

**Enterprise Networking
Office of Information Technology
The Ohio State University**

Networking Best Practices

November 5, 2004

Index

1 General Best Practices

- 1.1 Use port names and interface descriptions to the largest extent possible**
- 1.2 Save config files**
- 1.3 Make sure everyone knows contact and contract info for all vendors**
- 1.4 Keep an inventory of devices**
- 1.5 Use AAA services to control access and audit changes**

2 Standard Naming Conventions for Devices

2.1 Naming

3 Example IP plan

- 3.1 Range Type**
- 3.2 Standard IP address allocation**
- 3.3 Document administrative procedures**
- 3.4 Put your devices in your DNS server (forward and reverse)**

4 Fundamental Tools

- 4.1 Built-in tools**
- 4.2 Syslog and Buffered Logging with timestamps**
- 4.3 Set the time across all devices**
- 4.4 Using SNMP for network management**
- 4.5 Use debugging tools**
- 4.6 Connectivity testing tools**
- 4.7 Configure name resolution**

5 Additional Tools

- 5.1 Generic SNMP platform**
- 5.2 Packet Sniffer**
- 5.3 Syslog server**
- 5.4 TFTP server**
- 5.5 Lab**
- 5.6 Data trending capabilities**

6 Procedures Worth Implementing

- 6.1 Change control**
- 6.2 Using network management tools described in the previous section**
- 6.3 Practicing critical procedures**

- 6.4 Periodic software updates**
- 6.5 Understanding how network devices are performing**
- 6.6 Understanding Layer 2 information**
- 6.7 Running vulnerability scans**

7 Effective Troubleshooting

- 7.1 Describing the Problem**
- 7.2 Collecting Data**
- 7.3 Isolating the Problem**
- 7.4 A Note on Software Upgrades**

8 Conclusions

1 General Best Practices

1.1 Use port names and interface descriptions to the largest extent possible

The built-in port names and interface descriptions provide an excellent repository for information and one that is readily available at the time the information is needed. It should be part of your team's standard operating procedures to enter an appropriate description for an interface or port when the port is enabled.

That information is then immediately available to others on the team. By using standard conventions for switch port names, and saving configs nightly via TFTP (a procedure that is further described below), you can also create simple searches on config files to track information and/or changes. Many customers have documentation processes for tracking when ports are brought up, users move, or other changes – often these paper-centric procedures are more time consuming and prone to error than using the switch port descriptors as the repository of the information.

Specific suggestions include:

Putting user names and/or location descriptions in switch port name fields – it is more likely to be accurate since the information could be entered as ports are brought up; and use keywords (like “printer”) in the port names, so that the information can be easily searched later.

1.2 Save config files

Config files are the ultimate repository of information about how your network is built, and as such they need to be guarded feverishly. They should be backed up to;

1. A TFTP server on the network (with an appropriate directory structure that ensures your team can find the most recent config when needed).
2. To a local Flash card whenever possible (so that the config can be moved quickly and accurately between devices without any network connectivity)

3. Regularly, so that an archive of config files is created (the archive will be useful for investigating changes and rolling back configs – these archived configs need to be clearly time stamped)

By saving device configs in more than one place, you not only provide additional protection, but also ensure that the configuration is available in the most effective means in the event that a reconfiguration of a device is necessary.

1.3 Make sure everyone knows contact and contract info for all vendors

The last thing your team should be doing during a network outage is searching for vendor support information. Besides the contact information, your team should also understand the level of coverage provided by the contract (7x24? On-site dispatch?) – any vendor's support will be more effective if you understand the details of the contract.

