Department of Information Systems

Arkansas. A State of Technology.

Basic Network Troubleshooting: Tips, Techniques & Tools

Prepared by:

DIS APSCN LAN Support

Revised: June, 2015

Table of Contents

Introduction

While network technologies have changed, the basic methods for troubleshooting networks really haven't, the real basics that demand an understanding of networking to the core level remain the same. You will eventually find yourself trying to fix a network related problem that usually appears in one of two forms. The first is slow response times or poor performance, and the second is a complete lack of connectivity

Establish A Baseline

The single most important tool you can use to troubleshoot your network is to have established and documented a network baseline. Due to the time consuming nature of establishing a network baseline through testing, analysis and documentation it is also the single most neglected tool.

What is a Baseline?

A baseline is a process for studying the network at regular intervals to ensure that the network is working as designed and documentation of the findings. It is more than a single report detailing the health of the network at a certain point in time. Establishing a baseline defines the parameters under which it operates, its limits and behavior under specific conditions.

Why Establish a Baseline?

Identify Normal vs. Abnormal Function

You can use a baseline analysis, which is an important indicator of overall network health, to identify problems. By monitoring your network over a long period and establishing a baseline, you obtain a useful reference of network traffic during normal operation, which you can then compare to captured network traffic while you troubleshoot network problems. You begin to see a pattern in the traffic flow, such as which servers are typically accessed when peak usage times occur and so on. If you are familiar with your network when it is operational, you can be more effective at troubleshooting problems that arise.

Plan For Potential Problems and Future Growth

Monitoring and documenting long term increases in the amount of network traffic, occurrences of problems, types of systems and services being used on the network, frequency of hardware failures, etc., can be used to predict the need for additional bandwidth, upgrades or replacement of hardware components, need for additional layers of security and/or changes in network use policy. Good documentation becomes justification in the request of funding for additions to the network as well as adding support staff.

Procedure For Establishing a Baseline

Develop and maintain a Site Network Map.

A network map helps you to:

- Know exactly where each device is physically located
- Easily identify the users and applications affected by a problem with each device.

- Systematically search each part of your network for problems.

You can create a network map using any drawing or flow chart application. Store your network map online. In addition, make sure that you *always* have a current version on paper in case you cannot access the online version.

