

Realtime
publishers

"Leading the Conversation"

The Shortcut Guide[™] To



**Network
Management
for the Mid-Market**

sponsored by



Greg Shields

Chapter 4: Network Troubleshooting and Diagnostics.....	59
Developing Good Troubleshooting Technique.....	59
OSI as a Troubleshooting Framework	60
Three Different Approaches	62
Tool Suites for Identifying the Problem	64
Telnet and SSH	64
Serial Port Tools	65
Network Monitoring	65
Network Discovery	66
Attack Identification and Simulation	67
SNMP Trapping	67
Ping, Traceroute, and ARP	68
MIB Browsers	68
IP Address Management.....	69
Subnet and Address Calculations.....	69
DHCP	69
IP Address Management Tools.....	70
Network Engineering Applications.....	71
Protocol Analyzers.....	71
Traffic Generators.....	73
Network Simulation Tools.....	74
Troubleshooting Involves Good Technique and Good Tools.....	74

Copyright Statement

© 2007 Realtimedpublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimedpublishers.com, Inc. (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtimedpublishers.com, Inc or its web site sponsors. In no event shall Realtimedpublishers.com, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtimedpublishers.com and the Realtimedpublishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtimedpublishers.com, please contact us via e-mail at info@realtimedpublishers.com.

[**Editor's Note:** This eBook was downloaded from Realtime Nexus—The Digital Library. All leading technology guides from Realtimepublishers can be found at <http://nexus.realtimepublishers.com>.]

Chapter 4: Network Troubleshooting and Diagnostics

The most difficult part of any troubleshooting process is often just learning that there is a problem. Throughout, this guide has discussed how an effective NMS can keep you informed about the status and health of your network. We've discussed how an NMS can inform you when a network fault occurs or when performance suffers. We talked about how that same NMS can assist with maintaining a stable and consistent configuration for the devices on the network. Utilization of an effective NMS with administrator notification goes far toward resolving this difficult part of troubleshooting—knowing if there even is a problem.

Next up in difficulty is finding out what that problem really is. It isn't a stretch to say that the same NMS that alerts you when a problem occurs can assist with problem identification. But sometimes the event or condition that triggers the alarm isn't always the root cause of the problem. If you receive an alert that a network link isn't meeting its performance SLA, you don't always immediately know what is causing performance to drop.

Once you know the root cause of the problem, the resolution is usually a Google search away. But finding that root cause can consume the vast majority of the time involved with fixing the problem. It is this process of network troubleshooting where network administrators truly earn our keep. The ability to quickly and efficiently perform troubleshooting when a problem occurs separates the veteran administrators from the green ones. No matter what your experience level with troubleshooting, maintaining a good tool suite along with a good technique is critically important. This last chapter will dig into both to help you become a better troubleshooter.

Developing Good Troubleshooting Technique

When a problem occurs on your network, what do you usually do first? Do you ping the network device to see whether it is up and responding? Do you dive into the network closet to determine whether the LED lights are still blinking green or switched to red? Do you contact the user who called or log into the NMS that notified to find out more about the problem?

None of these initial triage maneuvers are any better than any other. For those who ping the device first, you immediately get a low-level understanding of whether that device can communicate with your workstation. For those that dive into the network closet, you can quickly find out if the interface is showing errors or is down. For those that contact the user or work with the notifying agent, you get an immediate first-hand look into the error that kicked up the notification.

In each situation, the most important part of troubleshooting is developing a good technique. No matter what the problem is on your network, you'll find that having a good technique for finding that problem helps you quickly identify where the root cause exists so that you can work towards a solution.



It's worth stating that a good troubleshooter is a fast troubleshooter. Throughout this chapter, you'll notice that the troubleshooting processes, as well as the tools discussed, are designed to assist you with quickly coming to a resolution.

OSI as a Troubleshooting Framework

One commonly used framework for troubleshooting that helps structure your response to a known network problem is the International Standards Organization (ISO) Open Systems Interconnect (OSI) model. If you've worked with networking devices for any period of time, you are likely already familiar with OSI. It's the framework that encapsulates much of modern networking, and most network protocols live somewhere within its seven layers. Where you may not have used it before is as a troubleshooting guide for triaging an unknown problem on the network.

