

Realtime
publishers

"Leading the Conversation"

The Shortcut Guide[™] To



Network Management for the Mid-Market

sponsored by



SOLARWINDS

NETWORK MANAGEMENT SOLUTIONS

Greg Shields

Introduction to Realtimepublishers

by Don Jones, Series Editor

For several years, now, Realtime has produced dozens and dozens of high-quality books that just happen to be delivered in electronic format—at no cost to you, the reader. We’ve made this unique publishing model work through the generous support and cooperation of our sponsors, who agree to bear each book’s production expenses for the benefit of our readers.

Although we’ve always offered our publications to you for free, don’t think for a moment that quality is anything less than our top priority. My job is to make sure that our books are as good as—and in most cases better than—any printed book that would cost you \$40 or more. Our electronic publishing model offers several advantages over printed books: You receive chapters literally as fast as our authors produce them (hence the “realtime” aspect of our model), and we can update chapters to reflect the latest changes in technology.

I want to point out that our books are by no means paid advertisements or white papers. We’re an independent publishing company, and an important aspect of my job is to make sure that our authors are free to voice their expertise and opinions without reservation or restriction. We maintain complete editorial control of our publications, and I’m proud that we’ve produced so many quality books over the past years.

I want to extend an invitation to visit us at <http://nexus.realtimepublishers.com>, especially if you’ve received this publication from a friend or colleague. We have a wide variety of additional books on a range of topics, and you’re sure to find something that’s of interest to you—and it won’t cost you a thing. We hope you’ll continue to come to Realtime for your educational needs far into the future.

Until then, enjoy.

Don Jones

Foreword

I've written a number of books—including several for Realtimepublishers—focused on network administration. Most of my books, however, have been focused on *enterprise* administration. In today's industry, *enterprise* has become a poorly defined, overused word. I usually use this term to mean *very* large, distributed organizations—ones with tens of thousands of users and annual IT budgets in the millions. Many of the management techniques and tactics I've discussed have been oriented for these larger companies, and might not be appropriate for smaller companies who are trying to work on an IT budget of “merely” a few hundred thousand dollars, or even just a “few” million. It's an important distinction to make: *Most* businesses in the world today *aren't* super-sized enterprises, and *most* businesses don't spend more on IT each year than some small countries have in their entire annual budgets.

That's why I think it's high time we release a book like the one you're now reading. In it, my good friend and colleague Greg Shields tackles network administration from the small- to medium-sized business' point of view. His focus is on the underlying technologies of network administration, and on using tools and technologies that are more practical for a midsized company's operations staff and budget. He'll take a detailed look at options such as open source tools in addition to helping you come up with a good set of specifications for various types of tools—whether you decide to build them yourself, go open source, or evaluate commercial products.

Greg's also going for a lot broader coverage in this short book than I have in other books, where I focused mainly on network device management. He's also looking at network performance monitoring, network analysis, security issues, and much more. That's appropriate, too; the largest companies tend to segregate these tasks across dedicated teams, but midsized companies tend to rely on a handful of “do everything” professionals who may even be responsible for server maintenance and the occasional desktop support call. This book's coverage will reflect those broad and varied responsibilities, although it'll stay firmly focused on the network infrastructure—sorry, no server and desktop support help, here!

The network's been taken for granted for far too long, and it's great to see experienced authors like Greg taking a heightened interest. With Voice over IP (VoIP), intense media streaming, and other new functionality, the network is working harder and harder to deliver the functionality companies require. That means the network administrator—for too long, the guy who was known only for hoarding IP addresses—has to be smarter and more efficient about making the network do its job unfailingly. Management refers to email as the “killer application;” without the network, of course, email would be dead in the water. I think this book takes just the right focus on the network: Keep it running, keep it efficient, and when things *do* go wrong, fix them fast. Greg assures me we're in for a fun ride, so let's jump right in.

Don Jones

Series Editor

Realtimepublisher.com

Introduction to Realtimepublishers	i
Foreword	ii
Chapter 1: FCAPS, Network Management Fundamentals, and Fault Management	1
FCAPS and the Life Cycle of Proactive Management	1
What Is FCAPS and How Can I Leverage it in my Environment?	2
Moving from Reactive to Proactive	3
Ad-Hoc Management	4
Fault Management	5
Configuration Management	6
Accounting Management	7
Performance Management	7
Security Management	8
Choosing the Right Suite of Tools	8
Network Management Fundamentals	9
SNMP	9
MIBs	10
SNMP Traps	12
Syslog	13
Key Steps in Identifying and Correcting Faults	16
Fault Detection	16
Event/Alarm Generation	18
Fault Isolation	19
Fault Correction	20
Key Metrics for Fault Management	20
Mean-Time Between Failures	20
Mean-Time To Restore	21
Network Uptime	21
Relating to Your Business	22
Chapter 2: Performance Management	23
Key Steps in Managing Performance	23
Performance Baselineing	24
Monitoring Deviations	26
Performance Reporting	27

Performance Correction	28
Key Measurements in Performance Management	29
Bandwidth Utilization	29
Network Latency	30
Interface Errors and Discards	31
Network Hardware Resource Utilization	31
Buffer Usage	32
The Business Metrics of Performance Management	33
Availability and SLAs	33
Bandwidth Monitoring	34
Link Costs	35
Traffic Management and Prioritization	36
Additional Tools for Managing Performance	37
Traffic-Generation Tools	37
Traffic-Analysis Tools	38
Wireless Performance Tools	40
Performance Affects Business	40
Chapter 3: Configuration Management and Security	41
Key Steps in Managing an Environment Configuration	42
Establish a Baseline	43
Document Configuration	43
Control Change	44
Audit Environment	45
Ad-Hoc/Manual Configuration vs. Managed Configuration	46
Configuration Standardization	46
Configuration Backup and Archival	47
Post-Incident Restoration	48
Policy-Based Configuration	48
Inventory and Mapping	49
Rogue Device Identification and Adjudication	49
Provisioning	50
Deprovisioning	50
User Access	50

Business Drivers for Configuration Management	51
MTTR Reduction	51
Loss of Business Revenue	51
Security	52
Regulation and Compliance	52
Auditing Requirements	53
Personnel Turnover	54
Supporting Technologies	54
Configuration Analysis and Comparison Tools	54
Task Scheduling Tools	55
RADIUS/TACACS	55
Understanding Security Management	56
Practicing Good Network Security	56
SNMP Community Strings and SNMP Weaknesses	57
Port Scanning and Port Minimization	57
Penetration Testing	57
Vulnerabilities, Exploits, and Patches	58
Configuration and Security Management Provide Measurable Benefit	58
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 Realtimepublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimepublishers.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 Realtimepublishers.com, Inc or its web site sponsors. In no event shall Realtimepublishers.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.

Realtimepublishers.com and the Realtimepublishers 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 Realtimepublishers.com, please contact us via e-mail at info@realtimepublishers.com.

Chapter 1: FCAPS, Network Management Fundamentals, and Fault Management

Building an exceptional network involves the proper mix of skilled individuals, an intelligent design, the correct hardware, and the knowledge and experience to put it together correctly. In the largest of networks, that mix regularly produces some of the best networks in the world. However, there are hundreds of ways to design and run a network.

Money is often considered the differentiator between the best-run and the worst-run networks in the world. When businesses have plenty of cash to throw at their network infrastructure, they end up with industry experts who create best-in-class designs that leverage the market's greatest tools, don't they? Maybe, but having all the money in the world doesn't necessarily mean you're spending it well. Plenty of companies throw millions at their network and don't achieve the great things they had hoped. That suggests that wisdom, not just piles of money, plays a big role in making things great.

Businesses that operate in the small to midsize business (SMB) space and within the mid-market don't usually have the luxury to afford industry experts and the most expensive tools. In the mid-market, the people we label as "network engineers" wear multiple hats, doing server administration in the morning, acting as the Help desk in the afternoon, and working throughout the night on network administration. Thus, SMB and mid-market businesses must think wisely when making network infrastructure decisions. This guide is written specifically for you, the harried network administrator, in an attempt to show you some of the wisest tools and techniques to administer, troubleshoot, and automate your network infrastructure.

FCAPS and the Life Cycle of Proactive Management

The hardest part of any administrative activity is determining exactly what to administer. You're given the keys to the network and told, "Make sure it stays up." The relative stability of modern networks have buoyed the expectations of business management to assume the network is a utility function, just like the lights and power that run the building infrastructure. This increase in expectation on the part of business has resulted in a greater expectation of the network administrator.

To help determine both what and how to ensure that the network "stays up," this guide will discuss a number of topics relevant to the heavily burdened network administrator. You may not know the four fields of a Border Gateway Protocol (BGP) packet header or the intricacies of Open Shortest Path First (OSPF) convergence, but you do know that you've got a bunch of servers that need to interconnect with a bunch of workstations and generally never go down.

This guide is broken into four chapters. This chapter, Chapter 1, will discuss the concept of FCAPS, an acronym that describes a networking model used as a discussion framework for the types of things to keep an eye on inside your network. Each letter in FCAPS discusses one component of network management, all of which we'll explore in a minute. We'll discuss the breakdown of FCAPS and the take-aways from the model that you can use to start doing proactive network management. We'll also talk about fault management and some key tools and metrics you'll want to investigate to help identify and resolve network faults.

Chapter 2 expands on the tools discussed in Chapter 1 to encompass performance management. We'll talk about some key measurements that describe performance characteristics and validate a well-oiled network. The business impact of performance management will also be discussed, giving you the business drivers that justify monitoring and managing performance. And we'll finish with a few key tools and concepts for documenting and reporting on performance.

Chapter 3 continues to build on the core concepts, discussing configuration and security management. As we move from the concept of an ad-hoc network to a fully managed network, we'll focus on the items on your network necessary to bring under management to ensure a stable and secure configuration. In this chapter, we'll discuss some technologies that support a secure configuration and the techniques used by the smart networks to validate security.

Chapter 4 introduces the concepts of network troubleshooting and diagnostics. Often the most difficult part of network engineering is troubleshooting devices when they become problems. This chapter will outline a few tool suites for identifying problem devices, discuss the benefits and tools associated with IP address management and DNS problems, and outline network engineering applications that assist with the troubleshooting process.

What Is FCAPS and How Can I Leverage it in my Environment?

Before discussing effective network management practices, it helps to frame the conversation within an easily understood model. The FCAPS model was originally designed by International Telecommunication Union (ITU-T). This organization dates back to 1865 with original responsibilities of ensuring efficient and on-time production of high-quality recommendations covering all fields of telecommunications. In 1996, the ITU-T created the concept of the Telecommunications Management Framework (TMN), which was an architecture intended to describe service delivery models for telecommunication service providers based on four layers: business management, service management, fault and performance management, and element and configuration management.

Although the TMN was a valid initial mechanism for aligning telecommunications assets to business goals, the ITU-T refined the model in 1997 to include the concept of FCAPS. FCAPS expanded the TMN model to focus on the five functionally different types of tasks handled by network management systems: fault management, configuration management, accounting management, performance management, and security management.

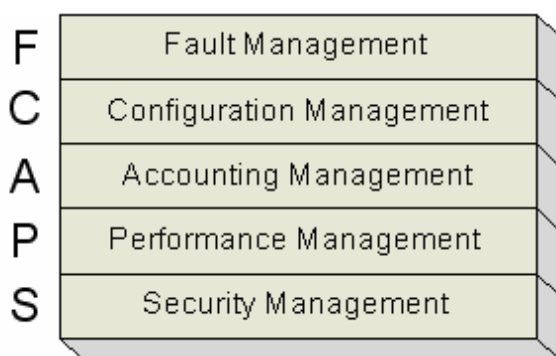


Figure 1.1: The FCAPS model explains the five functional layers of network management.

Although the initial release of the FCAPS model by the ITU-T was intended for telecommunications networks, it was the incorporation of FCAPS into the International Standards Organization (ISO) Open Systems Interconnect (OSI) model that highlighted its usability in describing the necessary functions of network management.

We'll discuss each of these layers in a minute. Within the FCAPS framework, we'll analyze the role of the network administrator and the necessary tools and techniques of network management that support it.

Moving from Reactive to Proactive

If this is the first you've seen or heard about the FCAPS model, you're probably wondering, "How does this affect my ability to better manage my network?" Knowing how to interconnect a series of routers and switches to create a functioning network involves one set of skills. However, as network complexity increases geometrically with an increase in the number of connections, properly managing that network as it grows and scales involves a whole new set of skills.

A little later, this chapter discusses a few technical concepts that enable network management and monitoring, such as the Simple Network Management Protocol (SNMP), SNMP traps, Management Information Bases (MIBs), and the use of Syslog. The use and functionality of these core technologies, among others that later chapters will discuss, are primarily based on the recommended network needs identified by the FCAPS model.

As information networks are growing in size and reliability and as business expects greater uptime and performance from their network backbone, there is an increased need to proactively monitor activity, notify when or before problems occur, dynamically reroute based on conditions, and provide alerting to administrators. To ensure this, it is imperative that the network administrator move from a **reactive** approach to network management to a **proactive** approach. The tools we will discuss over the course of the next few chapters will enable the network administrator to achieve this goal.

Ad-Hoc Management

Most networks begin small. And in those small networks, it is often the work of a small group of trusting individuals to complete the network build and administration tasks. Formalized mechanisms of change control often don't exist in the smallest of networks as the cost of administrative overhead required to support them does not outweigh their benefits.

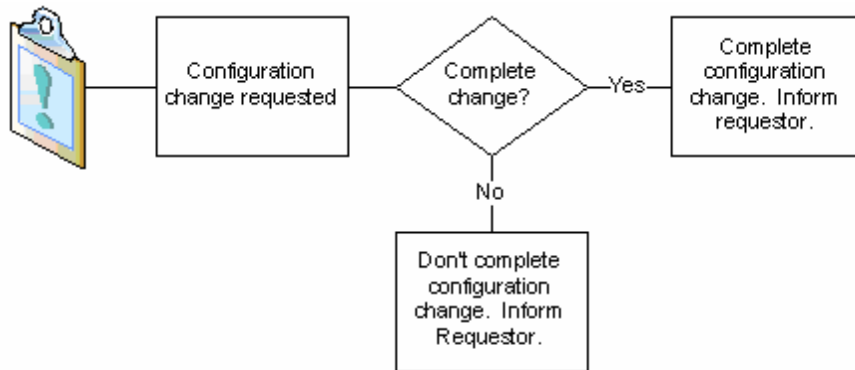


Figure 1.2: In the ad-hoc network, the decision to complete a change lies completely with the administrator.

However, as the size and complexity of the network increases, the number of configurations and the number of *configurators* increase as well. Adding to this complexity is the highly text-based interface of most business-class network devices.

```

firewall multiple-vlan-interfaces
firewall module 1 vlan-group 1
firewall vlan-group 1 200-202
interface vlan 200
ip address 10.0.1.3 255.255.255.0
standby 200 ip 10.0.1.4
standby 200 priority 110
standby 200 preempt
standby 200 timers 5 15
standby 200 authentication Secret
no shutdown
interface vlan 201
ip address 10.0.2.3 255.255.255.0
standby 200 ip 10.0.2.4
standby 200 priority 110
standby 200 preempt
standby 200 timers 5 15
standby 200 authentication Secret
no shutdown
  
```

Listing 1.1: Implementing even a simple switch configuration involves numerous text-based lines of code.

As most companies grow their business and correspondingly grow their networks, the number of configuration items grows geometrically with the size of the networks. Businesses must retreat from ad-hoc management to some mechanism of configuration control to ensure network stability. The FCAPS model can outline the necessary management tasks.

The next five sections will discuss the five functionalities of FCAPS and their associated management tasks. As you move your network from ad-hoc management to full-management, consider the incorporation of network tools that facilitate these management tasks.