Figure 8a *Examples of a Site Network Map*

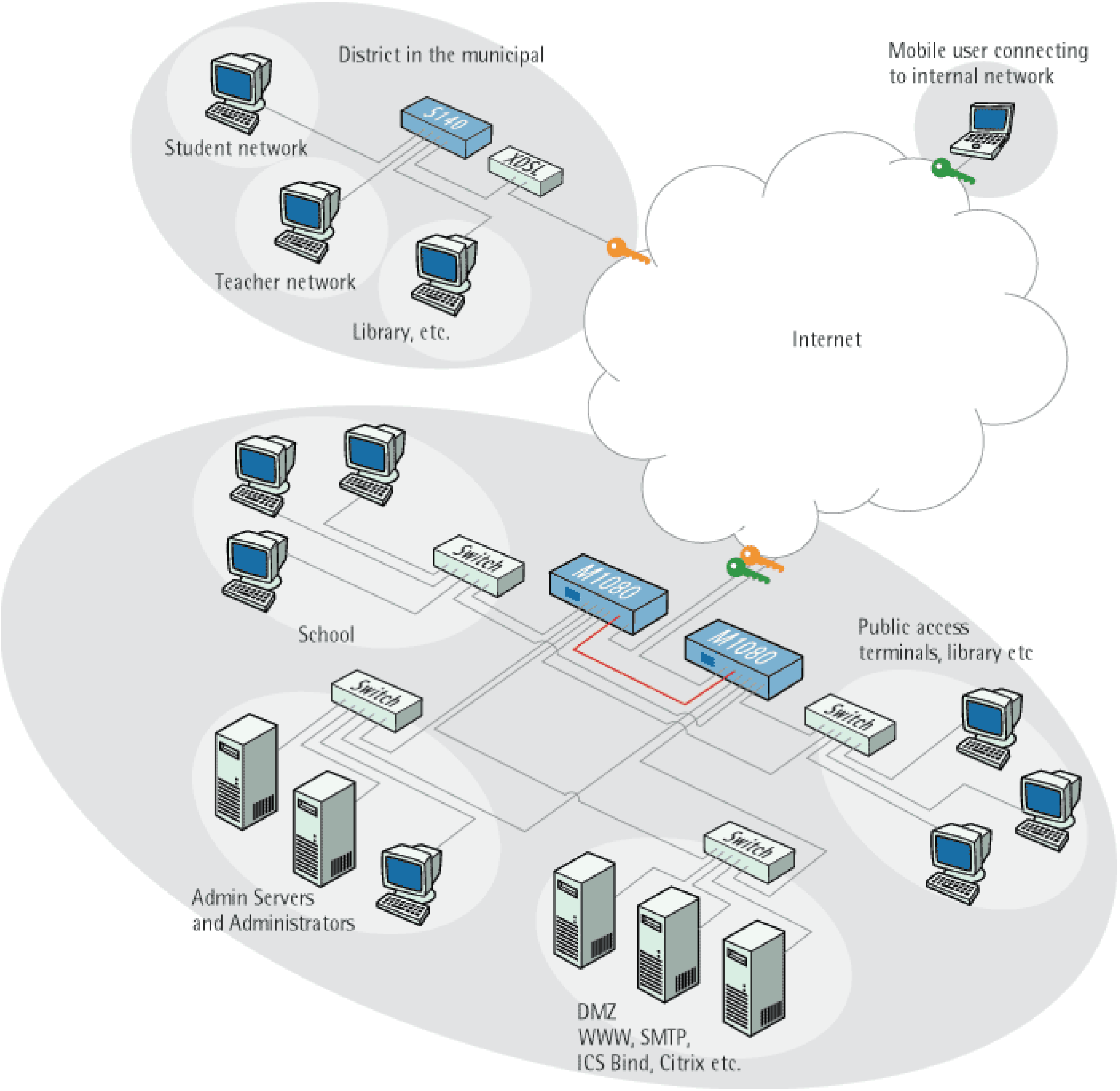
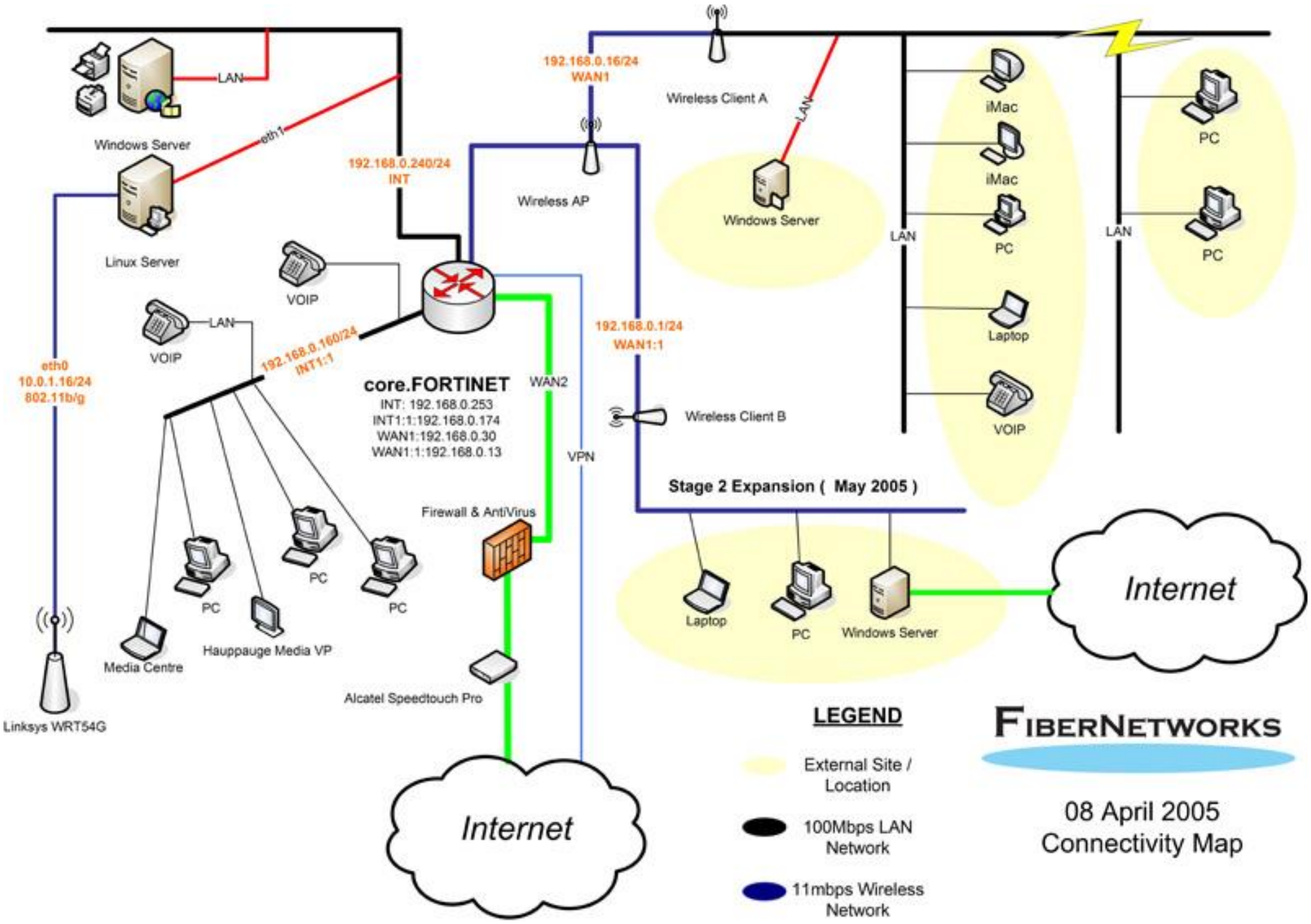