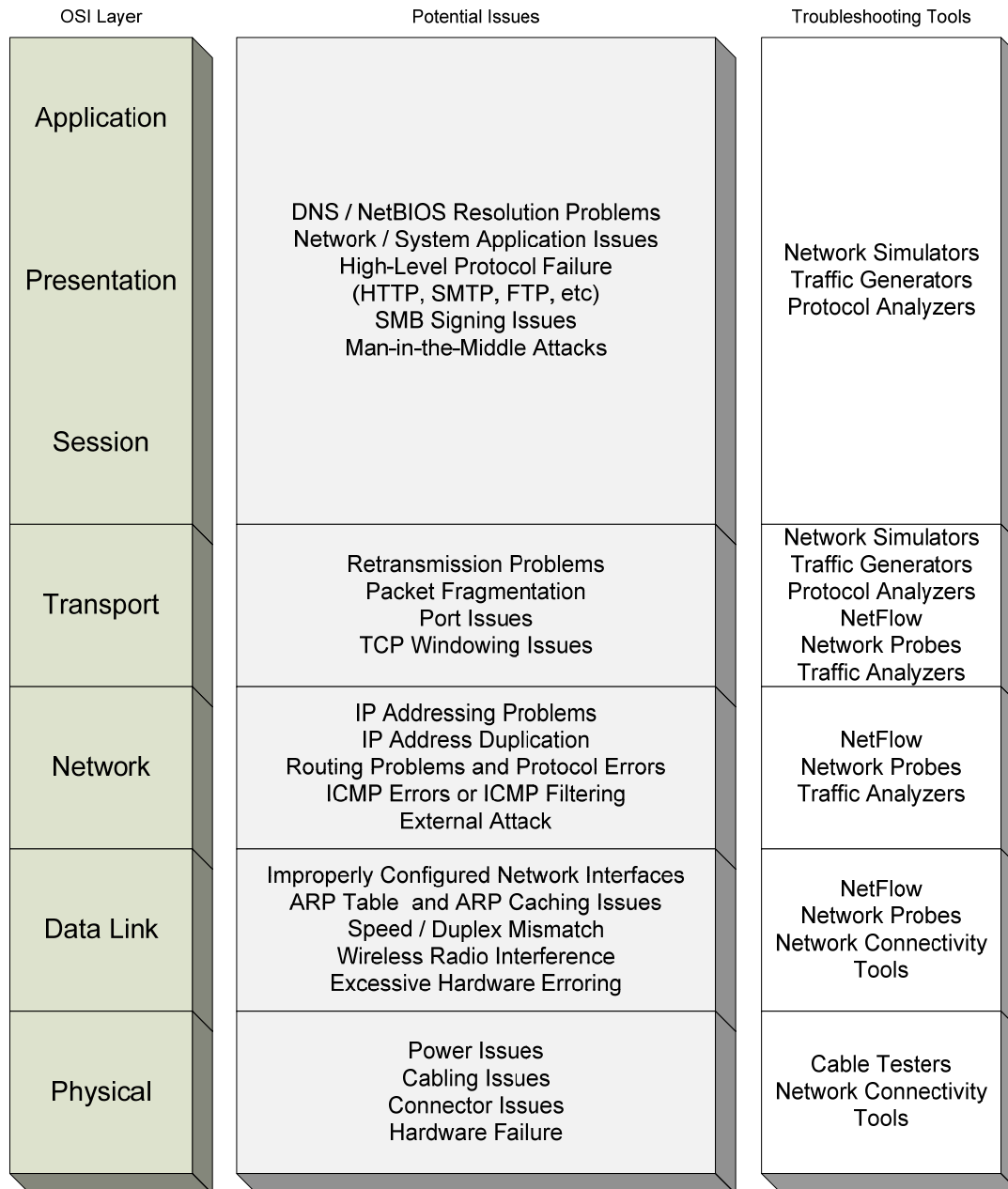



Figure 4.1: The OSI model is an excellent mental framework to assist the troubleshooter with identifying network problems.