Fault Management


The F in FCAPS discusses the management tasks associated with fault management. Among other traits, an effective fault management system will recognize that a problem has occurred, alarm the administrator when that recognition occurs, and provide information as to the location and manner of the fault. Twelve management tasks are identified by the FCAPS model as necessary for a successful fault management system:

- Fault detection
- Fault correction
- Fault isolation
- Network recovery
- Alarm handling
- Alarm filtering
- Alarm generation
- Clear correlation
- Diagnostic test
- Error logging
- Error handling
- Error statistics

Configuration Management

Configuration management is the concept of ensuring a consistent, repeatable, and auditable configuration on all devices that make up the network. Effective configuration management ensures that devices contain the right configuration based on policy. It also provides a mechanism for rapid return to operations of faulted devices, as an effective configuration management tool will store each device's configuration in a separate searchable repository. In today's environment of stringent compliance regulations, only through effective configuration compliance will a network pass an auditor's review. Consider the following management tasks when choosing any configuration management system:

- Resource initialization
- Network provisioning
- Auto-discovery
- Backup and restore
- Resource shut down
- Change management
- Pre-provisioning
- Inventory/asset management
- Copy configuration
- Remote configuration
- Automated software distribution
- Job initiation, tracking, and execution

 A detailed discussion of configuration management will continue in Chapter 3.

Accounting Management

Of the five items in FCAPS, accounting management tasks are potentially the least relevant for many networks. Although some network backbone architectures and organizations honoring Service-Oriented Architectures (SOAs) may incorporate charge backs and cost-based servicing, most networks in the mid-market likely don't incorporate these accounting systems. However, some components of accounting management—such as the need to track resource use for metrics generation—should be a component of any mature network. The following list highlights the eight considerations for tools that enable accounting management:

- Track service/resource use
- Cost for services
- Accounting limit
- Usage quotas
- Audits
- Fraud reporting
- Combine costs from multiple resources
- Support for different accounting modes

Performance Management

Performance management involves the effective monitoring of a network's response time and the proactive management of needed upgrades to support its users. Performance management expands upon simply answering the question, "Why is the network so slow?" It involves proactively analyzing a network's activity and making informed business decisions about expansion before performance becomes critical. Businesses who engage in monitoring and taking action based on performance management can recognize substantial return on investment (ROI) based on prevention of loss of worker efficiency due to network conditions. When looking at performance management systems, look for the following traits:

- Utilization and error rates
- Performance data collection
- Consistent performance level
- Performance data analysis
- Problem reporting
- Capacity planning
- Performance report generation
- Maintaining and examining historical logs

Security Management

Aligned with the needs of configuration management, the tenants of security management ensure the integrity and reliability of the network. Many network devices by default enable security through a shared password concept, which can be a violation of established security policies. Enabling successful security management means segregating the roles and responsibilities of administrators and users, logging their activity, and ensuring the privacy of data on the network. An effective security management system will provide mechanisms for security administrators to easily record network activity and parse that activity for anomalies. Consider the following activities as critical for an effective security management system:

- Selective resource access
- Access logs
- Data privacy
- User access rights checking
- Security audit trail log
- Security alarm/event reporting
- Take care of security breaches and attempts
- Security-related information distributions

Choosing the Right Suite of Tools

It should be obvious that no single management system can likely handle each and every one of these network management activities. However, a suite of tools can provide for the necessary subset of capabilities needed by your business. When considering a network management suite of tools, your business should consider a trade study process for identifying your requirements and separating requirements from those items considered “nice to have.” The process of completing a trade study can run from an informal intra-group decision-making process to the use of a formalized trade study framework.

Product Scoring		Product 1		Product 2		Product 3	
	Weight	Score	Weighted Score	Score	Weighted Score	Score	Weighted Score
Fault Detection	9.0	8	72.0	10	90.0	9	81.0
Fault Correction	9.3	4	37.3	4	37.3	5	46.7
Diagnostic test	10.3	6	61.8	8	82.4	2	20.6
Error statistics	4.3	8	34.4	2	8.6	5	21.5
Network recovery	5.3	4	21.3	8	42.7	3	16.0
Total		30.0	226.9	32.0	261.0	24.0	185.8

Table 1.1: An example of a formalized trade study that weighs the need for features and compares each product's capability to deliver on that feature.

Network Management Fundamentals

Four fundamental technologies are enablers for much of the characteristics discussed in the FCAPS model. These technologies are SNMP and the related concept of SNMP traps, MIBs, and Syslog. These four technologies interoperate to provide monitoring capabilities for devices, notification to administrators when preconfigured conditions occur, and storage of device and log information in searchable formats. Throughout the rest of this guide, we will refer back to these core technologies as we discuss additional tools available to network administrators. Most important, these technologies are key in moving a network from an ad-hoc and reactive mode to one that is fully and proactively managed.

SNMP

SNMP describes a network protocol as well as an information framework used to provide remote monitoring and configuration capabilities for network devices. The main purpose of SNMP is to enable the centralization of network device management and monitoring through a common language across all devices.

SNMP-enabled devices are said to be SNMP agents. Enabling SNMP on these devices provides the capability of being remotely interrogated by an administrator through a Network Management Server. The NMS incorporates high-level management software that typically has the capability of storing agent device configurations within a management database. The NMS additionally has the capability of receiving notifications from devices when preconfigured conditions occur. This notification is called an SNMP trap and will be discussed in the next section.

The utility of the SNMP protocol is in its extensibility. As the protocol and framework are device-independent as well as NMS-independent, this allows for the interconnection of all SNMP-capable devices of any vendor into a single domain of management. The NMS chosen for management of the devices is not reliant on the device vendor and multiple NMS' can be used to manage the same devices.

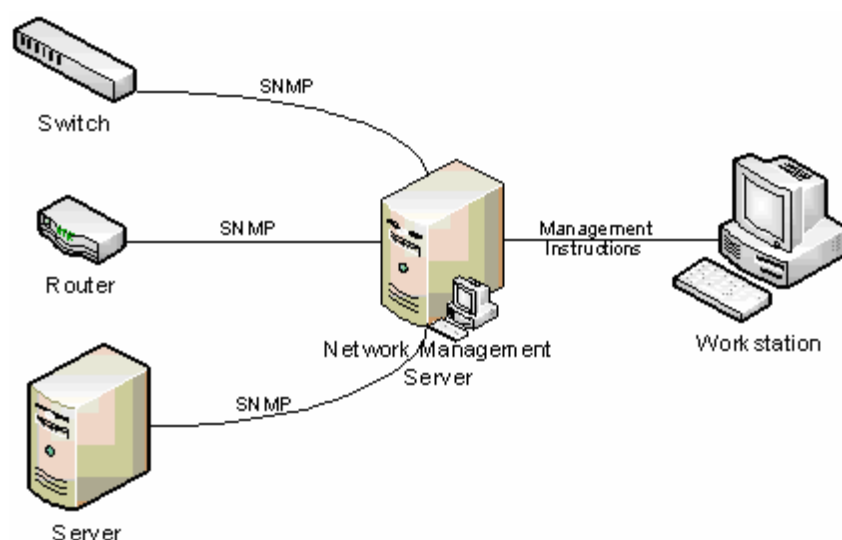


Figure 1.3: SNMP devices report information to the NMS. Administrators instruct the NMS to collect and update information on managed devices.

SNMP can be configured to allow for writing and/or updating of configurations on managed devices as well. This capability, leveraged through the NMS, greatly enhances the configuration control capabilities within a managed network. Using SNMP to manage configurations means that configuration files can be centrally stored within the NMS database. The NMS can be used as a gatekeeper for devices, ensuring that only the approved and agreed-upon configuration is enabled on the device. The configuration database can additionally be backed up using typical network backup devices, ensuring that configurations can be rapidly restored in the case of a disaster event.

Listing 1.2 shows the commands needed for a basic SNMP configuration on three different types of Cisco devices. The first is a Cisco IOS-based router; the second is a Cisco CatOS-based switch; the third is a Cisco PIX firewall.

```
router1#config terminal
router1(config)#snmp-server contact admin@abccorp.com
router1(config)#snmp-server location Downtown Office 2nd Floor Router
router1(config)#snmp-server chassis-id 123456
router1(config)#snmp-server community public ro
router1(config)#snmp-server community private rw

Console> (enable) set snmp community read-only public
Console> (enable) set snmp community read-write private

firewall1#config terminal
firewall1(config)#snmp-server contact admin@abccorp.com
firewall1(config)#snmp-server location Downtown Office 2nd Floor Router
firewall1(config)#snmp-server community public ro
firewall1(config)#snmp-server community private rw
```

Listing 1.2: Commands for a basic SNMP configuration on three different types of Cisco devices.

MIBs

MIBs are collections of device characteristics that are available for reading and writing via SNMP. Think of a MIB as a small database that contains all the characteristics of the device. Individual device settings within a MIB are called *MIB variables*. Within a device, all MIB variables are collected into a single document object called a *MIB module*. It is within these MIB modules that each managed attribute of a device is described and its interfaces are discussed. MIB modules are specific to each particular device and must be preloaded into the NMS for the NMS to be able to manage the object.

MIB modules are typically delivered with the network device or can be downloaded from the device manufacturer's Web site. As the number of MIB modules for a complex network can get extremely large, effective management of these MIB modules is one of the primary features of a good NMS.

Although MIBs are not addressed by their location on a network, there is a hierarchy to MIB modules, called the *MIB tree*, which allows for each MIB across all device types to be uniquely addressed. Managed by the Internet Assigned Numbers Authority (IANA), this unique addressing ensures that two manufacturers cannot use the same MIB module information. The unique address for any MIB module is called its Object ID (OID) and is represented by a long string of dot-separated numbers.

This long string of numbers references the object's location on the tree. The first number in the OID references its position at the top of the tree—the most general descriptor—and each subsequent number further defines the object in relation to its position on the tree.




For example, the OID for a Synoptics 3000 concentrator is 1.3.6.1.4.1.45.1.3.2. The first five numbers of this OID reference iso(1).org(3).dod(6).internet(1).private(4). As you can see, the tree is exceptionally general at the top levels, not even reaching the Internet until the fourth level. Substantial numbers of objects not typically thought of as network devices can be managed by SNMP. Some examples are water-level indicators, air quality measuring tools, and entryway monitors.

Using the OID as the addressing for that device configuration, four types of communication transactions can occur between the NMS and an agent. The following list highlights these four types:

- **Get**—The Get operation is initiated by the NMS to retrieve information from a managed device. Because Get operations require exact addressing, the NMS will provide a complete OID to the agent to locate the characteristic of interest. The agent will respond with the value of the requested MIB variable and the NMS will store this information and/or notify the administrator of the result.
- **Getnext**—Similar to a Get operation, the Getnext operation is used when additional information is needed. Where the Getnext operation is different is that it does not need to provide exact addressing information for the requested characteristic. Instead, Getnext will request the next characteristic in the tree.
- **Set**—SNMP can be used for both read and write operations. The Set operation is used when the administrator wants to update or change the value of the requested characteristic.
- **Trap**—An SNMP trap is a way for an agent to notify the NMS that a preconfigured condition has occurred. Traps are the main component of the notification piece of SNMP.

SNMP Traps

As stated earlier, an SNMP trap describes the ability for an agent to notify the NMS that a preconfigured condition has occurred. The trap is a unidirectional notification from agent to NMS that includes the OID of the characteristic of interest and its associated MIB value. Trap characteristics must be configured within the NMS and relayed to the agent prior to the condition occurring. The agent must be preconfigured with a network location to send traps. Multiple locations can usually be configured for redundancy purposes.


 For example, let's assume the agent on a managed device is configured to watch the internal temperature conditions on that device. The administrator sets a threshold of 100° for those conditions. When the temperature exceeds 100°, the device will initiate a trap to the NMS notifying it (and the administrator) that the condition has occurred. Typical NMSs provide the capability of setting trap characteristics such as value, time exceeded, amount of time exceeded prior to trapping, and return-to-normal thresholds. These added trap characteristics prevent situations occurring in which, for example, the temperature rapidly bounces back and forth between 99° and 101° and the administrator is repeatedly notified.

Where SNMP traps make the biggest contribution in moving a network from reactive to proactive is in the ability to link trap information to administrator alerts. Typical NMSs can be configured to notify administrators via email, page, or SMS message when a trap condition occurs.

Remember that sending a trap does not accomplish anything towards fixing the condition on the device. It is still up to the administrator to resolve the issue once the notification of the issue has been raised. In some situations, however, if SNMP set commands are enabled, the NMS can be preconfigured to perform an action when a trap occurs. That action can be to change a MIB value or even shut down the system.

These action capabilities available to the administrator are based on the intrinsic capabilities built-in to the device and its SNMP interface as well as the feature sets available on the NMS. The decision to purchase an NMS may include a determination whether that NMS has these sorts of automatic capabilities built-in to the system.

Often, these capabilities require some code development. Although it is relatively easy to set up and configure an NMS to handle SNMP and SNMP traps, the greatest portion of any NMS installation is usually in the individual tuning of alerts.

 Many an administrator has lost nights of sleep due to an overly sensitive NMS alerting that wakes the administrator in the middle of the night for a non-critical event. Consideration should be made for quality of life issues for any administrator whose pager is linked to an overly sensitive NMS!

Syslog

Syslog is to centralized system logging as SNMP is to centralized system configuration control. Syslog is a mechanism for sending event messages from managed devices to a centralized server that runs the Syslog service. Although there are some minor differences between the various forms of Syslog that handle Windows, Cisco, UNIX, and other device messages, many native and commercially available Syslog applications can handle event messages from all these types of devices.

The Syslog service is configured on a server within the network environment to accept messages. The nature of the Syslog service is such that the transmission of message information is unidirectional from the sending device to the Syslog server. There is no acknowledgement of receipt. The service stores these messages in a searchable database that can be queried by a management workstation. Typical types of Syslog messages include system error messages, statistics, system warnings, security and access notifications, and access denied notifications, among others.

Syslog is particularly useful for ensuring security in networks due to the external nature of the log collection. If a network device is compromised by an intruder as part of an external attack, one of the tasks usually completed by that intruder is to remove any record of their presence on the device. This process of wiping clear the logs can prevent a security administrator from detecting the compromise of the device and hinders the ability to track the entry point and location of the intruder. By enforcing that all system logs are immediately copied both to the local device as well as to the Syslog server, there is a greater chance of the actions of the intruder being detected.

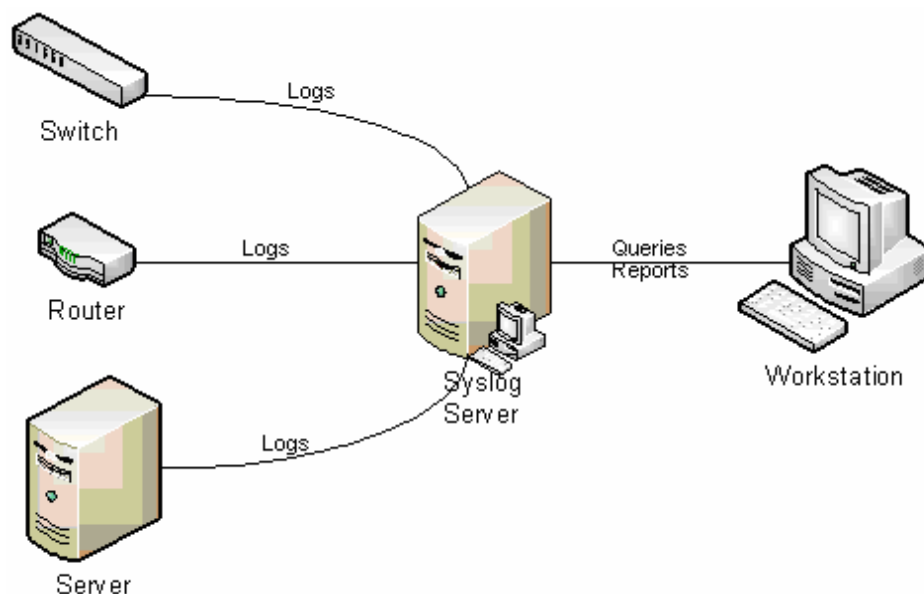


Figure 1.4: Similar in architecture to a network's SNMP infrastructure, Syslog centralizes event log information.

Although there is no standard that defines the content of a Syslog message, most Syslog messages consist of three parts:

- **PRI**—The PRI part of a Syslog message determines the message's priority. This priority is a numerical value that is a mathematical combination of the facility generating the message and the severity of the message.
- **HEADER**—The HEADER part of a Syslog message contains the timestamp denoting the creation of the message along with either the IP address or the hostname of the sending device. This part is used to identify the date, time, and origin of the message.
- **MSG**—The MSG part of a Syslog message contains the text of the message itself. The MSG part can typically have two fields, called the TAG field and the CONTENT field. The TAG field typically stores the name of the application or process that generated the message. The CONTENT field typically contains the message itself.