For vendor support contracts, it is critical that all team members

1. Know the contract info
2. Know how to reach support (phone or Web)
3. Have an account with the appropriate privileges (for downloading software, researching bugs, reading documentation, etc.)
4. Know the level of service purchased (*e.g.*, 4-hour response time versus Next Business Day)

1.4 Keep an inventory of devices

Having an accurate inventory of your active network infrastructure helps your team plan for growth and change; an inventory of spares assures that you have critical components ready in the event of hardware failures. The inventory should include information on

1. Hardware – What devices are on the network, how much spare capacity do they have for growth, how much memory/flash does each device have, what are the serial numbers (for asset/RMA tracking)?
2. Software – What version(s) of code are you running, is the code consistent across the network – this helps your team with bug searches and researching release notes.
3. Lab gear – What additional devices are in the lab (discussed below), are they consistent with the production network?
4. Spares – Not only do you need to know the number and types of spare equipment (including their location!), you need to also track the memory and flash capabilities of spare processors to insure that they can run the appropriate software required on the production network. Sparing is closely linked to the redundancy discussion and support contract information discussed above
5. Software image library – It is also important to keep an inventory of the software images running on the network – so that the appropriate image can be loaded when a new (or replaced) device is added to the network. (These images would reside on a server with TFTP capabilities.)

1.5 Use AAA services to control access and audit changes

AAA (Authentication, Authorization, and Accounting) services let you control who is accessing the network infrastructure, determine what privileges they are granted, and audit their activity. In addition, it provides the most scalable and secure manner of adding or removing user-specific privileges as employees come and go – the alternative is to change router and switch passwords network-wide every time an employee leaves (or to leave the old passwords in place). Most switches support AAA services via RADIUS or TACACS+.

Index

2 Standard Naming Conventions for Devices

Standing naming conventions not only ensure that each device will have a unique name, but also allow you to imbed information in the name. For most customers, the most important piece of information is the location (*e.g.*, closet name and floor number) – because there is no way to determine this through dynamic tools. Imbedding this information in the name makes it easier for new hire, help desk personnel and others to understand where the device is located – one more piece of information that moves problem resolution along.

2.1 Naming

Example computer naming convention

Many people have views on the naming of network-enabled devices. Opinions on this topic range from being very specific, to obscuring the importance of specific devices by using generic names for everything.

The following covers some of the advantages and disadvantages of the different styles. Enterprise Networking subscribes to the approach of using device names that identifies the service, or the piece of equipment that is being used. An example of this is how we name our network gear.

[Building ID][Location ID]-[Type of Interface]-[Interface Port]

The Building ID that we use is the building letter codes from the campus map, and the Location ID follows it. The Location ID is simply a numeric character that represents if the device is the first, second or nth piece of networking equipment in that building. For the Type of Interface we use “fa” for fast Ethernet, “vl” for VLAN, and “gig” for a gigabit Ethernet port.

Using the above format, or a variation on it would provide you with built in identifiers when trouble shoot your network.

The other popular view is that if a specific name is assigned in DNS then it is providing information that could be used against the device in a targeted attack. Due to this thinking many people choose to use generic name styles, some choose to pick a theme such as states, cities, trees, or planets. However, some people choose to use a variation on what we recommend for Dynamic Host Configuration Protocol

(DHCP) ranges, changing the “dhcp” in dhcp-123-456-789-xxx.<domain>.ohio-state.edu to the building location where the subnet resides.

While obscuring server names may hide the type of server in DNS, with the various network-scanning tools available, and due to the need for a server to respond on specific ports security by obscurity does not provide much protection.

Index

3 Example IP plan

Let's develop the above-mentioned example a little further and assume that every subsidiary is assigned a network with a 24-bit subnet mask (255.255.255.0 or /24). Some devices require static IP addresses while others may be configured to use DHCP. In any case we need to reserve some space for those devices that require static IP addresses:

3.1 Range Type

- .1 Network Gateway
- .2-22 Network components, hubs or switches (01-20)
- .23-33 Servers (01-10)
- .34-40 Printers (01-06)
- .41-80 Static IP Clients
- .81-100 reserved
- .101-199 Clients configured by DHCP (01-99)
- .200-254 reserved

This design assumes you don't need more than 10 servers per subsidiary, no more than 20 hubs and/or switches, no more than 6 printers. Also although the IP address range from 200-254 (54 IP addresses) could be used for future expansion of the DHCP pool. There are 19 more IP addresses reserved in this design in the range of .81-100. This should be enough space for some future requirements.

3.2 Standard IP address allocation

As with standard naming conventions, standard IP address allocation allows you to imbed important information the IP address – because this is the only real identifier a packet carries when it traverses the IP network. Many people choose to standardize on a naming convention, which may include theme or location information to aid in identifying devices.