Figure 8b *Examples of a Site Network Map*



Consider including the following information on your network map:

- Location of important devices and workgroups (by floor, building, or area)
- Location of the network backbone, data center, and wiring closets, as appropriate for your network
- Location of your network management stations
- Location and type of remote connections
- IP subnetwork addresses for all managed switches and hubs
- Other subnetwork addresses, such as Novell IPX and AppleTalk, if appropriate for your network
- Type of media (by actual name, such as 10BASE-T, or by grouping, such as Ethernet), which you can show with callouts, colors, line weights, or line styles

- Virtual workgroups, which you can show with colors or shaded areas
- Redundant links, which you can indicate with gray or dashed lines
- Types of network applications that are used in different areas of your network
- Types of end stations that are connected to the switches and hubs

NOTE: *Complete data about end station connections is usually too detailed for the network map. Instead, maintain tables that detail which end stations connect to network device, along with the MAC addresses of each end station. Some diagramming programs allow storing detailed information about devices within the drawing database. This information is then viewed in onscreen tables, printed or exported to another program file type.*

Include Details of Logical Connections.

With the advent of virtual LANs (VLANs), you need to know how your devices connect logically as well as physically. For example, if you have connected two devices through the same physical switch, you can assume that they can communicate with each other. However, the devices can be in separate VLANs that restrict their communication.

Knowing the setup of your VLANs can help you quickly narrow the scope of a problem to a VLAN instead of to a network connection.