Without going into too much detail on the history and use of model, let's take a look at how you can extend the OSI model into a framework for problem isolation. Figure 4.1 shows the seven layers in the OSI model and some issues that typically occur related to each layer. Let's discuss each of the layers in-turn from the bottom-up:

- *At the Physical layer*, problems typically involve some break in the physical connectivity that makes up the network. Broken network connections, cabling and connector issues, and hardware problems that inhibit the movement of electricity from device to device typically indicate a problem at this layer.
- *At the Data Link layer*, we move away from purely electrical problems and into the configuration of the interface itself. Data Link problems often have to do with Address Resolution Protocol (ARP) problems in relating IP addresses to Media Access Control (MAC) addresses. These can be caused by speed and duplex mismatching between network devices or excessive hardware errors for the interface. An incorrectly configured interface within the device operating system (OS) or interference for wireless connections can also cause problems at the Data Link layer.
- *At the Network layer*, we begin experiencing problems with network traversal. Network layer problems typically occur when network packets cannot make their way from source to destination. This may have something to do with incorrect IP addressing or duplicate IP addresses on the network. Problems with routing data or ICMP packets across the network or protocol errors can also cause problems here. In extreme cases, an external attack can also spike error levels on network devices and cause problems identified at the Network Layer.
- *At the Transport layer*, we isolate problems that typically occur with TCP or UDP packets in Ethernet networks. These may have to do with excessive retransmission errors or packet fragmentation. Either of these problems can cause network performance to suffer or drop completely. Problems at this layer can be difficult to track down because unlike the lower layers they often don't involve a complete loss of connectivity. Additionally, Transport layer problems can often involve the blocking of traffic at the individual IP port layer. If you've ever been able to ping a server but cannot connect via a known port, this can be a Transport layer problem.
- *The Session, Presentation, and Application layers* are often lumped together because more recent interpretations of the OSI model tend to grey the lines between these three layers. The troubleshooting process for these three layers involves problems that have to do with applications that rely on the network.

These applications could involve DNS, NetBIOS, or other resolution, application issues on residing OSs, or high-level protocol failures or misconfigurations. Examples of these high-level protocols are HTTP, SMTP, FTP, and other protocols that typically "use the network" rather than "run the network." Additionally, specialized external attacks such as "man-in-the-middle" attacks can occur at these levels.

Network problems can and do occur at any level in the model. And because the model is so highly understood by network administrators, it immediately becomes a good measuring stick to assist with communicating those problems between triaging administrators. If you've ever worked with another administrator who uses language like, "This looks like a Layer 4 problem," you can immediately understand the general area (the Transport layer) in which the problem may be occurring.

 You'll hear seasoned network administrators often refer to problems by their layer number. For example, when you hear "that's at layer 3," it can mean an IP connectivity problem. Layer 4 can reveal the problem is due to a network port closure. Network administrators jokingly refer to problems that occur with a system and not part of their network as those "at layer 7."

Let's talk about three different ways in which you can progress through this model during a typical problem isolation activity.

Three Different Approaches

Network administrators who use OSI as a troubleshooting framework typically navigate the model in one of three ways: Bottom-Up, Top-Down, and Divide-and-Conquer. Depending on how the problem manifests and their experience level, they may choose one method over another for that particular problem. Each of these approaches has its utility based on the type of problem that is occurring. Let's look at each.

Bottom-Up

The Bottom-Up approach simply means that administrators start at the bottom of the OSI model and work their way up through the various levels as they strike off potential root causes that are not causing the problem. An administrator using the Bottom-Up approach will typically start by looking at the physical layer issues, determine whether a break in network connectivity has occurred, and then work up through network interface configurations and error rates, and continue through IP and TCP/UDP errors such as routing, fragmentation, and blocked ports before looking at the individual applications experiencing the problem.

This approach works best in situations in which the network is fully down or experiencing numerous low-level errors. It also works best when the problem is particularly complex. In complex problems, the faulting application often does not provide enough debugging data to the administrator to give insight as to the problem. Thus, a network-focused approach works best.

Top-Down

The Top-Down approach is the reverse of the Bottom-Up approach in that the administrator starts at the top of the OSI model first, looking at the faulted application and attempting to track down why that application is faulted. This model works best when the network is in a known-good state and a new application or application reconfiguration is being completed on the network. The administrator can start by ensuring the application is properly configured, then work downward to ensure that full IP connectivity and appropriate ports are open for proper functionality of the application. Once all upper-level issues are resolved, a back-check on the network can be done to validate its proper functionality. As said earlier, this approach is typically used when the network itself is believed to be functioning correctly but a new network application is being introduced or an existing one is being reconfigured or repurposed.