At first blush, the lack of standardization in the Syslog format can appear to be a weakness in its implementation. However, it is within this lack of standardization that Syslog gains its popularity and its near universal acceptance. Syslog requires only priority and origin information for each message, so it becomes exceptionally easy to use for all types of devices from Cisco to Microsoft Windows to all flavors of UNIX to network hardware appliances. This universal acceptance of the Syslog format means that Syslog can be used in the collection of nearly all device information on the network.

Often, problems on the network involve more than one device and correlating the event information for each of these devices can be cumbersome. But leveraging a centralized mechanism for storing event information for all devices means that a longitudinal timeline across all devices can be easily queried. This gives the administrator a more holistic view of the network and provides better detail into identifying the problem.

When choosing a Syslog system, it is important to choose one that has the capability to handle Syslog messages from each of the types of devices connected to it. As Listing 1.3 shows, different device types can have slightly different log formats. The onus for reading these device formats is on the application hosting the Syslog service.

```
*Mar 6 22:48:34.452 UTC: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Loopback0,changed state to up

2000 Feb 21 12:00:27 PST -07:00 %SYS-4-SYS_HITRFC: 65% traffic detected
on switching bus

<110> Oct 16 08:58:07 64.103.114.149 CisACS_13_AdminAudit 18729fp11 1 0
AAA Server=tfurman-w2k,admin-username=local_login,browser-
ip=127.0.0.1,text-message=Administra
tion session finished,
```

Listing 1.3: Examples of Syslog messages from Cisco IOS and CatOS devices. Note the differences in their format.

There are multiple applications hosted on Microsoft Windows as well as various flavors of UNIX that can support these formats. Those hosted on Microsoft Windows are typically non-native applications layered on top of the operating system (OS). For UNIX, the native syslogd daemon is the standard. Tradeoffs in usability, searchability, a graphical interface, and cost exist between a native UNIX solution and a non-native Microsoft Windows solution. During your decision-making process, consider these tradeoffs, the types of connected devices, and the querying and reporting capabilities you will find necessary in completing your log analysis tasks.

Listing 1.4 shows the commands needed for a basic Syslog configuration on three different types of Cisco devices. The first is a Cisco IOS-based router; the second is a Cisco CatOS-based switch; the third is a Cisco PIX firewall.

```
router1#config terminal
router1(config)#logging 192.168.0.200
router1(config)#service timestamps log datetime localtime show-timezone
msec
router1(config)#logging facility local1
router1(config)#logging trap warning

Console> (enable) set logging timestamp enable
Console> (enable) set logging server 192.168.0.200
Console> (enable) set logging server 192.168.0.200
Console> (enable) set logging server facility local1
Console> (enable) set logging server severity 4
Console> (enable) set logging server enable

firewall1#config terminal
firewall1(config)#logging timestamp
firewall1(config)#logging host 192.168.0.200
firewall1(config)#logging facility 17
firewall1(config)#logging trap 4
firewall1(config)#logging on
```

Listing 1.4: Commands for a basic Syslog configuration on three different types of Cisco devices.

Key Steps in Identifying and Correcting Faults

Fault management is the first item in the FCAPS model, dealing specifically with the identification, isolation, and correction of network faults. Although proactive management is always a consideration with administering a network, fault management specifically deals with the issue of post-incident remediation.

Typical fault management systems use the four steps of fault detection, event/alarm generation, fault isolation, and fault correction to break down the complicated task of identifying and resolving the fault.

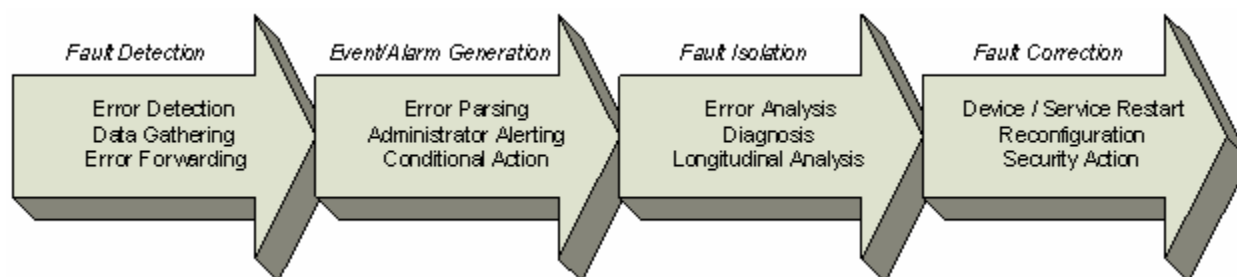


Figure 1.5: The four phases of fault management.

Fault Detection

As Figure 1.5 illustrates, multiple actions can occur at each phase when a fault occurs. The major difference between reactive network management and proactive network management occurs in the first two phases. Business networks with small numbers of devices likely do not have an NMS in place to notify administrators when a fault occurs. In these cases, often the mechanism for fault detection is when a user determines a change in the state of their connection. They contact the appropriate personnel in the IT organization—often a Help desk—and notify them that they have noticed an anomaly on the network.

For the smallest of networks, this ad-hoc mechanism of notifying IT personnel of a network fault is sufficient. In these networks, uptime requirements and user expectations are low for the health of the network.

Ad-hoc notification, however, does have the impact of creating a high number of false positives. As non-technical users are relied upon for the notification of faults, they can overuse their responsibility, notifying the IT department during non-fault situations. This has the effect of causing additional workload on IT to track down phantom problems. Ad-hoc notification also works predominantly with up/down situations as more complicated performance, stability, and security calculations are difficult.

The proactive network leverages technologies such as SNMP and Syslog to provide device notification of fault conditions. Two types of fault management, active and passive, can be configured within the NMS for notification:

- **Passive fault management**—In passive fault management, SNMP-enabled devices notify the NMS when a preconfigured condition has occurred. Passive fault management can track when conditions vary from nominal, but it is reliant on the administrator to preconfigure SNMP agents with the business' definition of nominal activity. This type of fault management is highly successful for identifying problems not associated with a device outage: performance issues, individual service or interface problems, out-of-boundary changes in network traffic flow, and so on. However, passive fault management suffers from an inability to notify during a complete device outage. After all, if the device is non-functional, the SNMP agent cannot raise a notification to the NMS.
- **Active fault management**—Active fault management is a mechanism for notifying when those outage conditions occur. Active fault management typically runs concurrent with passive fault management and usually from the same NMS. In active fault management, the NMS sends a network PING command to each managed device on a regular basis and listens for the reply. If the device does not reply after a preconfigured interval, the active monitoring will notify the administrator of a device outage. For this reason, active fault management is often also referred to as “up/down monitoring.”

When choosing a fault management system, consider one that can provide both active and passive management. Well-designed fault management systems can also provide some internal logic that enables the network administrator to pre-generate a series of if/then statements associated with the recognition of faults. Six categories of logical conditions and actions can make up a typical fault management system.

- **Time of day**—Most Service Level Agreements (SLAs) for business networks assign a period of network operability. This period can occur from typical business hours (8:00am to 5:00pm, Monday through Friday), to full 24/7/365 operations. In either case, there are usually times in which alerting is nonsensical considering the actions taking place on the network, such as during maintenance windows. Time of day logic allows the administrator to assign times when fault detection will be disabled or enabled but lacking notification.
- **Trigger condition**—The trigger condition for a fault is the administrator-configurable variable that identifies faults of interest. Device MIBs can notify on dozens or hundreds of possible conditions but only a few will be relevant per device for each particular network. Effective NMSs incorporate robust logic for trigger condition creation.

- **Reset condition**—Reset conditions are the opposite of trigger conditions. When a device has gone over a threshold and a fault has been identified, there must be some mechanism of automatically resetting the notification of the fault to nominal. Similar to trigger conditions, effective NMSs incorporate similar robust condition logic to enable creation of reset conditions.
- **Alert suppression**—In some situations, the administrator might want conditional-based suppression of fault notification. In those cases, alert suppression logic allows for the creation of values that prevent fault identification. This may be based on known issues within the network, such as known problems with particular network devices.
- **Trigger and reset actions**—Robust NMS systems should provide for a suite of actions to occur when a fault is recognized and again when that fault has been reset. The next section will discuss examples of these actions.

In all cases, these data gathering and error forwarding actions are components of the managed device. The managed device will forward the error, usually through SNMP or Syslog, to the NMS for processing.

Event/Alarm Generation

When a fault occurs and the fault information has been forwarded to the NMS, the NMS has a few actions of its own to accomplish. First, that error must be parsed and compared with the pre-generated logic as explained in the previous section. When alerting matches have occurred, the NMS will trigger a notification to the administrator through one of many mechanisms. Some of these mechanisms for notification include:

- Sending an email message
- Sending an SMS message to a cell phone or pager
- Playing a sound or recorded message on the management workstation
- Logging the alert to the Network Event log
- Logging to a text file
- Sending a Syslog message
- Sending an SNMP trap
- Logging the alert to a Microsoft Windows event log
- Sending a Microsoft Windows Net-Message
- Executing an external program
- Executing a script
- Speaking an alert message using a text-to-speech engine

As you can see, numerous capabilities for notification exist. A robust suite of notification actions on fault triggering and fault resetting ensures that administrators are properly notified no matter where they may be and no matter what they are using as their primary notification device. Additionally, robust external program and script execution further enhances an NMS' proactive capabilities, enabling common triggers to be automatically resolved by the system through execution of an application of a program or a script.



For example, assume that a router has a specific networking problem that regularly occurs and the known resolution is to flush the ARP cache. In that case, in which both the problem and the resolution are known, a trigger condition can be set up to run a script that automatically flushes the ARP cache when the condition occurs.



Be careful with automatic actions! Configuring scripts to automatically fire when conditions occur can sometimes exacerbate a problem or band-aid it without actually resolving it. What if that router problem is actually an external hacker attempting to infiltrate the network? Setting the action to automatic may prevent the administrator from noticing the attempt.

Fault Isolation

Fault isolation is a step involved with complicated problems where identifying the root cause is difficult and where the problem may span multiple devices. In these cases, a root cause analysis is often the troubleshooting approach to tracking down the problem.

For complicated problems, fault isolation can involve substantial error analysis and deep diagnosis of the problem. These sorts of deep-dive problems can occur in large networks with multiple administrators who manage devices within separate but connected domains of management.

In these cases, the task of fault isolation can involve review of logs across multiple devices and across multiple management domains. Aggregation of Syslog and SNMP data across multiple devices and management domains may be necessary to track down the problem. A *longitudinal* or *cross-device* approach to problem isolation can significantly improve results. In a longitudinal approach, logs are correlated across multiple devices and listed in time order across all devices. Using this approach, it is possible for the administrator to get a “big picture” view of the network and its interconnected devices.

One other approach is the use of network mapping to provide a graphical representation of the interconnected devices and show their status and interrelation. Many network management tool suites have the capability of automatically discovering devices on the network and auto-generating network maps of those devices and their connections.

Fault Correction

Fault correction is the obvious last step in the process of fault management. With fault correction, the administrator has identified the fault—often the most difficult and time-intensive part of the process, recognized the corrective actions through analysis, and is ready to commit the corrective action.

Three items are identified as potential actions involved with fault correction:

- Device/service restart
- Reconfiguration
- Security action

All three of these actions involve a potential change on the part of the device. Approval and personnel notification of these changes should be disseminated through some form of change management. Additionally, tools and technology exist to ensure that the change is logged into a configuration management database.



Chapter 3 will discuss change management fully and the tools that exist to enable this storage of configuration information.

Key Metrics for Fault Management

The last sections of this chapter discuss useful business metrics for measuring a network and the faults that may occur within that network. These metrics are useful for trending purposes so that businesses can understand the health of the network and recognize over the long-term when that health begins to degrade. This trending analysis is especially necessary during periods of business growth and the resulting network growth that accompanies it. As additional nodes on a network come online, the trending of metrics in fault management can provide the information needed to make informed decisions regarding purchasing and network expansion.

Mean-Time Between Failures

Mean-Time Between Failures (MTBF) is a hardware-based metric provided by manufacturers to customers to denote the average amount of time that occurs between failures on a particular device.



For example, when you purchase light bulbs, you are given the option of the “regular” brand that last for 1 year or you can pay a premium for “long life” bulbs that may last for 10 years.

Although some manufacturers will no longer release their MTBF statistics in today’s marketplace, analyzing industry studies on MTBF metrics for the products in your environment will help you make informed purchasing decisions. MTBF statistics directly relate to the network uptime metric and all relate to loss of worker productivity associated with network failure.

Mean-Time To Restore

Mean-Time To Restore (MTTR) is typically a metric internally defined by the business. Some businesses can sustain a multiple-day outage with little affect on operations. Some networks cannot survive more than a few minutes of outage before the dollars-per-minute of downtime grows critically expensive.

It is important for businesses to very early determine their MTTR metric for each network service based on the pain associated with the loss of that service. This is critically necessary as disaster recovery and fault prevention technologies come in many shapes, sizes, and costs. There is an inverse geometric relation between the sensitivity to outage (and the related shortened MTTR) and the cost to implement preventative mechanisms.

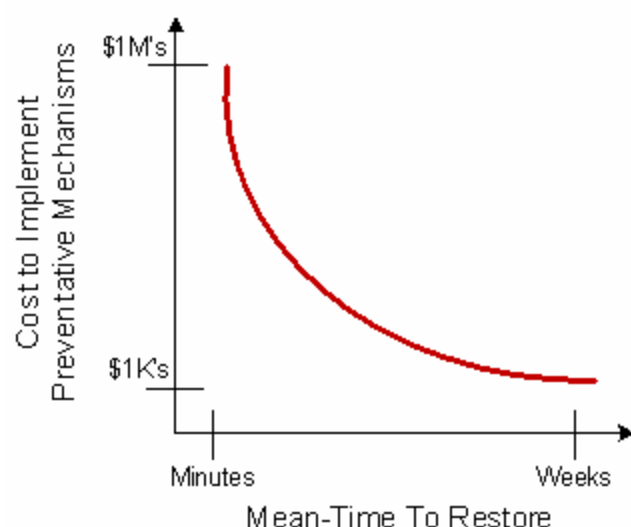


Figure 1.6: *The less downtime the business can handle, the more expensive it is to implement technology to prevent it.*

Network Uptime

Network uptime is the final metric associated with fault management. Network uptime can be defined as an integer number associated with the amount of time between individual device outages. More often, however, it is related to the amount of time that users on the network are not able to accomplish their daily tasks due to a network problem. This is more of a holistic approach to total system availability.

This number is different than MTBF because MTBF numbers typically relate to an individual device and its outage potential. In business networks, redundancy is typically incorporated into the network design that lessens the effect of a device outage on the ability for workers to accomplish work. Network uptime is a useful metric for reporting to management based on the ability for the network administrator to ensure worker productivity.

Relating to Your Business

Obviously, all the metrics in the world matter little if they mean nothing to your business. Depending on the needs of your business, the process by which you identify and resolve faults and the means by which you measure your success will be driven by internal needs. What is important to take away is that there are applications available in the marketplace that can move your network from reactively dealing with faults to proactive notification when faults occur and automated response to deal with them.

The next chapter will take what we've explored thus far about network management and relate it to another item in the FCAPS model—the concept of performance management. Chapter 2 will talk about the important measurements you can employ to determine the usability of your network infrastructure as well as the related business metrics. We'll go over some performance management concepts for documenting your existing environment and planning for expansion and discuss tools that can help you ensure a well-oiled network.

Chapter 2: Performance Management

The real productivity killer in most networks is a performance level that doesn't meet the needs of users. When network performance is consistently below acceptable levels, business cannot operate at full efficiency, workers can't accomplish tasks on time, and the regular movement of business suffers. To exacerbate this situation, smoking out performance issues on a network is virtually impossible without the proper toolset. If you've ever gotten the dreaded "the network is slow" phone call, you know how difficult it can be to track down the problem. This chapter will discuss the tools that can prepare you for when that call comes, enabling you to respond with "I'm on it. I know what the problem is."