Compile a Hardware, Software, and Configuration Inventory (Network Notebook)

Maintain online and paper copies of device configuration information. Make sure that all online data is stored with your site's regular data backup. If your site does not have a backup system, copy the information onto a backup disc (CD, Zip disk, and the like) and store it offsite.

For a complete picture of your network, have the following information available:

- **All passwords** - Store passwords in a safe place. Keep previous passwords in case you restore a device to a previous software version and need to use the old password that was valid for that version.
- **Device inventory** - The inventory allows you to see the device type, IP address, ports, MAC addresses, and attached devices at a glance.
- **MAC address-to-port number list** - If your hubs or switches do not have an OS that enables management, you should keep a list of the MAC addresses that correlate to the individual ports. Generate and keep a paper copy of this list.
- **Logbook** - Document your interactions with network devices and software systems (routers, remote access devices, security servers, and application servers), *no matter how trivial*. For example, document that you noticed a fan making noise one morning. Your note may help you to identify why a device is over temperature a week later (because the fan stopped working).
- **Change control** - Maintain a change control system for all critical systems. Permanently store change control records.
- **Contact details** - Store, online and on paper, the details of all support contracts, support numbers, engineer details, and telephone and fax numbers.

Collect Statistical Data for Network

The type of statistical data you collect will depend on capability of the systems you have installed or can obtain for collecting such data, your specific information needs and the network platform. The most fundamental should include:

- Total Bandwidth Utilization For LAN & WAN
- Outbound Bandwidth Utilization
- Inbound Bandwidth Utilization
- Protocols and Ports Used
- Time of Day for Highest Utilization
- Days of Week/Month for Highest Utilization
- Network Segment with Highest Utilization

Analyze Data and Establish Thresholds

Physical Health Analysis.

Review and evaluate your networks physical topology and verify that it conforms to current IEEE standards for the topology that you choose whether it is 10Mb over CAT5 copper cable or 10Gb over fiber optic cable. Be aware of the strengths and weaknesses of the various hardware devices used to connect to network resources and the limits of their function. Determine the packet error profile and determine a threshold of normal occurrence for your network. Example; Ethernet uses *carrier sense multiple access/collision detection* (CSMA/CD) as a media access method which by design will have collision errors in a non-switched Ethernet environment as devices attempt to gain access to the network. If you use any hubs in building the network, you should expect to have a nominal level of collisions.

Broadcast Analysis.

Broadcast traffic, or traffic simultaneously addressed to all computers connected to the network, as opposed to unicast or multicast traffic, is another normal occurrence for which you should establish a threshold. The key here is to understand the difference between a normal broadcast event and a broadcast storm. When a normal broadcast event occurs, the broadcast is from a specific physical device on a network for the express purpose of achieving a network communication cycle. There are conditions when a device protocol, such as those on a router or a switch, broadcasts information to update other routers and switches on the network to ensure routing and arp tables maintain consecutive and consistent data. Another standard broadcast occurs when a device attempts to locate another device and requires the physical address or IP address of another device.

When a specific workstation device has a default gateway assigned, a "normal" broadcast event can occur. The device knows, for example, the target IP address of a device on the internetwork. It is common for this device to broadcast an ARP sequence to attempt to locate the target hardware address. ARP broadcasting is discussed in detail later in this document.

A workstation that broadcasts an ARP sequence to locate a target server but doesn't establish a broadcast resolve and doesn't receive a target hardware address for the server provides an example of an "abnormal" broadcast event. If the target device fails or the source broadcast operation mechanism or protocol-sequencing mechanism of the device fails, the source workstation device could start performing a loop ARP sequence that could be interpreted as a broadcast storm. Such an event in itself could cause a broadcast storm.

Network capacity overload analysis.

When examining utilization, it is important to understand both the available capacity on any network medium and actual achieved utilization levels from an average, peak, and historical perspective. Every network LAN or WAN topology has an available capacity. Determining the utilization levels of a topology is important, but equally important is identifying any problematic utilization levels or saturation utilization levels. Saturation of any main network medium can cause outages on a network related to an end-to-end session. Use peak utilization and time measurement methods to identify any outages.