An example of standardization on our part is that when we provide a network for a department we reserve the first address of the network to be used as the network gateway. We then encode into the gateways name the location, device name, and interface port that provides that network connection.

Other examples of standardizations that customers have been to deploy are reserved blocks of IP addresses for particular devices that require static IP addresses – such as:

Host address 1 default gateway for the network
Host addresses 2-5 Switch interface addresses
Host addresses 6-15 Other network infrastructure devices
Host addresses 16-50 Static addresses (*e.g.*, printers that don't support DHCP)
Host addresses 51-254 DHCP pool

3.3 Document administrative procedures

Certain procedures that affect the network are carried out by other groups – like moving a user or bringing up a new port. These procedures should be well documented to ensure consistency and accuracy. With AAA services, it is also possible for more responsibility to be given to the individuals carrying out these tasks – for example, labeling a switch port and moving it to the appropriate VLAN when the wiring is brought up. This contributes to the accurate documentation of the network (see item #2 above).

3.4 Put your devices in your DNS server (forward and reverse)

DNS helps engineers, help desk personnel, and others quickly check the status of a device without having to refer to a diagram or spreadsheet for the IP address. Reverse DNS also helps in the event of traceroute command, mapping an IP address back to a host name.

Index

4 Fundamental Tools

Vendors offer a staggering array of network management tools, and you can obviously spend a great deal of money instrumenting your network. But the definition of network management tools should be expanded to include built-in software features that can provide critical data if configured properly. The key is that both the built-in and external tools need to be configured, up and running, and well understood by the team for them to be effective in times of need.

4.1 Built-in tools

Before taking advantage of the tools built in, clearly access to the devices must be established. Unfortunately, certain types of network problems interfere with “in-band” (over the network) communication with devices. In cases where this occurs at a remote site, the network team is blind to the status of the network until someone can physically assess it. This problem is compounded when the remote site lacks a technical individual who can contribute to gathering data and resolving the problem. Even on a campus with a network support team, out-of-band management can provide a method for engineers to gather data and resolve problems from remote sites (*e.g.*, working from home) saving valuable troubleshooting time that would be otherwise wasted in transit.

Effective out-of-band management tools include

- Console servers
- Modems on AUX ports

AAA services can be used to address the security issues, and modems in particular can be turned off or disconnected when not in use.

4.2 Syslog and Buffered Logging with timestamps

You should have these two types of logging set up for all devices, and your team should understand how to collect and interpret the information that is generated. Timestamps in particular are critical to understanding what has happened and when. Some built in tools allow you to change the level of logging for various functions, and these tools may also be appropriate for troubleshooting certain situations.

4.3 Set the time across all devices

It goes without saying that the logged data loses most of its value if it is not correctly time stamped. Many customers use Network Time Protocol to ensure that all of their network devices are synchronized to the same time – this allows you to assemble an accurate chronology of events that happened in very short periods of time. Most switches support Network Time Protocol.

4.4 Using SNMP for network management

Most network management tools rely on SNMP for their basic functions, and SNMP traps are still the most critical sources for unsolicited error messages. Operationally, someone should be monitoring traps and responding appropriately.

4.5 Use debugging tools

The included system debugging tools can provide a wealth of information – but they need to be used prudently as they can affect the performance on a device by introducing a CPU burden. It is important to work with them in a lab, or in normal mode, both to gain familiarity with the capabilities and to learn what normal output looks like. There are many specifics that help to provide info selectively – like debugging on ICMP only, or using ACLs with a debug tool to limit the traffic that is fully displayed. Particularly useful debugging tools include

- packet debugging with ACLs
- voice/telephony features
- other special features (*e.g.*, DHCP server, H.323 Gatekeeper)

4.6 Connectivity testing tools

Connectivity testing tools include ping, telnet, and traceroute – as well as enhancements that allow you to specify source addresses or change the packet size.

4.7 Configure name resolution

Configuring name resolution on your network devices allows you to use DNS names when troubleshooting from them, speeding up troubleshooting considerably.