Divide-and-Conquer

The Divide-and-Conquer approach is a fancy name for the “gut feeling” approach. It is typically used by seasoned administrators who have a good internal understanding of the network and the problems it can face. The Divide-and-Conquer approach involves an innate feeling for where the problem may occur, starting with that layer of the OSI model first, and working out from that location. This approach can also be used for trivial issues that the administrator has seen before.

However, this approach has the downfall of often being non-scientific enough to properly diagnose a difficult problem. If the problem is complex in nature, the Divide-and-Conquer approach may not be structured enough to track down the issue.

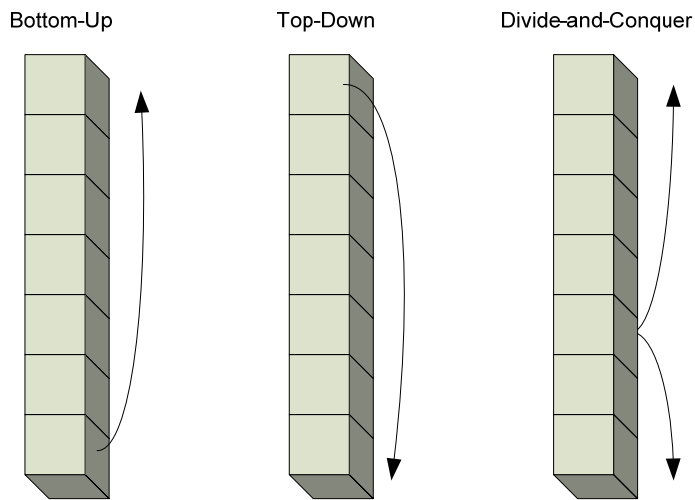



Figure 4.2: Depending on the type of problem, a Bottom-Up, Top-Down, or Divide-and-Conquer approach may be best for isolating the root cause of the problem.

No matter which approach you use, until you begin to develop that “gut instinct” for your network and its unique characteristics, you should consider a structured method for your troubleshooting technique. Although utilizing a structured method can increase the time needed to resolve the problem, it will track down the problem without missing key items that drive resolution “band-aiding.”

 Until you develop a solid foundation in troubleshooting, be cautious with the Divide-and-Conquer approach. This approach often treats the symptoms of the network problem without actually fixing the root cause.

Let’s move away from the general concepts of troubleshooting and talk now about some of the tools available for helping move through the layers.

Tool Suites for Identifying the Problem

A baker can't bake a cake without a good cake pan, and a mechanic won't get very far in fixing your car without a solid wrench set. Though the tools for network administrators are different and more difficult to wrap your hands around than these examples, the need for them is no different. Lacking the appropriate set of tools will usually prevent the job from getting done. This section will discuss some of the tools and their feature sets that you should add to your quiver to support your technical troubleshooting activities.


Telnet and SSH

Telnet, originally short for “TELEtype NETwork” and now considered a proper name all to itself, is the most common mechanism for forwarding a system's command-line console session to a remote host. Telnet is entirely textual and command-line driven, which makes its use difficult for newer administrators. Telnet is used by virtually all UNIX hosts as well as network devices for device configuration and administration.

SSH or “Secure SHell” is a similar protocol intended to accomplish the same goal as Telnet but with an element of built-in security. SSH uses public-key cryptography to authenticate a user to the system as well as provide confidentiality and integrity of data passing between the SSH client and server. SSH is quickly becoming the standard for remote terminal applications due to this added security built-in to its protocol.

For either protocol, the necessary tool in your troubleshooting quiver will be a Telnet or SSH client. Numerous clients exist, and some have more features than others. Some features you may want to consider when looking for a good Telnet or SSH client are:

- Text colorization
- Function key mapping
- Remote file copying support
- Server connection profiling
- Alarm generation
- Script recording and playback
- Session tabbing
- Secure password caching

 As a rule, always try to use SSH over Telnet when it is supported by your network devices. Telnet sends data and passwords across the network in clear-text, which allows an attacker to easily sniff the traffic as it traverses the network. This is especially true when connecting to devices across the Internet.

Serial Port Tools

Although a good Telnet or SSH client will help you connect to already-configured network devices, these devices often must be initially configured using an on-board serial port before they can connect to a network. The on-board serial port includes a cable transceiver that converts the network device's serial port to one that is useable by a desktop or laptop system. To connect the desktop or laptop system to the network device, a serial port tool is needed.