Other conditions exist when the capacity, even if available, may be in an overload condition in certain topologies.

Consider, for example, a 10Mbps shared media Ethernet topology operating at 60+% utilization levels. The Ethernet topology in a shared configuration normally allows for a specific maximum capacity of 10Mbps or 100Mbps. Can the shared Ethernet medium sustain the applied utilization levels and continue to operate in a positive manner? Although capacity levels may only be operating at a peak transition of 60% or 70%, and approximately 30% to 40% of medium may appear available, the CSMA/CD mechanism of shared Ethernet could trigger an excessive collision problem at this level. In shared Ethernet media, the collision-detection mechanism can increase to a level that causes problematic events at the physical level when utilization exceeds 30% of available capacity. In this example, a level as high as 60% of the available capacity can constitute a network overload condition

Fix Immediate Problems Identified

You should correct any problems identified during the process of establishing the baseline to optimize network performance and prevent future growth of the problem.

Network Problems Analysis

Performance Problems

Your network has performance problems when it is not operating as effectively as it should. For example, response times may be slow, the network may not be as reliable as usual, and users may be complaining that it takes them longer to do their work. Some performance problems are intermittent, such as instances of duplicate addresses. Other problems can indicate a growing strain on your network, such as consistently high utilization rates.

If you regularly examine your network for performance problems, you can extend the usefulness of your existing network configuration and plan network enhancements, before a performance problem adversely affects the users' productivity.

Sources of Network Slowness

- Poor routing
- Misconfigured router or switch
- Bad cabling
- Over utilized capacity
- Malware running on the network
- Misconfigured circuit between sites
- Excessive use of network protocols
- Electrical interference
- An overloaded server at the remote end of the connection
- Misconfigured DNS

Duplex and Speed Setting Mismatches.

Because twisted pair Ethernet infrastructure devices come with so many different options (i.e., auto-negotiation, full-duplex, half-duplex, 10Mbps, 100Mbps, 1000Mbps, etc.) with many possible combinations of these options, it is possible, if not probable, that you will encounter a condition of duplex or speed mismatch.

A duplex mismatch is a condition where two connected devices operate in different duplex modes, that is, one operates in half duplex while the other one operates in full duplex. The effect of a duplex mismatch is a network that works but is often much slower than its nominal speed. Duplex mismatch may derive from manually setting two connected network interfaces at different duplex modes, when a connecting a device that performs auto-negotiation to one that is manually set to a full duplex mode. This can also occur when both port are set to authnegotiate and one port on the link operating at half-duplex mode while the other port is operating at full-duplex mode as the result of a port reset switch reset. This occasionally happens when one or both ports on a link are reset and the autonegotiation process doesn’t result in the same configuration for both link partners. A switch-to-switch link that has been allowed to negotiate its behavior could end up operating a different behavior mode than its partner.

Communication *is* possible over a connection in spite of a duplex mismatch. Single packets are sent and acknowledged without problems. As a result, a simple *ping* command fails to highlight a duplex mismatch because single packets and their resulting acknowledgments at 1-second intervals do not cause any problem on the network. A terminal session that sends data slowly (in very short bursts) can also communicate successfully. However, as soon as either end of the connection attempts to send any significant amount of data, the network suddenly slows to very low speed.

A duplex mismatch causes problems when both ends of the connection attempts to transfer data at the same time. A large data transfer occurs over a TCP connection in multiple packets, some of which will trigger an acknowledgment packet back to the sender. This results in packets sent in both directions at the same time.

In such conditions, the full-duplex end of the connection sends its packets while receiving other packets; this is exactly the point of a full-duplex connection. Meanwhile, the half-duplex end cannot accept the incoming data while it is sending -- it will sense it as a collision. As a result, almost all of the packets sent by the full-duplex end will be lost because the half-duplex end is streaming either data packet or acknowledgments at the time.