Index

5 Additional Tools

The following additional tools should be considered fundamental to managing the network. In our experience, while most customers have invested in network management tools, many have not deployed them or trained their teams enough to take advantage of the tools. Your team needs to learn how to use these tools in advance of problems, so they are not trying to learn what they can do under fire.

5.1 Generic SNMP platform

A good, generic SNMP platform (*e.g.*, Cricket, OpenNMS) builds a dynamic network topology map, provides a trap receiver, does some polling for basic up/down status of network devices, and allows you to poll for specific MIB information (or browse a device MIB); in addition, it should be capable of doing basic thresholding (polling to see if a variable has exceeded a threshold and sending notification), and simple trending. These are basic, essential tools.

In addition, value-add network management software is typically run on the same platform, providing additional tools tailored to your environment.

5.2 Packet Sniffer

In many cases, visibility to the actual data packets on the network holds the clue to resolving an issue (especially when the problem turns out to be an application or authorization issue rather than a network infrastructure issue). Effective use of a packet sniffer also requires

- An understanding of how the SPAN function works on Catalyst switches
- Experience with building capture and display filters to isolate problems
- Experience decoding packets associated with applications, routing protocols, and other traffic
- A 10Mbps and 100Mbps hub available for certain situations
- The ability to ship packet traces to the TAC for investigation

5.3 Syslog server

The natural corollary to having your network devices send syslog messages is to have a syslog server to receive them. For the data to be valuable, you should also consider

- How to managing the syslog data – generally, each day's log file(s) should be archived and timestamped so that it is easy to find events by time and easy to search log archives for past events

- Logging to different syslog files – you may want to have different devices on the network log to distinct files, either on the same syslog server or different one (generally for large networks with hundreds of devices logging events)
- Logging to more than one syslog server

5.4 TFTP server

TFTP (Trivial File Transfer Protocol) provides the means for configurations and images to be uploaded to and saved from network devices, so it is critical that each network have a functioning TFTP server. The appropriate engineering and operations personnel must have access to the server so they can load software images and/or configs if necessary. In addition, to make the TFTP server a valuable archive of software images and config files, you should create a directory structure that organizes the saved data – for example, creating sub-directories based on network location (/tftpboot/BakerSystems/) or type of device (/tftpboot/SwitchConfigs/).

As mentioned earlier, config files should be backed up periodically, timestamped and archived.

5.5 Lab

A working lab can be used for several important functions, including

- Practicing critical procedures like software upgrades, supervisor module swaps (see next section for more detail) to ensure flawless execution in times of duress
- Testing redundancy and scenarios to better understand how the network will perform during link/component failures
- Testing migration plans for significant network changes (*e.g.*, changing switches)
- Testing software for bug fixes, new features, and configuration enhancements before rolling them to the production network
- General education of the network engineering and network operations team in networking technologies
- Assessing both the performance of new applications on the network, and the impact to the network of adding those applications. For the lab to have value, it should mirror to the largest extent possible the gear and topology of the actual network. In particular, it should be able to mimic LAN and should include some hosts and PCs for studying enduser applications.

5.6 Data trending capabilities

It is difficult to provide performance trending for all portions of a switched network, but there are several points in the network where you need to watch trends carefully to ensure that the network capacity is meeting demand. These include

- WAN link utilization and errors
- Internet link utilization and errors
- Shared media utilization and errors (*e.g.*, 10Base-T hub segments, Token Rings)
- Utilization and errors at traffic concentration points, links to server farm segments)

You should also try to get a sense of application response times across the network for certain critical applications, to have a performance baseline on which to judge changes. In addition, you may want to occasionally create ad-hoc trending reports on various aspects of the network

To create trend graphs, you need a tool that will automatically poll the appropriate devices and graph the results. Options include (among others)

- MRTG/Cricket/OpenNMS (shareware)

Index

6 Procedures Worth Implementing

It has been estimated that as many as 80% of network problems are due to human error. The procedures described in this section are designed to help protect against human-induced outages, and also to train the network team on critical problem resolution procedures to ensure that they are executed properly when required.