The previous chapter outlined the FCAPS model of network management and how that model will be used to guide the conversation on network management fundamentals. It then zeroed in on the F in FCAPS to talk about fault management. That discussion broke down the steps in fault management and talked about the best ways to implement formal and informal tactics in detecting and correcting faults. It also illustrated how implementing an effective network management system (NMS) that provides for monitoring and alerting is the first step in moving from a reactive administration model to one of proactive administration. Chapter 1 dove into four key technologies—SNMP, SNMP traps, MIBs, and Syslog—and how these four technologies are critical for the operation of a successful NMS.

This chapter will build on this foundation discuss how these technologies can be used for the P in FCAPS: performance management. Starting with an analysis of the four key steps in managing performance, this chapter will enlighten you about the items to document, the metrics to monitor, and the actions to take to ensure your network is operating at peak efficiency.

Key Steps in Managing Performance

Although FCAPS identifies performance management at the same level as fault management, one could argue performance management is a subset of fault management. They both involve the identification and elimination of issues on the network that reduce worker productivity. Identification of performance bottlenecks in your network is done with nearly the same troubleshooting and scientific method as the process for identifying faults.

However, where performance management differs is in how it affects worker productivity. Fault management is easy for a business to incorporate because of the natural on/off nature to faults. When a network device incurs a fault, that network device typically "goes down." Non-functioning network devices have a clear need for fixing because they prevent work from occurring. Within performance management, the clear line between up and down grays somewhat because no device is technically down. Performance issues not attended to can linger for an extended period of time, causing a long-term reduction in network and worker productivity.

It is this graying of the problem that makes performance management so difficult to track from a metrics point of view. If the network pipe between your main office and branch office has a 20ms latency, how does that affect worker productivity? What about if that latency grows to 200ms?

Let's take a look at a variation of the key action steps analyzed in Chapter 1. Figure 2.1 describes the four phases of performance management and the actions associated with those phases.

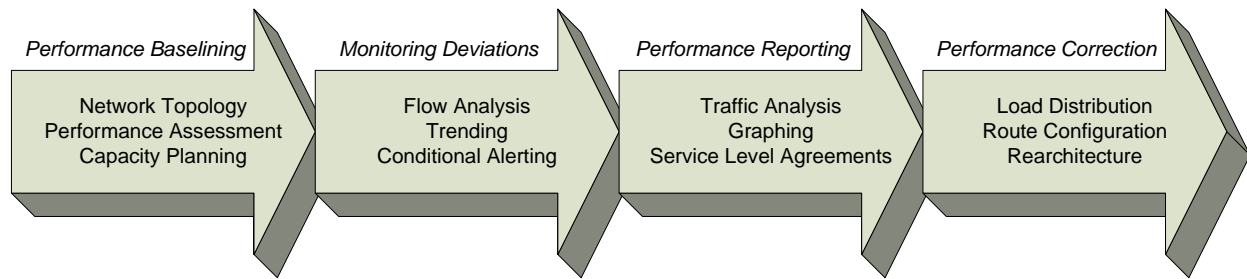


Figure 2.1: The four key steps in managing performance and their associated action steps.

This graphic shows that performance management is really about monitoring for deviations from nominal performance. In the first phase, a baseline of performance is captured using various baselining tools. This baseline is used in monitoring for deviation, and further tools and techniques such as trending and conditional alerting are leveraged to watch for those deviations. Performance reporting is a further component of the deviation monitoring that can provide a graphical representation of performance to the administrator and to business management. All these feed into the correction of the performance issue. That correction may involve a distribution of the network traffic, a reconfiguration of the route, or a rearchitecture of the underlying network.

Performance Baselining

The first step in any performance management activity is to truly understand your network. If you cannot understand your network in the “good times,” you have no comparison for identifying problems during the “bad times.” Plus, knowing what you consider good performance gives you the quantitative metrics to justify or invalidate users’ complaints of “the network is slow.”

There are three major components of performance baselining: Documenting your network topology and the components that make up that topology, completing a performance assessment of the critical applications you want to bring under management, and the use of both of these tools for understanding and planning for capacity needs.

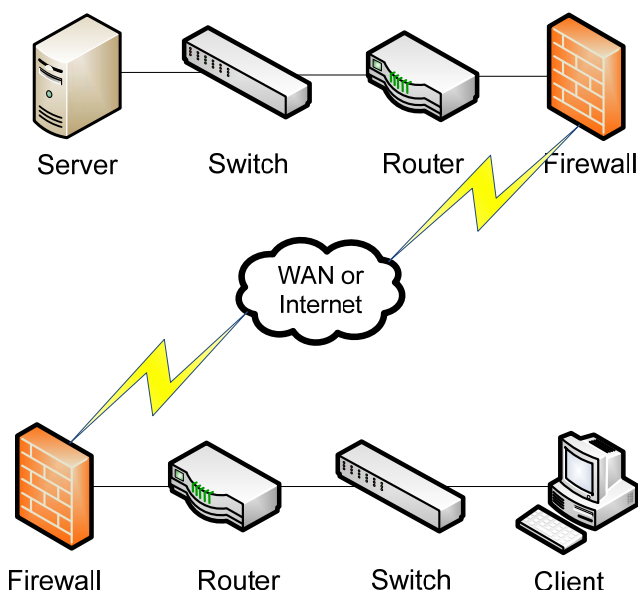


Figure 2.2: Connecting a server to a workstation involves numerous hops from server to switch to router to firewall, through a public network connection, and back through the same devices.

Network performance involves a thread of connections from device to device before the information gets from server to workstation. At each point along that thread, the configuration of the network device and its connection to other network devices will have an impact on the total time it takes data to traverse the network. If even a single connection is misconfigured at a lower connection speed or duplex, it can significantly reduce the total delay.

Effective NMS tools provide the capability for autodiscovery of network devices and their related interconnections. This process of network discovery significantly reduces the time for drawing your network topology. Additionally, a good NMS tool will provide the configurations that explain the lines between the devices to make it easy to spot misconfigurations.

Once the topology of a network is known, the process of completing the performance assessment grows simple. To understand the “good times” of your network, it is necessary to understand what is considered appropriate timing for data to traverse the network. Timing application initiation, data retrieval, and data writing across the network will give you an understanding of the nominal performance of the network. It is within this activity that the concepts of bandwidth utilization and network latency grow important. These terms will be discussed in a minute. But for now, recognize that a time-and-motion study of the network is part of your performance assessment action.

These time-and-motion studies can also be referred to as Network Readiness Assessments (NRAs). Specific to an NRA, you are documenting the attributes of your network to verify its capability to support its hosted applications. In an NRA, you will use the NMS tools described earlier to interrogate the devices on your network to ascertain TCP/IP characteristics such as packet jitter, loss, and delay. These characteristics can affect application response timing across the network both within the LAN and extended throughout your company WAN.

These tasks lead directly into capacity planning. Once you have an understanding of the performance baseline of your network, you can begin to plan for current and future capacity. Most networks are initially architected with consideration for supported applications. But the growth of business invariably adds application support requirements to the network over time. This upward trend in support requirements will add a burden to the network over time, and only through effective capacity management and planning will correct purchasing decisions be made to scale the network. Later, this chapter will discuss some useful metrics that your NMS can monitor and archive that will give you the objective ammunition you will need to justify future network enhancement requests.

Monitoring Deviations

Once you have an understanding of the nominal performance of your network, you can then begin configuring your NMS tool to watch for changes in that performance. This performance monitoring is often done through the use of network probes or network probing. With network probes, hardware devices are installed in-line with devices. These agents or devices report back to the NMS on the traffic characteristics going through the probe. Conversely, with network probing, a centralized device—typically the NMS itself—interrogates each device on a regular interval to pull counter information off that device.

Either technique for gathering data can provide the necessary information you need to monitor for performance deviations in your network. There are some factors in using probes that you will need to consider—such as the need to actually install and administer the probe as well as ensuring that the presence of the probe itself doesn't interrupt the flow of traffic around the network. This additional administrative overhead may drive you towards an agent-less solution.

In either configuration, an effective NMS will provide the capability of regularly gathering data characteristics and alerting when those characteristics exceed administrator-determined thresholds. Similar to how faults are detected, SNMP and SNMP traps are typically used by the NMS to handle configuration and notification associated with those characteristics. Some examples of interesting characteristics are:

- Input/output bits/second
- Current/average response time
- Peak traffic load
- Interface errors/discards
- Percent packet loss

What differentiates a good NMS from an inferior one is the capability to store performance characteristics in a searchable database. This functionality allows the administrator to effectively “go backwards in time” to compare performance characteristics from today's network with those of yesterday or last year. Because network performance issues may not necessarily involve noticeable spikes in traffic, this provides the administrator the ability to do long-term trending analysis on performance. If the use of your network is slowly increasing over time, you can monitor that use over many months to see where your capacity planning needs lie.

Performance Reporting

Obviously, without any reporting capabilities, all this monitoring and database storage of performance results isn't useful. So, an NMS must have a robust reporting engine that can align statistics with the network map and across multiple devices.

When looking for deviations in performance across the network, it is often useful to review graphs of inbound and outbound traffic from network device interfaces. In Figure 2.3, the graph shows the traffic coming from one device's inbound and outbound interface over a period of time. Graphs like this are created by the NMS at regular intervals and can be used to watch an interface for overuse or underuse.

In Figure 2.3, the X-axis of the graph shows the time of day, while the Y-axis shows the traffic in bits/second coming from the interface. These types of graphs are useful for analyzing short-term traffic patterns because you can see that both your outgoing and incoming traffic shows a slight uptick during the middle part of the day. This can be important in finding the source of a performance slowdown based on the time of day. In this example, depending on the capabilities of your NMS, you might be able to drill down further to see what types of traffic increase during this time period. Maybe users are doing their daily Web surfing during their lunch hour and this extra stress on the network is causing network applications to slow.

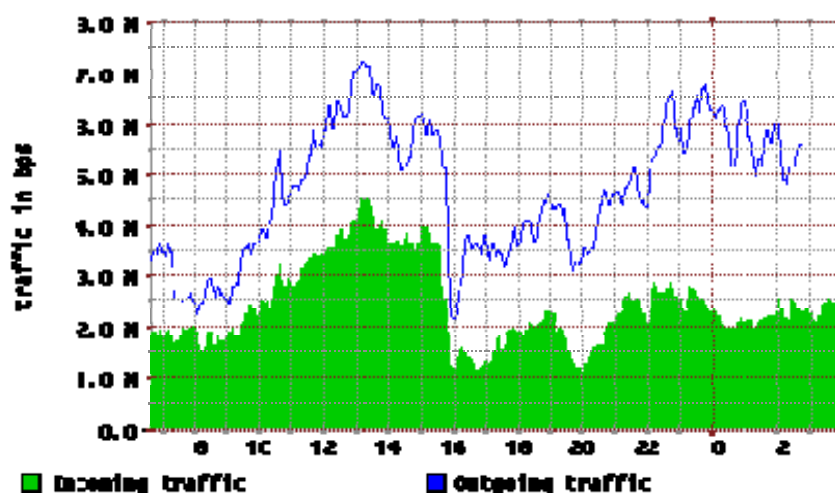


Figure 2.3: Network graphs provide a handy way to identify short-term traffic trends.

It is important to note that these sorts of traffic graphs are usually good for short-term and medium-term analysis. However, long-term analysis can be difficult due to the impact of non-business hours. When users are only using the network from 8:00am to 5:00pm but the traffic analysis graph monitors throughout the entire 24-hour cycle, compounding errors from those unused time periods can cause averaging effects in long-term graphs. These averaging effects have the tendency to reduce the graphed total network use in long-term graphs.

Performance Correction

As with fault management, the last step in performance management is actually fixing the problem. Within the performance correction step, you use the information given to you by the NMS, the graphs and trending analysis done in concert with the data in your NMS, and your own network intuition to determine that a change is necessary to bring the network back to nominal performance.

That change can take a number of forms. Adding extra interfaces to key network devices can distribute the load and increase performance. Depending on the type of device and load-balancing capabilities, performance can increase linearly with the number of additional interfaces added to the problem.

In some cases, especially those that involve problems over WAN connections, a change to the routing configuration can solve the problem. Often, though, these WAN-based routing configurations involve cooperation with the carrier provider hosting the WAN. In either case, rerouting traffic away from network hot spots to links with lower utilization can improve performance.

Lastly, in some cases in which performance is suffering substantially, the graphing and trending analysis can show that a complete network or network segment re-architecture may be necessary. In this case, movement of clients and servers closer within the network can increase their throughput. New devices with new technologies can increase performance or add compression or optimization capabilities. Conversion or encapsulation of client-to-server traffic within other protocols can improve performance by changing the method of access completely.



Not all network performance problems occur at the OSI model's layers 1, 2, and 3. Your performance problem may have nothing to do with physical connectivity or problems with TCP or IP routing at all. Some performance problems have to do with higher-level protocols and their tendency for "chattiness."

Databases and their associated application servers are a great example of this. If you separate a database from its application server over the network, you will often see a substantial performance loss. Consider correcting this problem by relocating the servers closer in terms of network proximity or by using other high-level protocols (such as RDP, ICA, SSH, and so on) to encapsulate the user's connection to their data.

Key Measurements in Performance Management

To properly undertake a performance analysis, you must have an agreed-upon yardstick of measurements that can quantitatively describe the qualitative behaviors on the network. This section will discuss five of these key measurements. The first two, bandwidth utilization and network latency, characterize the movement of data across the network from source to destination. The last three describe the state of the network device and its internal resources as that traffic moves in and out of the device.

Bandwidth Utilization

Before talking about bandwidth, it is important to dispel one common fallacy. The concept of bandwidth utilization is quite possibly one of the least-understood measurements in networking. This is the case because bandwidth and the use of bandwidth is actually not what most people really believe it to be. This comment is best explained through a comparison.

Bandwidth is defined as the amount of information that is physically possible to send through a particular media. The only people who can really talk about bandwidth are the physicists in the room because bandwidth is a measurement of the theoretical maximum capability of a particular network medium. Also, Ethernet is a baseband technology, which means each transmission fully utilizes all the available bandwidth. That is, all computers communicate at the transmission speed of the connecting medium.

A much more accurate description for the type of measurement we mean when we say “bandwidth” is “throughput.” Throughput is used to describe the measure of how much actual data could be sent over that media in a unit of time in the real world. Although this differentiation exists, in the real world, bandwidth and throughput are often used interchangeably, which is why it’s important to highlight the difference.



Bandwidth is equivalent to a medium’s theoretical maximum and throughput is equivalent to that medium’s real-world maximum, so you can consider them to be proportionally related.

No matter what the verbiage, when you think of bandwidth, think of it as the width of the pipe down which you’re trying to send data. The utilization of that pipe is the measurement of how full the pipe is. If the utilization of a 100Mbps pipe is 55Mbps, then you’ve got just about half the pipe left through which to send data.

Bandwidth in performance management is important for network architecture and capacity planning. When designing a network, you must ensure that the bandwidth across every node in the thread is capable of handling the load it will be assigned. As the utilization of available bandwidth grows to near 100%, the amount of time it takes to get data through that pipe increases because the data has to wait for the pipe to empty before it can start on its journey.

Network Latency

This concept of time delay in data transmission segues perfectly into the idea of network latency. Bandwidth gets a lot of attention because of the word's heavy use in the consumer market, but it could be argued that network latency more often than not causes the real problem in a network. In many business networks with 1Gbps links, utilization rarely goes above 10%. However, the time it takes for the data to traverse the network can range from less than one millisecond for a LAN connection to hundreds of milliseconds for a satellite connection to another continent.

```
C:\>ping 65.254.250.110

Pinging 65.254.250.110 with 32 bytes of data:

Reply from 65.254.250.110: bytes=32 time=80ms TTL=237
Reply from 65.254.250.110: bytes=32 time=80ms TTL=237
Reply from 65.254.250.110: bytes=32 time=80ms TTL=237
Reply from 65.254.250.110: bytes=32 time=80ms TTL=237

Ping statistics for 65.254.250.110:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 80ms, Maximum = 80ms, Average = 80ms

C:\>
```

Listing 2.1: *In Microsoft Windows, a rough estimate of network latency can be obtained by pinging a remote host and reviewing its round-trip time.*

As Listing 2.1 shows, you can use the PING command in most OSs to send a series of packets to a remote host and measure the amount of time it takes for that packet and its associated acknowledgement to complete a round trip back to the originating host.