The end result is a connection that is working but performs *extremely* poorly. Symptoms of a duplex mismatch are connections that seem to work fine with a *ping* command, but "lock up" easily with very low throughput on data transfers. The effective data transfer rate is likely to be asymmetrical, performing much worse in one direction than the other. In a duplex mismatch situation the collisions are usually late collisions. Viewing this standard Ethernet statistic can help diagnose this problem.

IP Address Conflicts

Duplicate IP addresses on the network causes problems with correct delivery of data packets. Duplication of IP addresses can occur when using static IP addresses configured manually.

DHCP automatically assigns TCP/IP addressing to computers when they join the network and automatically renews the addresses *before* they expire. The advantage of using DHCP is the reduced number of addressing errors, which makes network maintenance much easier. Because DHCP IP addressing is automatic and does not assign duplicate IP addresses, as sometimes happens with manual entries, DHCP is the preferred method of network IP assignment.

As always, a cost is associated with everything good, and with DHCP, the cost is increased network traffic. Some network services can consume huge amounts of network bandwidth, but DHCP is not one of them. The traffic generated between the DHCP server and the DHCP client is minimal during normal usage periods.

The bulk of the network traffic generated by DHCP occurs during two phases of the DHCP communication process: when the lease of the IP address is initially granted to the client system and when that lease is renewed. The entire DHCP communication process takes less than a second, but if there are a very large number of client systems, the communication process can slow down the network.

For most network environments, the traffic generated by the DHCP service is negligible. For environments in which DHCP traffic is a concern, you can reduce this traffic by increasing the lease duration for the client systems, thereby reducing communication between the DHCP client and the server.

Network Congestion

Whether it's due to a broadcast storm, increasing connections, excessive protocols or over utilization of the bandwidth, as packets increase latency increases and packets begin to be dropped.

Hardware Failure

Troubleshooting hardware infrastructure problems presents a significant challenge. It is often not an easy task and usually involves many processes, including baselining and performance monitoring. One of the keys to identifying the failure of a hardware network device is to know what devices exist on a particular network and each device's designed function.

Some of the common hardware components used in a network infrastructure are shown in ***Table 1****.*

Common network hardware components, their function and troubleshooting strategies.

Networking Device	Function	Troubleshooting and Failure
Hubs	Hubs are used with a star network topology and UTP cable to connect multiple systems to a centralized physical device.	Because hubs connect multiple network devices, if many devices are unable to access the network, the hub may have failed. When a hub fails, all devices connected to it will be unavailable to access the network. Additionally, hubs use broadcasts and forward data to all the connected ports increasing network traffic. When network traffic is high and the network is operating slowly, it may be necessary to replace slow hubs.
Switches	Like hubs, switches are used with a star topology to create a central connectivity device.	The inability of several network devices to access the network may indicate a failed switch. If the switch fails, all devices connected to the switch will be unable to access the network. Switches forward data only to the intended recipient allowing better data management than with hubs.
Routers	Routers separate broadcast domains and connect different networks.	If a router fails, network clients will be unable to access remote networks connected by the router. For example, if clients access a remote office through a network router and the router fails, the remote office would be unavailable. Test router connectivity using utilities such as ping and tracert.
Bridges	Bridges connect network segments within the same network. Bridges manage the flow of traffic between these network segments.	A failed bridge would prevent the flow of traffic between network segments. If communication between network segments has failed, it may be due to a failed bridge.
Wireless Access Points	Wireless access points provide the bridge between the wired and wireless network.	If wireless clients are unable to access the wired network, the WAP may have failed. However, there are many configuration settings to verify first.

With hardware considerations, you should also keep in mind cable failures. A break in a single wire of a four pair CAT5 or 5e slowness due to errors or total loss of connectivity. If you build your own cables, it would be wise to invest in an inexpensive Ethernet cable tester to use for diagnosing wire mis-matches when installing connectors onto cables.