Like Telnet/SSH clients, serial port tools come in many flavors. As an example, one very basic serial port tool, HyperTerminal, has been available with Microsoft Windows systems from the time of Windows 95 up until the release of Windows Vista. However, because network administrators make substantial use of these tools in network setup and troubleshooting, there are additional feature sets above those in the native tools that are necessary to ease administration. Some features you may want in a good serial port tool are:

- Rich copy-and-paste
- Multiple terminal emulation support
- Printing and print selection
- Automation and scripting
- Text-to-file exporting
- Extended serial support conversion

Network Monitoring

Network monitoring tools can either be a component of your NMS or a separate utility. In either case, a network monitoring tool is used to record and analyze the characteristics within its configured network. Network monitoring tools can monitor for network performance as well as network outage and device resource use. They typically aggregate multiple network devices into a single user interface for cross-device analysis. Some features in a network monitoring tool that are critical for the troubleshooting process are:

- Multiple device capability
- Traffic graphing support
- Device resource use monitoring
- Alerting and notification via multiple mediums
- SMS/text messaging support
- SNMP management
- Traffic analysis
- Built-in traffic filters and aggregators

Network Discovery

Knowing what is going on within your network is only useful if you're aware of all the devices that make up that network. If a problem on the network occurs because of a rogue device, it is often difficult to track down that device without a tool to do the tracking. Network discovery tools are those that scan the network for known device heuristics. When a device heuristic is found at a particular address, the network discovery tool logs the location and its believed device type, then reports that information to the administrator.

Numerous network discovery tools exist and each has a specific mechanism for seeking out devices—by IP address, MAC address, SNMP response, DNS entry, or even individual switch port on switching devices. Some features useful in a network discovery tool are:

- NMS integration
- Multiple IP range entry
- Fast scanning
- Device heuristic databases with SNMP
- Switch port mapping
- Data export to common file formats

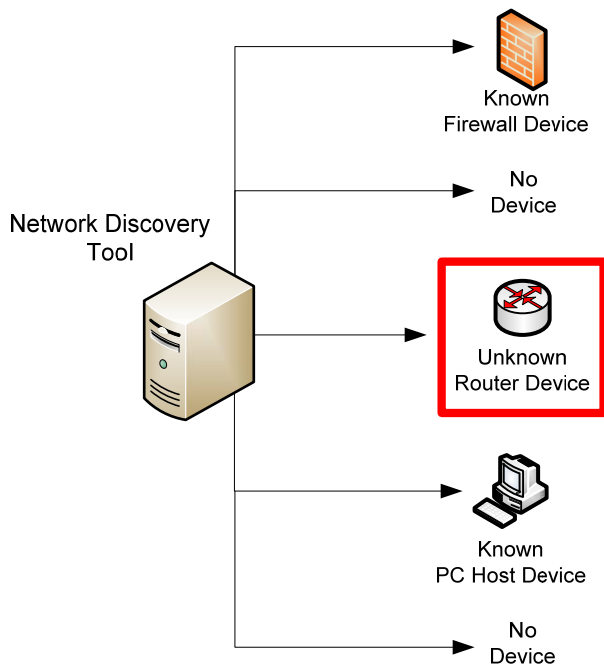



Figure 4.3: A typical network discovery tool will scan a range of addresses to look for the presence or absence of a connected device. Some network discovery tools can compare results with known devices to look for rogue devices on the network.

Attack Identification and Simulation

Administrators unfamiliar with the changes in a network's functionality during an external attack situation will be unprepared for fending off that attack once it occurs. Attack identification and simulation tools enable the administrator to identify when common network attacks occur such as broadcast storms, cache poisoning, replay attacks, and so on. They also allow for the simulation of such attacks upon a network to monitor and analyze the behavior of that network as well as to assist in preparing the network against a real attack by an outside attacker.

Attack identification tools such as network intrusion detection systems and network intrusion protection systems can be complicated to install and manage due to the prevalence of false positives and false negatives such systems can generate. The following list highlights features of interest in either type of tool:

- Performance monitoring elements
- Identification databases with real-time update
- Multiple attack profiles
- Dictionary and brute force capabilities
- Network device security checks
- Port scanning
- Network jamming
- Remote TCP resetting

 Attack simulation tools should be kept out of the hands of unprepared administrators, as such tools have the capability of inhibiting the successful operation of the network.