6.1 Change control

Change control provides a forum where change is planned, discussed, and shared – this is especially critical in larger organizations where multiple IT groups (*e.g.*, server group, infrastructure group) are making changes. By giving visibility to planned change, change control allows the IT staff to better anticipate the impact of upcoming change and leads to better educated post-change troubleshooting techniques.

Good change control includes

- Some sort of documentation of planned change that details what is changing, the anticipated impact, and a point-of-contact for the change
- An emphasis on “migration” approach to change whenever possible, versus a “flash-cut” approach
- A testing plan that describes how the implementers will judge whether the change is successful
- A plan for rolling back the change if necessary

Finally, change should be validated to the extent possible at the time; the network team should then prepare for the unexpected over the next days – and keep in mind the back-out plan if necessary.

6.2 Using network management tools described in the previous section

The various tools described in the previous section will not provide any value unless the network team understands what they can do and how to make them do it. As mentioned in the previous section, IOS tools like the debugging facility can provide rich data to those who are familiar with how to get it and interpret it.

6.3 Practicing critical procedures

There are some procedures that you don't want to have the network support team learning "under fire" – these procedures need to be practiced so they can be executed flawlessly when needed:

- Swapping non-trivial components (*e.g.*, RSP, Supervisor Module) that have their own software and config (this includes knowing how to retrieve and reload the appropriate software image and config file);
- Testing redundancy to make sure it works "as designed" (*e.g.*, dial back-up, Supervisor Module failover) and the team knows how to recover once the underlying issue is fixed;
- Upgrading and downgrading software images – this also tests the team's TFTP server functionality;
- Password recovery procedures;
- Backtracking configs – you may need to go back to a known good config from a current config; and
- Changing line cards in switches – understanding the impact on the config if the new line card is not the same as the old one.

6.4 Periodic software updates

Proactive software upgrades allow you take advantage of more stable versions as they become available, and will protect the network against bugs that may or may not have affected the network to date. Some customers resist doing software upgrades without specific bug fixes or feature requirements; however, Most software release strategies do deliver progressively more stable maintenance releases over time that will improve network stability.

One of the easiest ways to research software updates is to periodically read the release notes for maintenance releases beyond your current release – this will give you an idea of both the bug fixes and feature enhancements available to you by upgrading. You may occasionally find a bug fix that you want to take advantage of right away.

Before upgrading code, your team should carefully read the associated release notes, consider whether you want to take advantage of any new features that may be introduced, and ensure that the basic requirements (DRAM, Flash memory) are met.

6.5 Understanding how network devices are performing

Because network infrastructure devices are really just specialized computers, their health can be monitored in much the same way as a host. Specifically, your team should know how to

- Show CPU utilization and the process table, determine a baseline of expected CPU utilization
- Show memory utilization and understand how it is changing over time (*e.g.*, to recognize a memory leak)
- Understand the difference between a stack trace, a "crashinfo" file, and a core dump – and how the information they contain can be interpreted by you or the TAC.

6.6 Understanding Layer 2 information

Your team should be able to find and interpret critical Layer 2 information from the network, including

- Catalyst CAM table
- Speed and duplex settings and negotiation
- Layer 2 counters (for router or a switch) – including packet and error counters
- Layer 2 neighbors (using CDP)
- VLAN, trunking and VTP information
- EtherChannel information
- Spanning Tree information (how to set the root, find the root, working with Spanning Tree enhancements)

6.7 Running vulnerability scans

Finally, you should periodically scan the network from the inside and outside to ensure that the configured security is working, as it should and that no unexpected hosts or services are appearing on the network.

Index

7 Effective Troubleshooting

Much of the content described above is meant to prepare your team for the inevitable – because there will be problems, failures, mistakes, and unanticipated events. However, the suggestions above should reduce the likelihood that problems will be the results of mistakes and prepare your team, through practice and documentation, to work efficiently and quickly on the network. In fact, based on these suggestions, if a problem does occur

- Your team should be aware of change on the network – whether yours or others – that might be contributing to the problem;
- The designed network redundancy should be kicking in if necessary;
- The network will be instrumented to collect valuable data to aid in resolution;
- There will be a documented backout plan (assuming the problem is related to a planned change);
- Your team will have practiced critical procedures that might be required (hardware failure, software upgrade/downgrade, config backtrack);
- Your team will have accurate and clear documentation of your network.

