type route at the command line. To see a list of the route command switches on a Linux system, use the command route -help.

Table 8.13  Switches for the route Command in Windows

<table>
<thead>
<tr>
<th>Switch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Allows you to add a static route to the routing table.</td>
</tr>
<tr>
<td>delete</td>
<td>Allows you to remove a route from the routing table.</td>
</tr>
<tr>
<td>change</td>
<td>Allows you to modify an existing route.</td>
</tr>
<tr>
<td>-p</td>
<td>When used with the add command, makes the route permanent. If the -p switch is not used when a route is added, the route is lost upon reboot.</td>
</tr>
<tr>
<td>print</td>
<td>Allows you to view the system’s routing table.</td>
</tr>
<tr>
<td>-f</td>
<td>Removes all gateway entries from the routing table.</td>
</tr>
</tbody>
</table>
Review and Test Yourself

The following sections provide you with the opportunity to review what you’ve learned in this chapter and to test yourself.

The Facts

For the exam, don’t forget these important concepts:

- RAID 0 uses disk striping over two or more disks but offers no fault tolerance.
- RAID 1 uses two disks in a mirrored configuration.
- Disk duplexing is a RAID 1 implementation using separate hard disk controllers.
- RAID 5 is disk striping with parity, requiring three disks at a minimum.
- With a full backup, all data is backed up, and data can be restored from a single tape set. Full backups do not use the archive bit but clear it after files have been copied to tape.
- With incremental backups, all data that has changed since the last full or incremental backup is backed up. The restore procedure requires several tapes: the latest full backup and all incremental tapes since the last full backup. An incremental backup uses the archive bit and clears it after a file is saved to disk.
- With a differential backup, all data changed since the last full backup is backed up. The restore procedure requires the latest full backup tape and the latest differential backup tape. A differential backup uses the archive bit to determine which files need to be backed up but does not clear it.
- You should use an offsite tape rotation scheme to store current copies of backups in a secure offsite location. A commonly used rotation is the grandfather, father, son (GFS) rotation.
- You should periodically introduce new tapes into the tape rotation and destroy the old ones.
- Two key strategies are commonly employed for server fault tolerance: standby servers and server clustering.
- Hot, warm, and cold spares are designed to replace failed system components.
- Hot, warm, and cold sites are designed to provide alternative locations for network operations in the event of a disaster.
Chapter 8: Network Performance Optimization

- ping is a command-line utility designed to test connectivity between systems on a TCP/IP-based network.
- Ping the IP address of your local loopback using the command `ping 127.0.0.1`. If this command is successful, you know that the TCP/IP protocol suite is installed correctly on your system and is functioning.
- If you cannot ping other devices on your local network, but you can ping your local NIC, you might not be connected to the network correctly, or the computer might have a cable problem.
- Trace route is a TCP/IP utility that is used to track the path a packet takes to reach a remote host and to isolate where network problems may be. Trace route functionality can be used on Windows platforms with the `tracert` command and on Linux, UNIX, and Macintosh platforms with the `traceroute` command.
- Trace route reports how long it takes to reach each host in the path. It is a useful tool for isolating bottlenecks in a network.
- arp is the part of the TCP/IP suite whose function is to resolve IP addresses to MAC addresses.
- arp operates at the network layer of the Open Systems Interconnect (OSI) model.
- netstat is used to view both inbound and outbound TCP/IP network connections.
- The `netstat -r` command can be used to display the system’s routing table.
- nbtstat is used to display protocol and statistical information for NetBIOS over TCP/IP connections.
- The `ipconfig` command shows the IP configuration information for all NICs installed in a system.
- The `ipconfig /all` command is used to display detailed TCP/IP configuration information.
- The `ipconfig /renew` command is used to renew DHCP assigned IP address configurations.
- When looking for client connectivity problems using `ipconfig`, ensure that the gateway is correctly set.
- The `ifconfig` command is the Linux, Macintosh, and UNIX equivalent of the `ipconfig` command.
- The `nslookup` command is a TCP/IP diagnostic tool that is used to troubleshoot DNS problems. On Linux, UNIX, and Macintosh systems, you can also use the `dig` command for the same purpose.
Key Terms

- RAID
- Disk mirroring
- Disk duplexing
- Disk striping
- Full backup
- Incremental backup
- Differential backup
- Hot site
- Cold site
- Warm site
- Hot spare
- Cold spare
- tracert
- ping
- arp
- netstat
- nbtstat
- ipconfig
- ifconfig
- nslookup
- dig
- mtr
- host
- arp ping
- route