SNMP Trapping

We've talked about SNMP and SNMP traps before within this guide, but SNMP trapping tools have a different use than those in your NMS. SNMP trap receiving tools are out-of-band tools that can receive, analyze, and display low-level trap information from an SNMP-enabled device for purposes of troubleshooting and SNMP analysis outside the NMS. SNMP trap editing tools allow for the editing of trap templates to customize NMS response when traps occur. These tools incorporate some needed features for advanced SNMP manipulation:

- Data export to common file formats
- Trap manipulation
- Tree view
- Trap mimicking and simulation

Ping, Traceroute, and ARP

Although ping, traceroute, and ARP commands are available in virtually every OS in existence, the tools present natively in these OSs often involve minimal functionality. Additionally, they typically only allow for result output to the screen, lacking the ability to natively capture results into a more useable format.

For network administrators who regularly use these tools, the additional functionality of non-native variants of them may be useful for the troubleshooting process. Consider these added functions when looking for replacements for these tools:

- Enhanced ping timing response
- Data export to common file formats
- Graphical response representation
- Multiple, simultaneous host support
- IP address range support
- Enhanced traceroute result information
- Remote ping sourcing

MIB Browsers

As explained in Chapter 1, Management Information Bases (MIBs) are databases of characteristics about network devices. Those databases are released by the manufacturer and house readable and writable information about the configuration and status of the network device. A MIB Browser is a specialized tool that can peer into the data inside a MIB and pull out relevant Object ID (OID) information. Remember that OIDs are little more than strings of numbers used as unique addresses for device data. A good MIB Browser will include a pre-populated database of known OIDs and their related data. It will also enable the ability to “walk the MIB tree,” gathering all known data from that MIB and presenting it to the administrator.

The real power of an effective MIB Browser is in its ability to view and search the MIB for relevant information and allow the administrator to modify and customize that information as necessary. A good MIB Browser will typically include this functionality:

- Remote device support
- Large database of known OIDs
- View/search/walk via tree-view
- Editing functions
- Reading/writing support
- Multiple-device support



MIB Browsers are primarily used as customization tools for the SNMP-enabled devices plugged into your NMS.

IP Address Management

The next set of tools specifically deals with the management and maintenance of IP addresses. With typical Class C subnets consuming upwards of 254 addresses per subnet and most companies needing multiple subnets, the sheer number of addresses under management can grow huge as the number of subnets increases. Getting your arms around this task can be difficult. The tools discussed in the following sections are designed to assist with that process of managing the scope of IP addresses on your network.

Subnet and Address Calculations

Pundits and conference speakers offer sessions on “How to Subnet in your Head in 90 Minutes.” And there are whole Web sites devoted to assisting with the process of defining the hosts in a subnet based on subnet masking parameters. Thus, obviously this binary math isn’t an easy process. Tools also exist that can assist with this tedious process. These tools give the administrator the ability to define subnet masks and report on the available addresses, broadcast address, and network address associated with those subnet characteristics.

Some tools provide the capability to input hosts into the resulting framework to help identify whether that subnet will provide the necessary space for the hosts in question. Good tools allow for the calculation of subnets both from the needs of residing hosts as well as by knowing the mask bit, host bit, and number of needed subnet information. When considering a subnet and address calculation tool, consider one with the following features:

- Forward and reverse DNS lookups
- Data export to common file formats
- Integration with ping tools
- Multiple calculation parameters
- Address assignment
- Classless Inter-Domain Routing support


DHCP

Interestingly, although the automatic assignment of addresses through the Dynamic Host Configuration Protocol (DHCP) is considered a network function, its administration is usually done by systems administrators. This is usually the case with small and medium-sized networks because the server that handles the DHCP service resides not on a network device but instead on a server.

However, the management of DHCP scopes can leak into the role of the network administrator in situations in which DHCP scopes fill up. In networks with many DHCP scopes at high utilization, when the scope fills to 100%, users interpret the resulting lack of network connectivity as a network problem. In those situations, the network administrator is often the first to be called in to troubleshoot the problem.

Including DHCP scope monitoring tools in your network administrators' toolset can help in these situations as full scope problems are difficult to track down using other tools. When considering a DHCP scope monitoring tool, look for one with the following capabilities:

- Tabular user interface
- Support for BIND and Windows-based DHCP
- Alerting and notification
- Visual identification of full and near-full scopes

 Problems associated with full and nearly-full DHCP scopes can be a troubleshooting nightmare. This is because the client error messages associated with a full DHCP scope in many OSs are unclear. Also, the resolution to the problem is often a re-segmenting of the network to add new subnets. It is for this reason many networks utilize multiple full Class C networks for workstation networks.