This section is divided into two parts – Basic Troubleshooting, which describes some fundamental approaches to troubleshooting problems, and Escalated Troubleshooting, which describes some best practices for working on more severe or complex issues. Clearly, a detailed primer on troubleshooting internetworking issues is beyond the scope of the document – instead the goal is to describe some overall troubleshooting guidelines.

7.1 Describing the Problem

The key in problem resolution is problem isolation – typically finding the source of the problem is much harder than finding a solution or a workaround that negates the problem. The first goal in problem

isolation is to define the problem as accurately as possible. In that spirit, your team should be asking these questions:

- Who is affected by the problem (certain users, certain application, certain physical or logical locations on the network)?
- What is the effect (intermittent problems, constant issues, do things work poorly or not at all)?
- What is *not* affected (different applications on the same PC, different users on the same floor, other users accessing the same application)?
- What changed recently (check change control and audit logs – you might be surprised) and when did the problem start?
- What information is in the obvious logs (traps, syslog)?
- What is the periodicity of the problem if it re-occurs (same time of day, every *x* hours, during traffic peaks, while backups occur)?

These questions may expose useful information that is not immediately obvious, and may also suggest an initial troubleshooting strategy.

In general, it is also worth eliminating the obvious possibilities first – by checking port counters, interface errors, basic connectivity, speed and duplex matches – even if your team thinks the problem may be more complex. In addition, your team should keep in mind that servers, applications, and PCs are also possible culprits and should not be ruled out of the troubleshooting strategy.

7.2 Collecting Data

Once your team has a good idea of the scope and characteristics of the problem, you can use the various tools described earlier to begin collecting data to support or rule out certain possibilities.

Especially in the early data collection phases, it is important that you capture troubleshooting sessions (telnet sessions) to a log so the data is available for later review. If your team is unsure about what data to collect, or you are about to take a dramatic step like rebooting a router or switch, you should capture the output of a couple of “show tech” commands just to get a snapshot of data at the time.

Ideally, an initial study of the problem has led to one or more hypotheses that can be supported or disputed by key pieces of data. In this case, the troubleshooting process is a recursive process of refining the hypothesis, gathering data to support or undermine the hypothesis, and further refining the hypothesis based on that data. Again, the suite of tools described earlier should be able to provide the data needed in most cases. If the problem has disappeared without leaving any meaningful data behind, you should plan in advance the data you want to capture when it reoccurs, and make sure the appropriate tools and scripts are set up in advance. It may not be possible to determine with certainty what the problem was, but as long as some strategic data is collected every time it happens, then the problem is moving toward resolution.

7.3 Isolating the Problem

Isolating the problem involves distilling the problem into the smallest possible set of devices and scenarios.

In order to do this, all possibilities must be initially considered, and then examined individually and systematically through data collection. An isolated problem is one where you can point to the device(s) that are not functioning correctly, for whatever reason.

At this point, one of three resolutions should be implemented:

- A fix (if the problem has to do with configuration)
- A replacement (if the problem has to do with a hardware failure)
- A workaround (if the problem is a bug that is not currently fixed)

7.4 A Note on Software Upgrades

It should be clear that customers who run earlier releases of code are more likely to run into documented *or undocumented* bugs. In the spirit of troubleshooting and taking advantage of the most stable code releases, it is often a good idea to incorporate a software upgrade into the troubleshooting process. In the cases where it does address the issue, this move obviously pays off; in other cases, it serves to eliminate certain possibilities and ensures that the code that is running is one with ongoing development engineering support.

In escalated situations, where more of the network is down, the problem is more evasive, or a particularly sensitive application/user pool is affected, the general approach to troubleshooting remains the same, augmented with some additional practices. Among the most important of these is coordinated troubleshooting – since it is likely that this type of problem requires more customer expertise, involving more people.

Index

8 Conclusions

As owners of a network, your team's responsibilities include:

- implementing a solid, well-documented network design;
- caring and feeding for the network;
- preparing your team for troubleshooting;
- investing in the appropriate tools; and
- taking overall ownership of network issues and driving them to resolution.

Index