Network latency is a good measurement of performance because it can be related to a network link's utilization and is much more easily measured. When a link's utilization is very high (for example, the pipe is full), network latency can be similarly very high (for example, data is waiting). One good measurement for identifying the performance on a network is to measure the latency between a client and its application server. Each device and link between devices in the thread between client and server (also known as a "hop") will add to the latency. Your performance baseline should recognize what are acceptable latency measurements, and your NMS should monitor for when that latency goes above acceptable measurements.

Interface Errors and Discards

Interface errors can occur when a problem exists on the network, such as a bad cable, line noise, or a malfunctioning device. Although interface errors are usually used as a metric for detecting faults on a network, their presence on a network device can help deduce an unexplained slowdown in network traffic. When interface errors occur, the traffic that is lost or corrupted in its transition to that interface on the network device needs to be re-requested and retransmitted. In situations in which substantial retransmission occurs, a significant reduction in performance can occur—even to the point of going beyond timeout conditions. Your NMS should monitor for interface errors and alert on situations when interface errors on an interface of a device exceed zero.

Network discards are rarer and are not necessarily always the product of a network problem. Discards occur when the policy of a network device instructs the device to ignore the traffic coming in on that interface. This happens often when device policies or Quality of Service (QoS) policies are enabled on the interface. In these situations, if the traffic is legitimate and you intend for it to be routed, the device should be reconfigured not to discard the traffic. Remember that some traffic likely should be discarded based on the policies assigned to the network device. In any case, similar to errors, NMS monitoring of discards will help identify when this situation occurs.

```
router1#show int
Ethernet0 is up, line protocol is up

[...lines removed...]

 11809 packets input, 933204 bytes, 0 no buffer
Received 8612 broadcasts, 0 runts, 0 giants
 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
 0 input packets with dribble condition detected
 7737 packets output, 651486 bytes, 0 underruns
 0 output errors, 0 collisions, 1 interface resets, 0 restarts
 0 output buffer failures, 0 output buffers swapped out
```

Listing 2.2: Some of the interface statistics on a Cisco router will provide information about interface errors.

Network Hardware Resource Utilization

Like any computer, a network device is a data processing device. And, as with any data processing device, there are hardware resources onboard a network device that it needs to do its job. If those hardware resources are being overutilized, the network device will not be able to quickly process network traffic.

Statistics on resource use are available for most network devices and your NMS should monitor for their overuse. During nominal conditions, network devices do not typically incur substantial processing requirements except in exceptionally high-use conditions. Thus, if a network resource begins to incur substantial resource use, it is likely that there is a problem on the network. That problem can be related to a device with debug logging enabled at too high a level, an overactive inbound interface, or a security situation involving some hacking attempt.

There are three very important statistics on network devices to monitor in terms of hardware resource utilization: CPU load, memory usage, and buffer usage. The next section will discuss buffer usage; the other two are both important counters because of their direct impact on that device's ability to perform its function. Your NMS should provide the capability to display monitoring information regarding device CPU and memory utilization and alert based on overuse of those resources.

Buffer Usage

Network devices typically incorporate a series of memory blocks to be used when transferring data between the internal components of the device. If an interface needs to send a packet of data to a routing processor, it must first reserve a buffer location to store the data packet. Failures with this buffer request and assignment process can be one of the biggest factors in packet drops, which lead to retransmissions and ultimately network performance loss.

A number of metrics of buffer use can be monitored and managed through the NMS. These metrics include:

- Total buffer number
- Number of permanent buffers
- Number of buffers in free list
- Buffer hits/misses/trim
- Buffer failures

Although buffer tuning is not normally a necessary task with network devices, knowing the status of available buffers and their hit and miss rate can help identify the source of network performance lag. Like the other metrics discussed in this section, your NMS should have the capability to interrogate and report on buffer metrics.

```
router1#show buffers

Buffer elements:
  500 in free list (500 max allowed)
  2370 hits, 0 misses, 0 created

Public buffer pools:
Small buffers, 112 bytes (total 16, permanent 10):
  12 in free list (0 min, 10 max allowed)
  1770 hits, 33 misses, 22 trims, 28 created
  8 failures (0 no memory)
Middle buffers, 600 bytes (total 90, permanent 90):
  90 in free list (10 min, 200 max allowed)
  595 hits, 0 misses, 0 trims, 0 created
  2 failures (0 no memory)
```

Listing 2.3: The Cisco `show buffers` command can provide information about buffer status.

The Business Metrics of Performance Management

A 20% reduction in the response time of a critical network application can potentially lead to a 20% reduction in the productivity of the worker using that application. Understanding the interrelatedness between network application performance and its affect on worker efficiency is critical to business. Conversely, there can be an equal loss in IT worker performance if those workers are constantly tracking down “the network is slow” comments. For a business to get the most out of the production network, a few tried-and-true metrics have been developed that help the technology people in IT relate to the dollars-and-cents people in business. This section will discuss a few of the metrics you can incorporate that leverage the monitoring capabilities of your NMS to provide useful reports to your business leaders.

Availability and SLAs

The concept of availability can be defined as the ability for workers to access the data and applications they need to accomplish their daily tasks. As Chapter 1 discussed, availability metrics are usually measured in whole-system terms: Does the user have access? Is the network up? As most modern networks incorporate some form of redundancy in order to increase availability, business is typically less interested in individual interface outages except when those outages affect that fundamental question.

Service Level Agreements (SLAs) can be created as contracts between IT and the business that outline the performance and availability requirements enforced upon IT by the needs of the business. These contracts establish, among other things, minimum acceptable performance metrics for applications on the network. Creating an SLA means that you and the business have agreed upon set standards for application performance. They also give you ammunition for identifying when hardware purchases are necessary to meet that SLA and for setting the quantitative standard for combating those “the network is slow” comments from your users.

One other major area in which SLAs are used is within purchase decisions of network carrier providers. When going through purchase decisions for WAN connectivity over commodity networks, SLAs with the carrier provider are necessary to ensure your nominal performance of the WAN link. Provider carriers are notorious for holding their customers to the precise verbiage of the SLA contract. This is due to the substantial cost of chargebacks to the customer when the provider's performance breaks SLA guidelines.



SLAs are contracts. Thus, the verbiage in the SLA is held to a high standard. If you require an SLA from your business or (more importantly) a carrier provider, check the fine-print definitions and reimbursement conditions very carefully.

Because of this, be aware in contract negotiations of those contractual metrics you want to enforce on your carrier provider. Table 2.1 provides a list of network performance metrics that can be defined within an SLA as well as the threshold for breaking that metric. For each line item in the table, a typical SLA will also specify what the cost or amount of the chargeback will be to the customer if the threshold is exceeded.

Metric	Threshold
Availability	99.9% uptime—equivalent to 8.7 hours of downtime per year
Mean Time to Restore	4 hours
Committed Information Rate (CIR)	512Mbps (burstable to 1Gbps)
Latency	< 50ms

Table 2.1: Example SLA metrics with a carrier provider and the associated thresholds.

Bandwidth Monitoring

With some notable exceptions, utilizing WAN connections through network carrier providers rarely means a direct point-to-point connection. Frame Relay connections are a particular type of WAN connection through the carrier provider's network that can bounce over numerous hops while within the provider's network. The provider does not provide a specific connection from one site to another. Instead, they provide a guarantee that the traffic will exit "the cloud" within a predetermined amount of time.

These types of connections are excellent for SMBs and businesses in the mid-market because dedicated point-to-point connections are expensive. By using Frame Relay connections, the connection from your home office to your remote offices goes into your provider's internal "cloud" once it leaves your internal network. You are not responsible for maintaining a direct line from office to office. Through SLAs and other contracts, your network carrier affirms a particular performance for those connections.

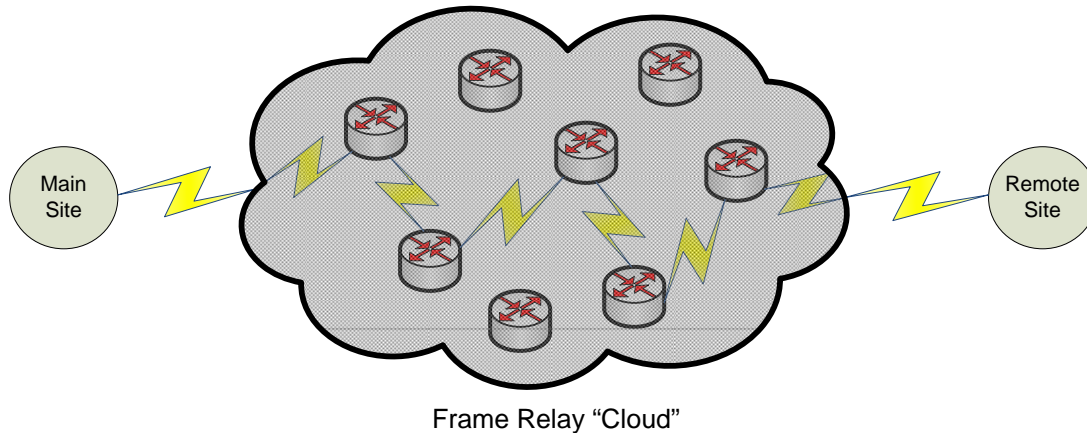


Figure 2.4: Inside the carrier provider's Frame Relay "cloud," your main site traffic can route through any number of hops before it exits at your remote site.

However, it is your money buying that link. Thus, it's often up to you to manage the monitoring of that link's bandwidth and other performance characteristics. In some contracts with carrier providers, the onus of notification is laid on the customer to notify the provider when the link is not performing.

In this case, your NMS can also assist with measuring bandwidth and latency metrics as well as notifying when a link drops. Bandwidth can be approximated by measuring the amount of time taken to move a data structure of a known size from one end of the pipe to another. Latency can be measured by round-trip time for ping packets. With an effective NMS, measurements are available out of the box to measure bandwidth characteristics across Frame Relay connections.

Whenever using carrier providers, consider configuring your NMS to regularly monitor the state of network connections and beyond just up/down notification. As the responsibility can be upon you the customer to notify (and receive the appropriate chargebacks) when the carrier's connection degrades, configure bandwidth monitoring to alert when conditions are out of specifications.

Link Costs

Link costs are a network term that describes the relative network "cost" associated with sending a data packet across that link. Network routing protocols such as Open Shortest Path First (OSPF) utilize link costs to make routing decisions when given redundancy options. Each link in a network is assigned a number whose absolute value is meaningless but whose proportional value in comparison with all other link costs in the network determines how traffic is routed.

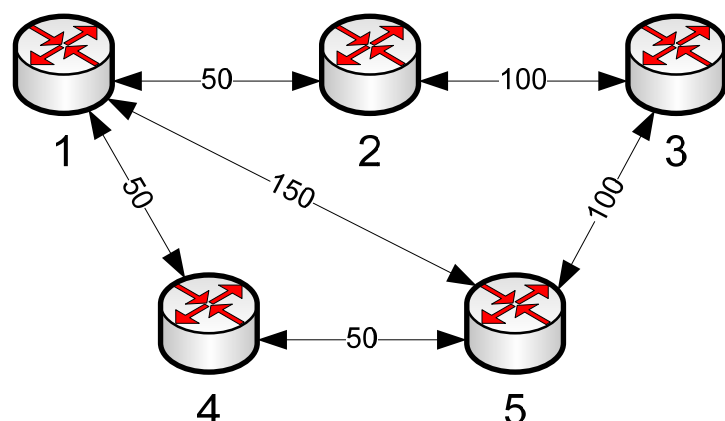


Figure 2.5: Each connection between routers is assigned a link cost. That cost is used to determine the least costly route to move data across the network.

The best way to illustrate this concept is with a picture. In Figure 2.5, for data to move from Router 1 to Router 3, it has three possible paths:

- Option 1: Router 1 → Router 2 → Router 3 = 150
- Option 2: Router 1 → Router 4 → Router 5 → Router 3 = 200
- Option 3: Router 1 → Router 5 → Router 3 = 250

Option 1 has a link cost of 150 added across the two hops from Router 1 to Router 2 to Router 3. This is the lowest of the possible routes the packet can take without retracing its steps across a hop. Thus, the traffic will default to Option 1.

Link costs are useful to monitor when redundancy options are added to a network. As you can see from Figure 2.5, if the connection from Router 1 to Router 2 goes down, it is still possible to send traffic to Router 3, hopping through an alternative path. Although Option 2 involves more hops, it will become the backup path because its link cost is lower than Option 3.

Link costs are important to monitor and maintain as business metrics because the link costs should be relevant to business application performance during non-optimal conditions. If your routing is configured such that a failure causes application performance to go beyond acceptable thresholds, a network rearchitecture or rerouting analysis may be necessary.

Traffic Management and Prioritization

The last business metrics that should be monitored for performance management reasons are traffic and prioritization. The Internet is a great place for research and for communicating with others in your industry, but it's also a great place for streaming music and video and overactive Web surfing. All this activity can impact the performance of your network connection to the outside world.

As noted during the performance graphing discussion, it is reasonable to assume that some Internet browsing by employees will occur during the lunch hour (unless you have a highly restrictive policy against employee surfing!). However, overuse of your corporate network for employee Internet browsing can have a detrimental effect on your network performance.

The concepts of traffic management and prioritization go hand in hand. Traffic management is the idea of implementing a policy-based approach to network traffic, granting or denying that traffic based on centralized rules. Traffic prioritization takes that one step further by allowing network devices to prioritize important traffic (for example, critical database to critical application server) over that of non-important traffic.

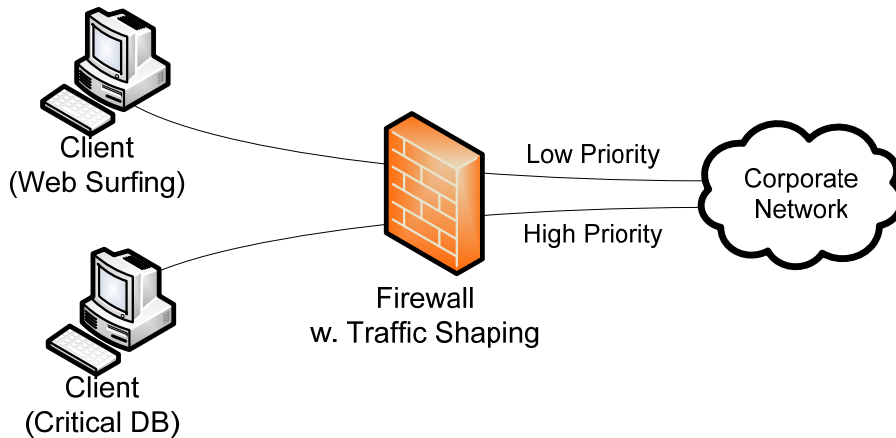


Figure 2.6: Firewalls and other network devices that incorporate traffic shaping can prioritize critical traffic over non-critical traffic within a network.

Both mechanisms require some form of management tool that enables policy-based management and traffic shaping to monitor, validate, and route traffic correctly. But in networks constrained by budgetary restrictions, enabling traffic management and prioritization can proactively reduce the productivity loss associated with misuse of available network resources.

Additional Tools for Managing Performance

Throughout, this chapter has discussed a number of tools—both procedural and technical—that enable you the network administrator to better manage the performance of your network. This section will discuss three additional tools that help in specific situations: Simulating loads, analyzing traffic patterns, and monitoring wireless communications.

Traffic-Generation Tools

In some situations, the only way to effectively identify when network performance will go critical is to simulate load on that network link. Traffic-generation tools are used for just that purpose. They generate an abundance of traffic to route over a configured connection. This administrator-configurable traffic can be adjusted to simulate increasing load over a connection. While the load is manipulated, application response time is judged during each phase in the traffic loading.

As with any network traffic, these tools must have a host on the receiving end configured to accept the incoming traffic. Typically, traffic from these tools is sent to destination port udp/9, which is configured on UNIX systems as a “discard port.” All traffic routed to udp/9 is read by the destination network interface and immediately discarded. This concept of a discard port allows for the successful receipt of traffic without overwhelming the destination system with the volume of traffic being generated by the source.