If you are having issues with full or nearly-full scopes due to machines that repeatedly come on and off the network, consider reducing the DHCP lease time to a very short amount of time before re-segmenting the network. DHCP renewal traffic is very minimal on today's networks and the added traffic from the increased number of DHCP lease renewals should not significantly impact network performance.

IP Address Management Tools

Where the intersection of the systems and the network administrator can cause difficulty is in the management of available IP addresses for subnets not serviced by DHCP. In typical networks, these subnets often house the network servers and server infrastructure. Because servers are critical components of the network, management of their IP address space is important to ensuring their uptime and availability.

In early networks, systems administrators often use a “ping and pray” approach to finding an available IP address on a server subnet. In this approach, they ping various addresses on the server subnet and look for the first one that does not respond. They then configure the new server with that IP address and “pray” that it wasn't in use by a server experiencing an extended outage. In dynamic situations with servers going up and down for extended periods, this can be especially problematic.

A better approach to using “ping and pray” is to incorporate an IP address management tool that monitors for use of IP addresses in critical subnets. The tool can store the last known-connected device for each IP as well as notify the administrator how long that IP address has either been used or has gone unused. When looking for such a tool, consider the following features:

- Forward and reverse DNS lookups
- Data export to common file formats
- Active monitoring
- Database storage
- SNMP support

Network Engineering Applications

The next suite of applications is used in network engineering and analysis activities. These tools are used when a high-level approach is needed to understanding how the network system as a whole operates, both within each individual device and between the internal network and any externally connected networks. These tools are necessary as they provide the ability to mock-up and analyze potential networking configurations, which enable the administrator to identify where performance bottlenecks and bad designs could impact the network before any purchases are made.

Any network design activity involves some measure of on-paper engineering to ensure that the correct level of connectivity is ensured to support the needs of its hosted applications and users. These tools also assist the designer in validating the correctness of their designs. The three tools we'll look at in this section are protocol analyzers, traffic generators, and network simulation tools.

Protocol Analyzers

Most systems and the applications they run are like “black boxes,” meaning that they internally perform some function while visibility into their inner workings is relatively limited. Because of this behavior in most applications, troubleshooting them when they're not working is difficult. The administrator has to rely on the status messages sent to the system for information on the health of the application.

One way in which some applications reveal a little about their inner workings is in how those applications' individual servers communicate between each other and between server and client. Often, a savvy network administrator can gain a lot of knowledge about an application by watching the packet-by-packet traffic flow going in and out of an application's host server. A protocol analyzer is the tool that enables this capability.

Protocol analyzers are configured to use network interface cards (NICs) in “promiscuous mode” to watch all the traffic along a particular link. Typical NICs only process the data that is addressed to them, but a NIC in “promiscuous mode” will process all data no matter which device it is addressed to. In this manner, the administrator can watch all the traffic coming out of the problematic server and get a good understanding of the inner workings of the failed application.

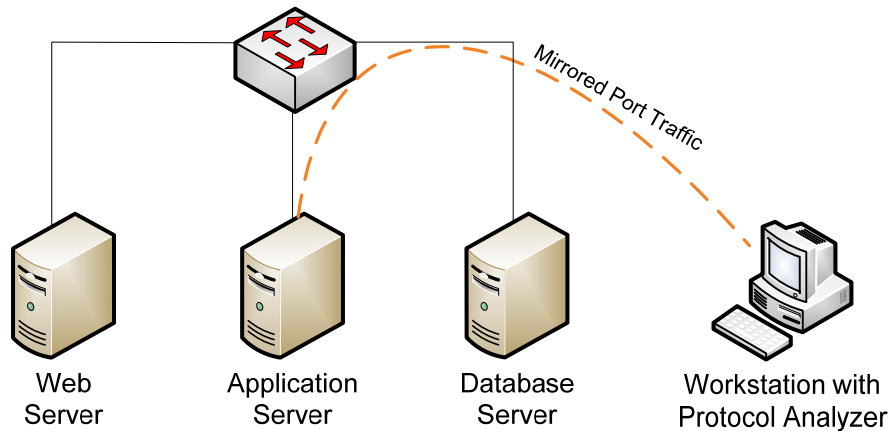


Figure 4.4: In a switched network, for a protocol analyzer to monitor traffic across a link, it is often necessary to mirror that link's traffic to the protocol analyzer.

Protocol analyzers are useful also in finding problems between network devices as well. When network devices are interconnected, they need to communicate with each other to maintain network routing tables (“convergence”) and nearest-neighbor information. By using a protocol analyzer to monitor this network device-to-network device communication, a trained network administrator can track down problems between network devices.

The problem with protocol analyzers is that they produce huge amounts of data, and parsing that data for useful information is a difficult task. A good protocol analyzer will be designed in such a way to categorize, group, and isolate that traffic into flows that are useful for the administrator. Good protocol analyzers also include display filters that convert the binary packet data into human-readable information. Some features of an effective protocol analyzer are:

- Color coding
- Display filters for common protocols/applications
- Traffic graphing and tree mode
- Flow, packet, and protocol analysis
- Low system resource use
- Capture save and replay

There are two big gotchas with protocol analyzers and the process of capturing a packet stream. First, setting a NIC into promiscuous mode and completing a capture is extremely resource intensive for the machine doing the capture. Most protocol analyzers will drop packets when the processor cannot keep up with the flow of incoming data. This can invalidate a capture because of the missing packets. Thus, a good idea when doing a capture is to limit the capture to just the hosts and the protocols for which you need data. Gathering more data than that also adds unnecessary “noise” to the useful data you’re trying to gather.

Second, most modern networks are switched these days, which means that packets are routed by the switching and routing infrastructure only to their ultimate destination and not to every host on the switch. If you’re in a switched network and you notice you’re not seeing any data, you’re experiencing this feature. To get the correct data to the protocol analyzer, you may need to mirror the network port in question to the port where the protocol analyzer resides. The mirroring process should be a feature of your network hardware.

Traffic Generators

The logical opposite of protocol analyzers, traffic generators push out volumes of traffic rather than gather them. The intent with a traffic generator is to simulate load on a network link so that performance metrics can be obtained during periods of known load. Also, traffic spike situations can be simulated to give the administrator a perspective of the network and link behavior during periods of high use. These tools are handy for application testing for applications that will be used over latent network links, like those that span continents or satellite connections.

Good traffic generators have the capability of configuring the amount of traffic to be sent across the connection, the type of traffic to send, and a concurrent measurement of the latency of the connection during the period of use. Network conditions such as jitter, loss, latency, and drop rate can be simulated by configuring them in the generator. An effective traffic generator will include some of the following features:

- Dynamic load adjustment
- Estimated circuit bandwidth entry
- Graphical interface
- Adjustable load percentages

Network Simulation Tools

Network simulation tools allow the administrator to build a mock-up of potential network configurations for purposes of functional and data flow diagramming, pre-purchase functionality engineering, and logical-to-geographical mapping. Some network simulation tools have the capability to map to existing network connections and devices to administrator-defined geographical maps. This functionality allows the administrator to easily see green and red indicators that tell which locations in the extended network are experiencing problems.

This is especially handy in larger networks than span multiple sites. By converting device hostnames and/or IPs into geographical representations, it is easier for the network administrator to triage events as they occur. Network simulation tools typically include some of the following feature sets:

- Green/red indicators
- Administrator-configurable mapping
- Web page support
- Real-time NMS updates

Troubleshooting Involves Good Technique and Good Tools

As has been illustrated throughout this chapter, effective troubleshooting involves the mix of good troubleshooting technique along with a best-in-class toolset. Like the baker and his cake pan or the mechanic and his wrench set, without that toolset, the network administrator cannot perform their job function. The tools used by the network administrator aren't necessarily ones that you can grab out of a yellow toolbox on the back of a truck, but they are mechanisms for enabling the administrator to complete their job.

Throughout this guide, we've discussed a number of ways that implementing good proactive measures into an SMB or mid-market network can improve uptime, monitor fault and performance issues, and generally keep the network humming along. As you can see, good network management involves implementation of good technology to keep an eye on the bits and bytes as they pass through the network. It also involves good practices by the IT department in ensuring that notifications are set up correctly, devices are configured and updated as according to policy, and performance is watched carefully. It is of critical importance that you develop your own skills to take the data you receive from this technology and turn it into something useable and useful for your network.

Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit <http://nexus.realtimepublishers.com>.