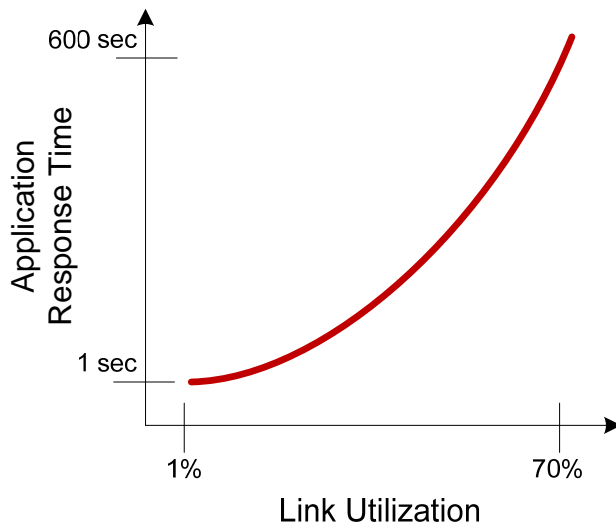


Figure 2.7: Increasing link utilization can have a negative effect on application response time. Traffic-generation tools are used to simulate this link utilization.

Traffic-generation tools are often components of the network toolkits that accompany NMS software packages. To successfully simulate a number of conditions, they typically include the following configurable characteristics:

- Target name or IP address
- Port number, UDP or TCP
- Packet size
- Percentage of circuit bandwidth to consume



Because traffic-generation tools have the capability to generate substantial amounts of traffic, they can be misused as tools to generate Denial of Service (DoS) attacks.

Traffic-Analysis Tools

Traffic-analysis tools are those that allow for collection of performance data and graphical analysis of that data across multiple network devices. These tools are used during performance baselining and performance reporting activities to gather an understanding of the underlying traffic on the network, its type and time of day, and its source and destination.

Effective traffic-analysis tools will be highly graphically oriented to provide the administrator with charts and tables representing a global view of the network. These charts and tables typically have drill-down capabilities to provide the administrator with more detailed information about a particular traffic flow. As the types and network protocols involved with network traffic are many and complicated, a good traffic-analysis tool will also be able to identify within each traffic flow the protocol type and graphically display those types for analysis.

There are generally two types of mechanisms for analyzing traffic across a particular network connection. The first involves the incorporation of a network probe device that is installed in-line between two network devices. The network probe monitors the traffic between the two devices and reports what it sees to the NMS. The second uses a technology developed by Cisco called NetFlow. Built-in to many network devices, the Cisco NetFlow technology eliminates the need to incorporate an in-line device. Instead, the traffic is monitored at the interface of each of the NetFlow-enabled devices. Using NetFlow reduces the administrative overhead of installing, managing, and ultimately removing network probes.

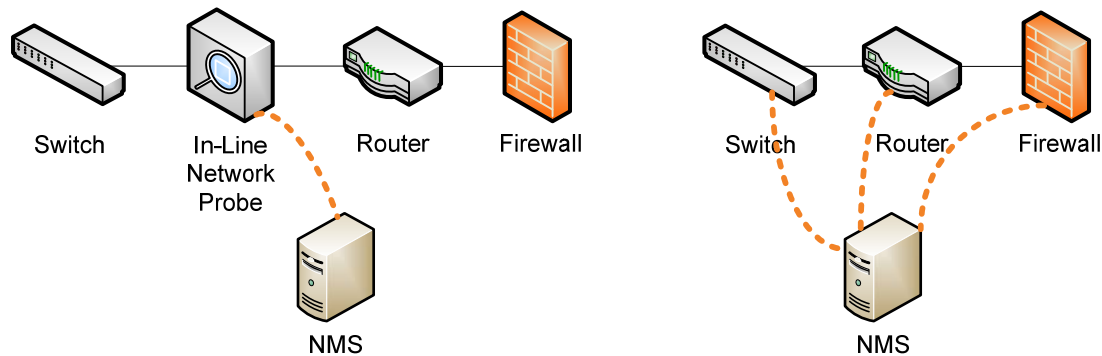


Figure 2.8: On the left, an in-line network probe sends SNMP performance data to an NMS. On the right, an NMS communicates with Cisco NetFlow-enabled devices to probe the internal performance counters on each device.

No matter which type is used, traffic-analysis tools complement the aforementioned traffic-prioritization tools that identify and prioritize traffic types based on administrator-set policies. These tools enhance mere analysis by allowing administrators to actively fix overactive protocols on their networks. Characteristics that can be analyzed by these types of tools include:

- Global view of consumption
- Consumption over time
- Consumption by hour of day
- Application consumption patterning
- Type of traffic
- Traffic data archival
- Historical analysis
- External traffic sourcing

Wireless Performance Tools

Wireless networks have their own special needs—from the additional security needs that ensure hackers can neither see nor access internal networks to the location-based complexities of dealing with a network intended to operate through walls. These special needs require special administrative tools. Outside the typical feature sets associated with NMS' and traffic analysis tools, wireless performance tools typically provide additional features that enable access point signal strength metering, mapping of the network to identify “hot” and “cold” spots, and identification and analysis of clients and sessions.

Arguably, the most difficult part of wireless administration is the management of the Wireless Access Points (WAPs) in a network. Because wireless is designed so that users can roam from any WAP to any other WAP, a good wireless performance tool will provide information about dropped network packets, roaming patterns, and active devices. As devices move throughout the network, the administrator will likely want to identify clients, their attached WAP, and their signal strength to that WAP. Doing so will assist the administrator with identifying where building features may be interfering with the wireless signal. Wireless typically operates at a much lower bandwidth than wired connections, so good tools also can show the administrator where bandwidth contention is occurring between clients on the same WAP.

Performance Affects Business

As discussed earlier in this chapter, a 20% reduction in the response time of a critical application can potentially lead to a 20% reduction in workers' productivity. That 20% can mean the difference between an agile, successful business and one that is unable to keep up with the needs of its customers. As you can see throughout this chapter, there are numerous tools available that can provide a quantitative solution to what has historically been a qualitative problem—“the network is slow.” With these tools in place, you can proactively identify the slow spots on the network and begin implementing solutions before that call occurs.

The next chapter moves away from problem management and focuses specifically on the tenants of configuration management. Chapter 3 talks about how you can employ an NMS that can ensure a consistent configuration across your network devices. This consistent configuration will ensure your adherence to compliance regulations as well as ensure a consistent and repeatable configuration to all the devices on your network.

Chapter 3: Configuration Management and Security

The average mid-size company of 250 employees typically serves the same number of workstations with about 25 servers. Those nodes on the network are interconnected by about nine network devices, through firewalls, switches, and routers. For a network of that size, the average network device configuration contains about 300 lines per device. Multiplying those two numbers, you get the potential for more than 2700 individual configurations, just to connect a relatively small number of devices.

The big question is this: In a critical situation, could you rebuild those 2700 lines purely from memory? It's the implementation of configuration management into your network environment that helps you answer that question in the affirmative.

Chapter 2 discussed performance management in relation to the FCAPS model of network management. The chapter discussed how managing performance in a network can be virtually impossible without a baseline to measure it by, and talked about how to use your NMS to measure the changes in performance from your baseline and how those changes in performance can trace back to configuration inconsistencies or other underlying problems. The chapter also brought forward some good technical and business metrics that illustrate network performance and validate it to your business leaders.

This chapter will move away from the P in FCAPS and focus on the C and the S—configuration management and security. This chapter will discuss how you can use a good NMS to consistently manage, store, and audit the configuration of devices on your network. We'll explore the four steps in establishing and maintaining an environment configuration and relate those to the underlying financial reasons why configuration management has business relevance. The chapter will go over a set of features that an effective NMS should incorporate to assist with this task, and will finish up with a short discussion about how good device configuration dovetails into good device security.

Key Steps in Managing an Environment Configuration

As businesses elevate through the growth cycle, they typically go through phases of network control. When the typical business starts, it usually hosts few employees, zero to few IT personnel, and a very small network presence. Because the network presence in these small businesses incorporates few devices, the management of those devices is relatively simple. Documentation and change control of these small networks is usually ad hoc. This is not because of laziness on the part of IT but has more to do with the priorities of the typical business startup.

As it grows, a business' internal processes become more refined. The business requires more network services to support its internal processes. As the number of network services mounts, network infrastructure improvements become necessary, and this natural increase in network infrastructure geometrically increases the number of network configurations. For nearly all businesses, priorities eventually shift away from the early ad hoc mode to the need for more stability and a slower change rate. It is this changeover from a request-driven mode to an architecture-driven mode that can cause stresses on burgeoning networks. Incorporating good configuration management at the right time is the critical success factor in preventing those stresses from negatively impacting the flow of the growing business.

Configuration management can take on many forms, and a number of frameworks such as the IT Information Library (ITIL) and FCAPS have been created by the industry to provide a tangible outline to work within. Distilling those frameworks into something useable by SMBs and those in the mid-market can be the most daunting part of incorporating formal change control. Let's make the process easy by breaking it down into four easy-to-understand steps. Similar to the graphic used in each of the previous chapters, let's use a variation of it to discuss the steps needed for implementing good configuration management as well as the tasks associated with those steps.

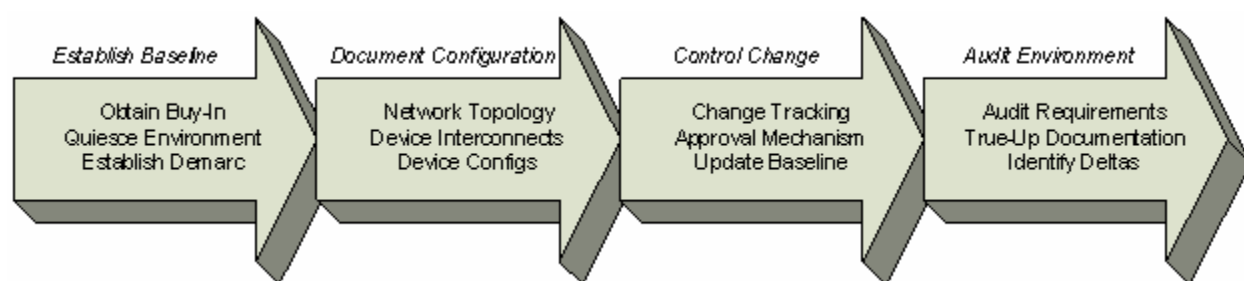


Figure 3.1: Network configuration management can be broken down into four easy-to-understand steps.

To take your network from one of ad hoc change control to one that incorporates good change management practices, you really only need these four basic steps. You'll first need to establish a configuration baseline, document that configuration, establish a mechanism for changing and updating the configuration, and put in place an auditing process to verify it. Let's talk about each of these steps in turn.

Establish a Baseline

Assuming your network is one that has not yet been baselined, the first step in implementing effective change control is to establish a baseline of your environment. This baseline is an understanding of the current configuration on the network and the interrelations between the devices that make up that configuration.

Even the smallest of networks involve a lot of change, so the first step in establishing a baseline is to quiesce, or quiet, the environment. This is not so much a technical step as one of an organizational “snapshot” of the environment past which all changes are logged until the documentation is completed and the change control processes established. This process of snapshotting the environment is a procedural one where a deadline date is established and all changes past that date are logged. This “line of demarcation” separates the previous ad hoc procedures from the new controlled ones.

Because creating the line of demarcation can have the potential of interrupting the normal operations of the network, it is imperative that business management buy-in is obtained prior to the activity. Their buy-in can be easily obtained by identifying the ROI associated with a healthily documented and controlled network. We’ll discuss some of the business metrics for establishing that ROI later in this chapter.



Of the four steps in the configuration management process, establishing the baseline involves little to no technology solution. Where your NMS can provide documentation automation is in the next step, where you document the configuration.

Document Configuration

Step one in this process is almost completely a procedural step within the company and does not often involve the use of technology. Once you’ve obtained buy-in from management on the desire to move towards proactive network management, you can incorporate the features of your technology solution to assist with the documentation of that configuration.

Looking back to the initial question of this chapter, network devices typically have hundreds of lines of code that define their configuration. This text-based method for configuring network devices makes them cumbersome to configure manually but makes them excellent for automatic configuration storage and archiving. If each of the network devices on your network is configured with little more than a text file, and that text file can be transferred over the network, your NMS should easily have the ability to store the configuration into a central database. Doing so should be the largest component of your configuration documentation.

A good NMS should also be able to inventory the network for individual devices and their interconnections. This inventory should drive the generation of a map of those devices and their connections. The combination of each device’s configuration file along with the map of its interconnections should fully document your network configuration. It is reasonable to assume that with a good NMS, each of these processes should be at least partially automated. An excellent NMS will incorporate full automation into this process, making this documentation step very easy.

Although historically much of this configuration capture has occurred through automated tools that utilize Telnet or SSH to log into the device, newer technologies now allow for full SNMP-based management of all devices. Moving from command-line technologies to SNMP allows for all facets of device management to be done through a single protocol, making administration much easier.

Control Change

The third step in managing the configuration of your environment is the establishment of a process to formally request, approve, and document changes to your configuration. This process of change control gives you a formal mechanism to ensure that changes are done correctly, that others in the IT organization and the business are notified of the changes, and that the changes get correctly reflected in your baseline. Whatever organization process you choose should ensure that the IT organization and/or business leaders of your environment are informed of the need for change and have a reasonable ability to either approve or reject that change.

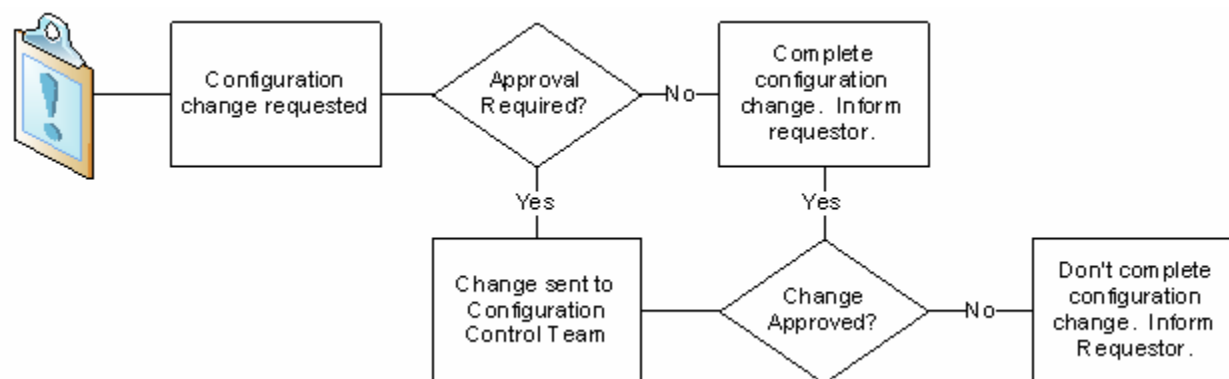


Figure 3.2: A simplistic formalized change control procedure.

From the perspective of your NMS, this change process should integrate with the configuration storage and archival process as well. Once you've stored your device configuration into your NMS, you need a process to update it after the change occurs. Some excellent NMS systems allow for the change to be made within the NMS and "pushed" out to the device through an update process. This feature allows the NMS to confirm the change and relate it to the configuration for other devices on the network and helps to reduce the chance that that change will negatively impact other devices on the network.

If the change involves an environment topology change, it must also integrate with the interconnection mapping capability of the NMS. This can be either through a re-scan of the devices on the network or an integrated update of the topology within the NMS prior to deployment.

Either solution ensures that change within the network is reviewed by others within the business, goes through an approval process, and is captured by the configuration management tool to ensure a one-to-one mapping between the actual configuration and the documented one.

Audit Environment

Lastly, any process for configuration management needs to include a process whereby that environment can be audited against its baseline. Whether network changes are done within the NMS automatically or done outside and synchronized with the NMS database, there are times when unapproved or inappropriate changes make their way into the network. It is the process of network auditing that validates your network's configuration and ensures that nothing inappropriate or unapproved has been done.

The auditing process is essentially the very same as the automatic documenting process discussed in step two, with the exception that your NMS should notify you when a network configuration doesn't match what is expected. This can either be done automatically and at regularly scheduled intervals, or it can be done as a manual or partially manual activity on a semi-regular basis.

Depending on the industry in which you do business, there may be one or more compliance regulations that require this auditing step to occur. Your ability to show successful compliance to network security regulations and prove your configuration can prevent you from expensive and damaging litigation.

 Later, this chapter will discuss more about compliance and compliance regulations.

Compliance Regulation	Industry
Sarbanes-Oxley Act (SOX)	Publicly-traded institutions
Gramm-Leach-Bliley Act (GLBA)	Financial institutions and those that handle personal financial information
Payment Card Industry Data Security Standard (PCI or PCI DSS)	Institutions that accept payment cards
Health Insurance Portability and Accountability Act (HIPAA)	Medical institutions

Table 3.1: Some compliance regulations that may have auditing requirements and the industries that must follow those regulations.

Ad-Hoc/Manual Configuration vs. Managed Configuration

Throughout, this guide has talked about how you can leverage a good NMS to move your style of network administration from reactive to proactive. Nowhere can you get more “bang for the buck” than in using your NMS for configuration management. This ability to standardize configurations, back them up in case of disaster, automatically restore them should they get corrupted, and automate many other network administration tasks eliminates much of the difficult and manual parts of being a network administrator. Removing these manual components allows administrators to focus time on more productive and value-added activities—such as performance management and reducing downtime.

Incorporating an effective NMS into your network will grant you these abilities. An effective NMS will have some, if not all, of the following feature sets to assist with this automation activity.

Configuration Standardization

With an average of 300 lines of code needed to configure a typical network device, it is reasonable to assume that over time small differences between devices can manifest. Ever tried to line up two files and analyze them line-by-line to verify they are identical? The process is painstaking and fraught with error. Standardizing on a framework for device configuration helps to reduce that error. Using an NMS to provide the framework and to automatically notify you when deltas occur between devices goes even further.

```
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname switch1
!
!
ip subnet-zero
!
vtp domain [smartports]
vtp mode transparent
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan 2
    name VLAN_2
!
vlan 3
    name VLAN_3
```

Listing 3.1: Network devices typically have very similar configurations; the differences appear within individual port network assignments. Standardizing on configurations reduces their complexity.

A good NMS will provide a suite of standardized device configuration templates that incorporate best practices for performance, readability, and security. By providing these templates, you need only “fill out the form” with the individual configuration. A good NMS will also provide the ability to do a side-by-side and line-by-line comparison between two devices to validate their similarities and differences. Standardizing on a single template for each network device type ensures that multiple administrators can manage the network with a minimum of administrator “personality” in coding and configuration style.

Configuration Backup and Archival

A configuration is only useful if it’s loaded onto the device and performing its function. This always holds true in a device’s running configuration but never holds true if the firmware of the device wipes clean. In those critical situations, it is often difficult to rebuild the configuration by hand. The stress of needy users and irritated business leaders can result in mistakes being made when they are needed the least.

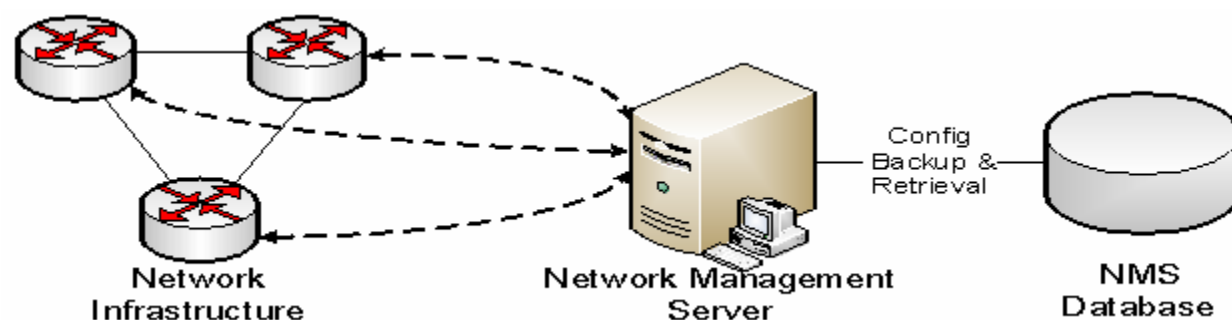


Figure 3.3: An NMS enables automated configuration backup, archival, and retrieval should a configuration become corrupted or problematic.

A good NMS will provide the ability to back up each device’s configuration files over the network and store those configurations into a centralized database. This database should have the ability to show and store multiple configurations over periods of time. This allows the administrator to rollback to a previous configuration if the situation warrants. The archival of configurations is additionally necessary to ensure that long-term backup of device configuration can be done should historical research be necessary by either administrators or outside auditors in the case of a compliance audit.

Post-Incident Restoration

Backing up your configurations is handy only if you have a mechanism for easily and quickly restoring them onto the correct device or its replacement. Once an event has occurred and the initial triage determining the source of the problem is complete, the best course of action in many critical down events is to “turn back the clock” and revert the device to its last-saved configuration.

In many cases—especially when good configuration control is in place—that last-saved configuration is equal to the configuration that was on the device prior to the outage. Restoration of the configuration onto the device will return it to service.

In some cases, however, the device may have been in a maintenance period where changes were being implemented or functionality testing was being performed. In those cases, either an administrative misconfiguration or a configuration mismatch or corruption caused the outage. Reverting the device to its last-saved configuration will bring the device back to a state where its functionality is known. The speed and ease of post-incident restoration is a critical determinant in choosing a good NMS for configuration management.

Policy-Based Configuration

As the number of devices on your network increases, you will begin to find that many devices are configured nearly the same across the network. It is this similarity in configuration that allows for the device configuration framework to be guided by corporate, network, and security policies.

The process to create a policy involves business and IT working together along with necessary compliance and security regulations. Often, an NMS can provide a known best practice as a starting point for creating device policy, such as one that automatically complies with SOX, HIPPA, or other regulations. This device policy becomes the framework in which all devices are ultimately configured.

By configuring devices according to an agreed-upon policy framework, a network now has the ability to quickly update all device configurations should that policy change. An effective NMS will offer the functionality to provide best practices in device policies and policy frameworks, and most importantly, incorporate the ability to deploy changes to that framework across all devices through an automated manner.

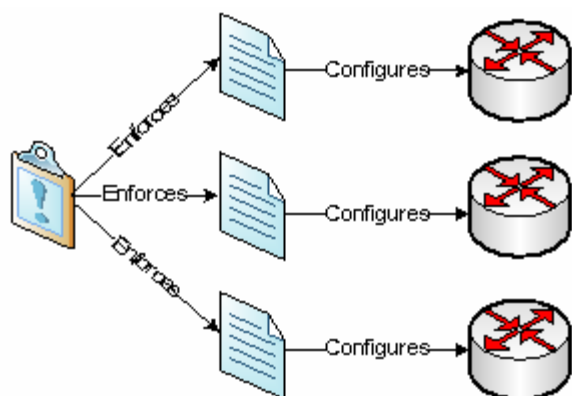


Figure 3.4: *With policy-based configuration, a single corporate policy can define and enforce multiple device configurations.*

As an example, presume that for performance reasons your business wants to prevent traffic associated with media streaming from traversing your network. A policy-based configuration will allow that denial to be enabled on all routing and/or switching devices in the environment. Should a change to corporate policy be later incorporated allowing streaming media traffic, that configuration can easily be updated on all devices vis-à-vis the policy. A configuration based on a policy framework means that those exclusions can be rolled out to the pertinent devices automatically with a minimum of manual intervention.

Inventory and Mapping

SNMP is a great protocol for identifying devices on the network. Combining SNMP with the routing and switching protocols that enable network devices to identify their closest neighbor enables a smart NMS to quickly build inventory and mapping data. Among other items, network inventory shows the administrator devices enabled on the network, their make and model, firmware version, network location, and connected hosts. As discussed earlier, the mapping component of your NMS should enable a graphical representation of the interconnections between devices on the network.

Rogue Device Identification and Adjudication

As a security function, your toolset should also allow for the capability to scan your internal network for known and rogue devices. Having the knowledge of the known devices on your network is good only for verifying their availability. However, being able to know when rogue or inappropriate devices appear on the network is critical for network security. Your NMS should include the ability to scan network ranges to look for devices that both should and should not be there as well as provide some limited information about the rogue devices and their configuration.

In some cases, the introduction of a rogue device onto the network is a harmful activity that should be immediately auctioned upon. Depending on the severity of the business security policy, a good NMS can be configured to either notify when that rogue device comes on the network or even go as far as to shut down the network port of its attached neighbors, isolating the rogue device from the rest of the network.

Provisioning

The ability to rapidly provision new devices onto the network improves the efficiency of the network administrator. Though small networks may not often add new devices into the network, there tends to be a related increase in device count as the number of new employees increases. Dovetailing with policy-based and standardized configuration capabilities, your NMS should have the ability to “cookie-cutter” those configurations while preserving individual device uniqueness.

This combination of standardized configuration along with device personalization ensures that each device on the network can be provisioned automatically. Just hook up the device to the network, enable its SNMP community strings, and instruct the NMS to download the correct configuration and begin regular configuration backup, monitoring, alerting, and auditing functions.

Deprovisioning

Although the provisioning activity is usually well done by good NMSs, where many fall short is in the ability to quickly deprovision them. The process of deprovisioning involves removing the configuration from the device and removing the device from the network. It also involves decisions associated with maintaining that device’s configuration in the NMS database and/or removal.

Where critical features exist are in device upgrade situations. When a device is upgraded either with a brand-new replacement or with an internal improvement or augmentation, a provisioning activity and a deprovisioning activity are rolled together. Your NMS should incorporate the ability to roll the configuration from the previous device to the new device while still maintaining the uniqueness characteristics of the new device. In the case of a device augmentation, the NMS should have the ability to roll the changes to the device IOS into its configuration.

User Access

Lastly, a granular and auditable user access policy should be a feature set on your NMS. Compliance regulations specify that users and user logons are to be segregated in such a way that users log on with individual accounts. They also specify that activity should be tracked into a database that cannot be manipulated by users other than the top-level administrator. This segregation of roles prevents collusion between administrators and reduces the chance that a single administrator can take down the entire network through his or her actions alone.



Through all the aforementioned features, you’ll notice a centralization of control to the NMS. This centralization of control has the effect of “putting all your eggs into one basket.” Thus, if not properly secured, it could be hacked by an aggravated administrator with a desire to damage the network. Every network device can be managed and controlled by the NMS, so the administrator’s malicious intent could indeed do major damage to each network device.

An effective NMS will have the ability to segregate administrators into roles and assign tasks to those roles. The lead administrator that is ultimately in charge of the entire system can assign users to roles and tasks, preventing any one user from gaining too much access into the system. Not having this distribution of work share can often cause problems on a compliance audit.

Business Drivers for Configuration Management

Moving your network administration style from reactive to proactive provides a lot of benefit to the business as well. It allows the administrator to better understand changes in the network prior to implementing them. And reducing firefighting often frees up enough time that an otherwise harried administrator can spend more time focusing on ways to prevent problems from ever happening.

Some of the business metrics that can be improved by the movement from reactive to proactive are described in the following sections. Each directly benefits by the changeover to formalized and automated network configuration management.



Metrics that define an IT person's worth are often driven by Number of Closed Tickets or Number of Resolved Problems. Unfortunately, driving an administrator's measurement of success by the number of problems fixed is kind of like paying a programmer by the number of lines of code they can write. A lazy programmer incentivized in this way will spend less time writing efficient algorithms and more time pushing out unoptimized code. An unscrupulous one may even artificially inflate the algorithm line count if they're behind on the house payment. Finding efficient and correct incentives for network administrators is just as critical.

MTTR Reduction

Chapter 1 discusses MTTR, which is a metric associated with the Mean-Time To Restore a particular system. If a company's MTTR requirements are measured in minutes, an administrator has only a very small amount of time to get a failed system up and operational. If network device configurations are backed up and can be quickly restored to failed devices or replacement devices, this significantly reduces the MTTR during a failure event.

Your MTTR requirement should be driven by your business needs. A short MTTR will drive the change to a tightly controlled environment.

Loss of Business Revenue

The MTTR conversation links directly to loss of business revenue. When a network device goes down during the business day, customers are unable to do business with the company and employee productivity reduces to zero. That reduction in productivity has a direct bearing on the company's bottom line.

Implementation of an NMS that can automatically triage a down device, notify an administrator, and provide suggestions for remediation all increase the speed of recovery and reduce the accompanied loss of revenue associated with an outage event.

Security

Security is a critical component of all networks. However, in most networks, the majority of the security exists at the perimeter. Inside the LAN, security controls can be lax. This has the tendency of creating a security profile with a solid outer shell and a soft inside. This typically happens in networks because of the complexity of enabling and implementing centralized security controls, access control, and a cohesive security policy.

With the incorporation of policy-based configurations on network devices, the security profile for a network can be enhanced. This is especially so when the policies that drive each device configuration are generated from known security best practices. When considering the ROI associated with an NMS purchase, consider one that provides such best practices right out of the box to augment the security of your soft internal network.

Regulation and Compliance

Depending on the industry in which you do business, there may be governmental or other regulations that drive a certain security and auditing stance within your organization. For most of those regulations, the concepts of role separation, policy enforcement and follow-up auditing, and securing of data during storage and transport are of great importance.

For an example of how that can impact network operations, consider the PCI DSS standards used by vendors or service providers that handle, transmit, store, or process information using payment cards. These standards require a predetermined level of internal security for any networks that transmit personal financial information. Depending on the number of payment card transactions your business does in a year, that level of security increases. Your business can incur costly and damaging litigation should an information disclosure event on your network result in the loss of that personal financial information to an outside party. By ensuring a stable and auditable configuration and security profile, you fulfill the PCI DSS along with other compliance regulations while reducing your liability.

PCI DSS High-Level Requirement	Affected Through Effective Network Device Change Management Leveraging a Good NMS
Install and maintain a firewall configuration to protect cardholder data	Yes
Do not use vendor-supplied defaults for system passwords and other security parameters	Yes, for network devices
Protect stored cardholder data	Yes, upon data transmission
Encrypt transmission of cardholder data across open, public networks	Yes
Use and regularly update antivirus software	Yes, on device IOS
Develop and maintain secure systems and applications	Yes, for network devices
Restrict access to cardholder data by business need-to-know	Yes, through process
Assign a unique ID to each person with computer access	Yes, for network devices
Restrict physical access to cardholder data	No
Track and monitor all access to network resources and cardholder data	Yes, for network devices
Regularly test security systems and processes	Partially, through auditing capabilities
Maintain a policy that addresses information security	Partially, through process

Table 3.2: The PCI DSS security regulation and how effective network device change management leveraging a good NMS can affect compliance with that regulation.

Auditing Requirements

Hand-in-hand with compliance regulations is the requirement to validate your configuration. Any change process that incorporates manual steps can introduce error into the result due to data entry errors, missed steps, or incorrect documentation. Because of this, a regular and out-of-band auditing activity needs to occur on your network. Depending on the features associated with your NMS, that activity can either be highly manual or highly automated.

With highly manual activities, additional error is introduced as the administrator reviews the logs and configurations and completes the true-up between the documented and the actual configuration. Additionally, the cost in actual dollars and administrator time to complete the activity can be prohibitive or even prevent it from actually occurring.

With a highly automated activity, the process of comparing the actual and archived configurations can be done with very little effort by the administrator. Thus, what may have normally been an annual activity can be configured to run daily or hourly. This enhances the network's security profile and ensures that inappropriate or unapproved changes are immediately identified and adjudicated.

Personnel Turnover

In terms of loss associated with inadequate security, it is well known that the people inside the company are the biggest risk. Personnel turnover, especially in cases of a terminated employee, have the chance of interrupting network operations should that employee leave behind backdoors into the network or cause damage prior to departure.

In an environment that incorporates centralization of user accounts and passwords along with role-based security attached to each username, it is less likely that a terminated employee can cause this sort of damage. When all network devices look to a centralized location for their identification, authentication, and access privileges, the elimination of privileges can be done very easily and from a single point rather than requiring an emergency password change on all network devices. This has the effect of eliminating or reducing the problem associated with employee turnover.

Supporting Technologies

This chapter has discussed several feature sets that make up a good NMS. There are some additional supporting technologies that may be incorporated either within your NMS or through associated tool sets that help enable those features. Three that are discussed in this section are configuration analysis and comparison tools, task scheduling tools, and RADIUS/TACACS.

Configuration Analysis and Comparison Tools

Network devices are notoriously command-line driven due to their highly optimized architectures and a long-standing tradition towards text-based configuration. Doing configuration via command line is exceptional for experts and for doing batch manipulation of many devices, but it can be very cumbersome for non-experts or cross-network updates. In those cases, a graphical interface that retains the text-based nature of the text file yet enhances it with additional features can be a key tool.

These graphical “skins” for configuration files make an excellent addition to either your NMS or your network administrator’s tool set. Some features that these graphical tools provide are:

- Color-coding to differentiate configuration sections
- Side-by-side comparisons, also with relevant color-coding
- Support for multiple device vendors
- Real-time configuration change detection and alerting
- Bulk-update capability
- Management and administrator reporting
- Itemized configuration change history
- Device grouping
- Individual port status

Task Scheduling Tools

Many of the tools discussed are handy but only so when the administrator is sitting in front of his management workstation. Adding to these tools the capability to schedule reports, device backups and restoration, and configuration updates mean that administrators can move activities to known times in the future. The ability to configure when these tasks occur allow for regular activities to take place without needing to remind the administrator to accomplish the task. Consider the ability to schedule tasks a necessary tool for your tool set.

RADIUS/TACACS

Much of this discussion regarding security lies on the centralization of identification and authorization information away from the individual network device. Most network devices have the capability of setting up access based on username on the device. This works well when the number of devices is small and the users using those devices trust each other. However, in larger organizations with many devices and many administrators, the need for centralization of account authority grows with device count.

Remote Authentication Dial-In User Service (RADIUS) and Terminal Access Controller Access-Control System (TACACS) are two technologies that enable this centralization of access. Rather than creating “enable” passwords on each device, RADIUS/TACACS allows users to use individual accounts for login and enable rights.

As stated before, many compliance regulations require unique usernames and passwords for all administrators for tracking purposes. Thus, using the default per-device authentication mechanism may violate certain compliance regulations. Most network devices have the capability to authenticate against one of these external authentication databases.

```
!
aaa new-model
aaa authentication login default group RADIUS local
aaa authentication login CONSOLE local
username root privilege 15 secret MyP@ssword
!
no enable password
no enable secret
!
!
ip radius source-interface Fa0/0
!
radius-server host 10.0.0.1 auth-port 1645 acct-port 1646 key P@ssword2
!
Line console 0
  logging synchronous
  login authentication CONSOLE
!
line vty 0 15
  Privilege level 15
  login authentication default
!
```

Listing 3.2: An example of a device connection to a RADIUS server for centralized administration.

Understanding Security Management

Moving the conversation away from the C in FCAPS and focusing it on the S for a moment, let's briefly explore security management. Security management is a critical part of any network, and doing it correctly makes the difference between a hacked network and one that can survive external attack.

The focus of this discussion is on how your NMS can enable good security management in your network. An effective NMS will help you practice good security as well as notify when you when commonly known best practices are not being followed for the devices on your network. This section will focus on five places where your network architecture and your NMS can assist with creating a good security posture for your environment.

Practicing Good Network Security

Obviously, the most important facet of security is simply doing it correctly. A correct security posture for your network will drive fewer infiltration successes. Much of this starts with generating a cohesive security policy for your network. This is of critical importance because lacking a good security policy prevents enforcement of security procedures when situations do occur. That security policy should take into account factors such as:

- Password policies
- Acceptable use policies
- Lockdown and access policies
- Mobile device access and lockdown policies
- Business data encryption policies
- Antivirus, anti-spam, anti-malware, and anti-spyware policies
- Security policy violation adjudication procedures

The incorporation of a sound and cohesive security policy in your organization will drive the creation and enforcement of other policies and procedures that fulfill the need for security. From a very high level, the need for security should drive answers to these questions:

- Does network traffic route to its appropriate destination and nowhere else?
- Does only authorized traffic pass into and around the network?
- Does that traffic pass at a recognized and acceptable volume?
- Does data arrive intact without corruption and without interception by inappropriate internal or external entities?
- Does traffic that requires encryption traverse the network encrypted and with the correct level and strength of encryption?
- Are mechanisms in place to ensure malware and potentially unwanted software do not enter or traverse the network?

SNMP Community Strings and SNMP Weaknesses

As described earlier, SNMP is the major component of configuration enumeration and manipulation for most network devices. However, SNMP strings themselves have weaknesses that should be taken into account when deploying SNMP-based NMSs to manage them.

Unlike user accounts, which can be segregated to individual persons through the use of RADIUS/TACACS servers, SNMP community strings are often not unique for each device in the network. If you've taken the time to segregate your administrators and their logins, they may still have a backdoor into the network via the shared SNMP community string on each device. An SNMP community string is effectively a password into the system, especially when used to update device characteristics, so extra care must be incorporated to ensure their security.

One feature of a good NMS is the capability of rapidly changing SNMP community strings for all devices on the network during events such as administrator termination or suspected infiltration. SNMP community strings should be changed as part of a regular password change cycle. And, if possible, each device should use a different string. As each device can have as few as two strings for reading and writing data and potentially more, a manual update of these strings can be costly in terms of time. Automating this process ensures it is done regularly, further enhancing the security profile of a network.

Port Scanning and Port Minimization

Although port scanning from the Internet is usually considered a bad thing, port scanning by an approved administrator with good intentions in mind is a good thing. By completing port scans on internal hosts—and especially those connected to the Internet—the network administrator can ensure that only the necessary services are enabled and listening on a particular host.



This is especially a problem on Microsoft Windows servers, which tend to listen on numerous ports, some of which are of very high risk in untrusted network environments.

Port scanning is a feature of many NMSs and NMS tool sets. The port scan can identify on which ports a server is listening and may also provide some information about which service is listening on that port. Once the host is scanned, the network administrator can work with the systems administrator to shut down any unnecessary services. In a case in which the service cannot be disabled, the network administrator can modify firewall rules to prevent external entities from accessing the high-risk service. This back-and-forth process encapsulates the idea of port minimization.

Penetration Testing

Taking the concept of port scanning even further is the idea of penetration testing. Penetration testing can be done either internally or by an external entity such as a security consulting organization.

No-cost and for-cost tools are available that simulate dozens or hundreds of known exploits against a particular host, whether that host is a network device or a server. Pointing these tools towards the hosts and network devices in the demilitarized zone that separates your internal network from the Internet is an excellent way to get a feeling for your network security posture. For network administrators that lack the experience in penetration testing and network device hacking, these tools can provide a report discussing where security vulnerabilities may exist.



Many of these tools are excellent for identifying the technical security posture for your organization, but investing in a security assessment by an external organization may provide better results. Additionally, recognize that as technical security controls improve over time, one of the biggest vectors for infiltration today is through “social hacking.” With social hacking, an external entity bypasses technical controls by talking to employees and pretending to be a member of the organization. Any penetration testing should include a look into the feasibility of an external entity to use this method as well.

Vulnerabilities, Exploits, and Patches

Lastly, all computer devices have bugs and vulnerabilities, either in the code or within that system’s architecture. No matter how you architect your external security posture, a problem with the device itself that goes unpatched becomes a vulnerability that can be exploited. Only through constant and vigilant patching of those vulnerabilities can those holes in your security posture be fixed.

As a network administrator, you will want to ensure you are constantly apprised of known vulnerabilities in your network devices. Subscribe to newsletters, visit pertinent Web sites, and keep yourself aware of these problems as they occur. You also will want to identify a regular period of time that critical patching—and the associated downtime—can occur on your network.

Configuration and Security Management Provide Measurable Benefit

This chapter talked about many of the topics associated with configuration and security management in your network. It has also related how implementing those technical tasks can directly benefit the business. Depending on the timeframe of your business’ life cycle, the maturity of its network, and your desire to move from reactive to proactive, implementation of automation may be a critical success factor. If you and your business make the decision to become proactive in network administration, consider the factors discussed in this chapter when choosing an NMS that assists with this automation.

The final chapter will depart from proactive management and talk specifically about the steps and the tools you need to effectively troubleshoot network devices when they have problems. It will explore keys topics—such as IP address management, network engineering applications, and DNS—that will help you when the network shows a problem.

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.

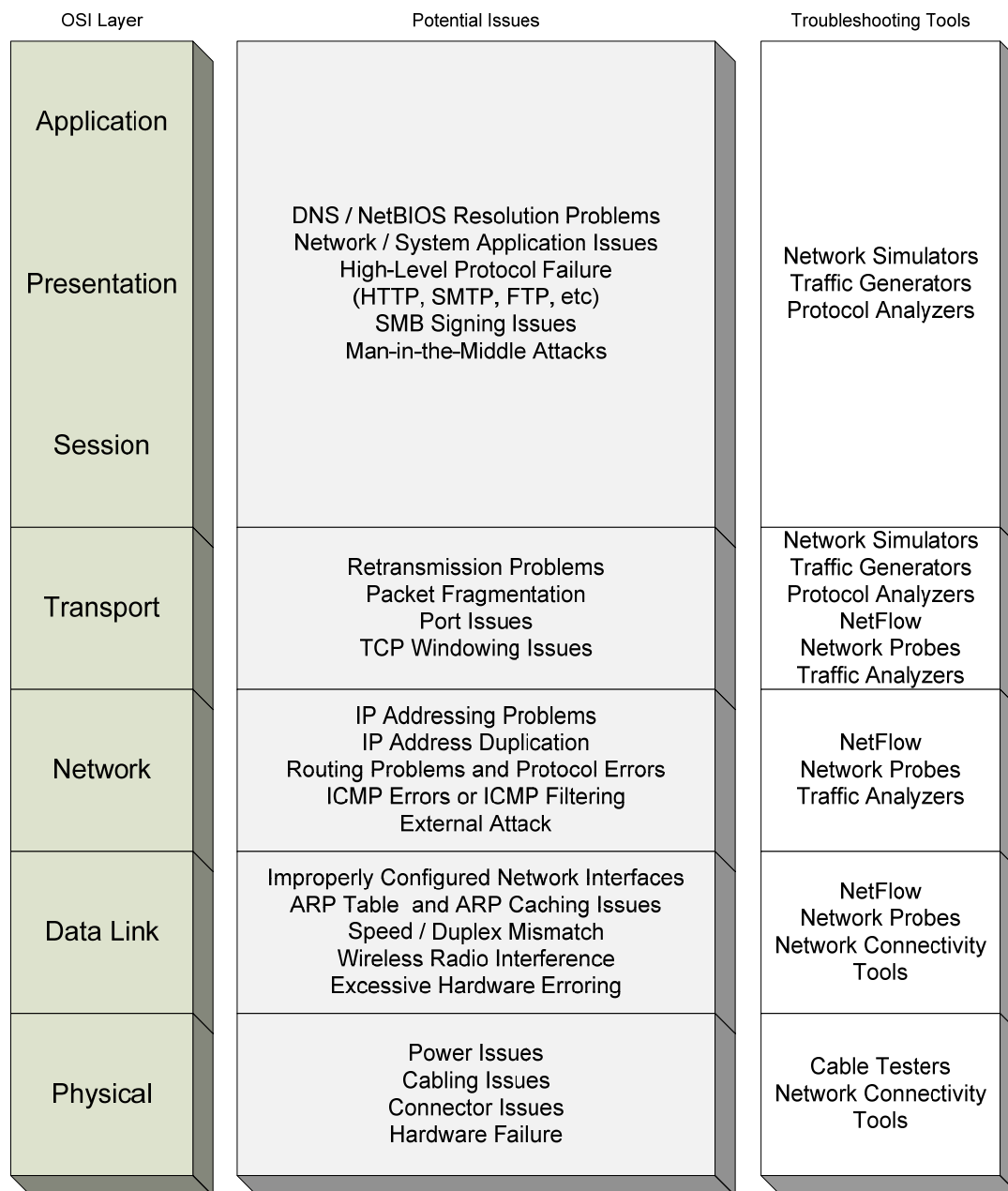


Figure 4.1: The OSI model is an excellent mental framework to assist the troubleshooter with identifying network problems.

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.

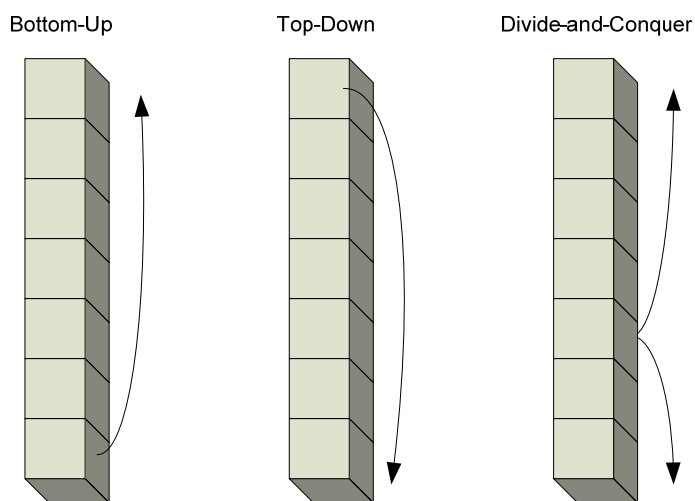



Figure 4.2: Depending on the type of problem, a Bottom-Up, Top-Down, or Divide-and-Conquer approach may be best for isolating the root cause of the problem.

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

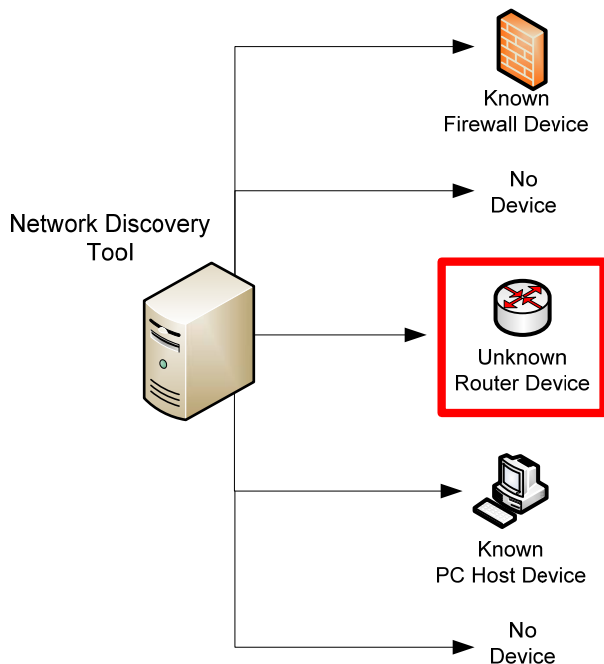


Figure 4.3: A typical network discovery tool will scan a range of addresses to look for the presence or absence of a connected device. Some network discovery tools can compare results with known devices to look for rogue devices on the network.

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 writeable 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.

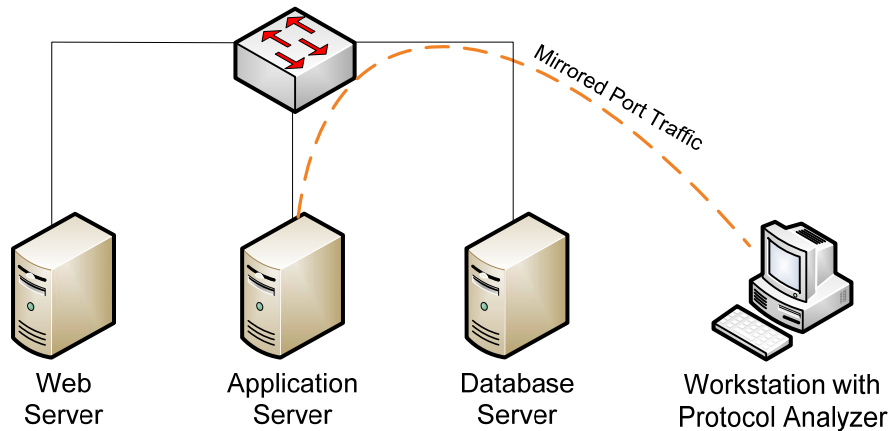



Figure 4.4: In a switched network, for a protocol analyzer to monitor traffic across a link, it is often necessary to mirror that link's traffic to the protocol analyzer.

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.