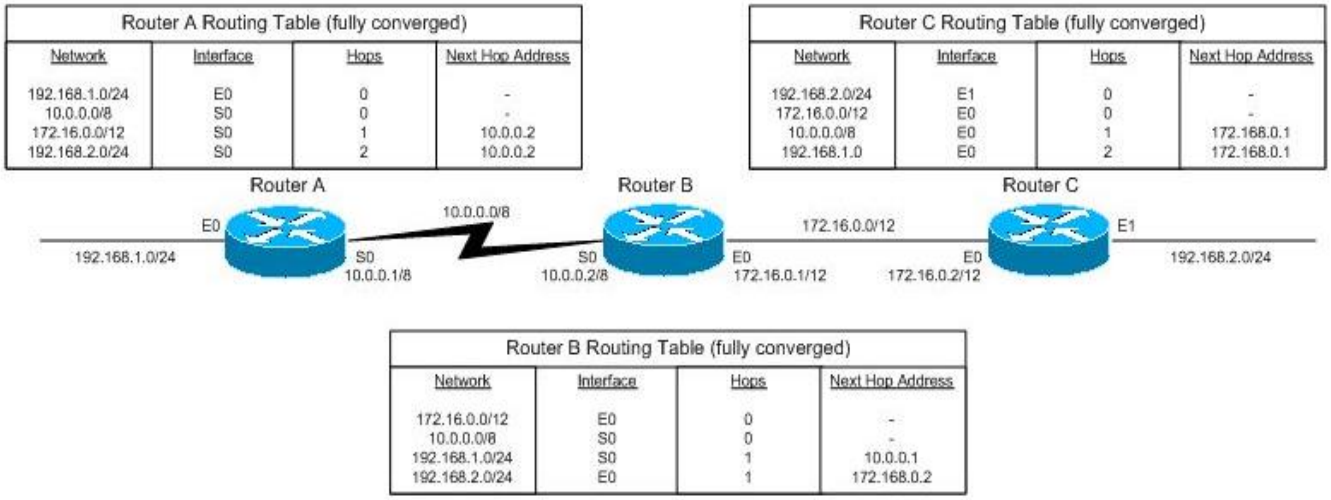
As with copper cables fiber optic cables come with their own set of potential problems. Fiber optic cables are very delicate and damaged can result from improper handling during installation. Even after installation, any kind of stress, whether minor mechanical loads or temperature extremes, can result in micro bends or other fiber stress that in turn may lead to increased cable loss and transmission errors, or even eventual fiber failure and breakage. They can also become cloudy over time due to heat and UV exposure. Water permeation of cables will result in optical loss increases in the fiber from hydrogen infiltration.

Network Loops

Loops in the network topology can cause a plethora of symptoms. Slowness that progressively gets worse, complete lack of communication, IP address conflicts, etc.

Routing Loops

One of the main issues with distance-vector routing protocols is that they are susceptible to routing loops – a direct result of their slow convergence times. A routing loop can occur in the distance vector world because of the way routers exchange information. For example, let’s say that we have a network as shown in the figure below. Three routers exist in this example, connecting four networks. We’ll begin with a network that is fully converged – that is, all routers are aware of all networks, as shown in the diagram. On a fully converged network, everything works well.



Router A Routing Table (fully converged)

Network	Interface	Hops	Next Hop Address
192.168.1.0/24	E0	0	-
10.0.0.0/8	S0	0	-
172.16.0.0/12	S0	1	10.0.0.2
192.168.2.0/24	S0	2	10.0.0.2

Router C Routing Table (fully converged)

Network	Interface	Hops	Next Hop Address
192.168.2.0/24	E1	0	-
172.16.0.0/12	E0	0	-
10.0.0.0/8	E0	1	172.168.0.1
192.168.1.0	E0	2	172.168.0.1

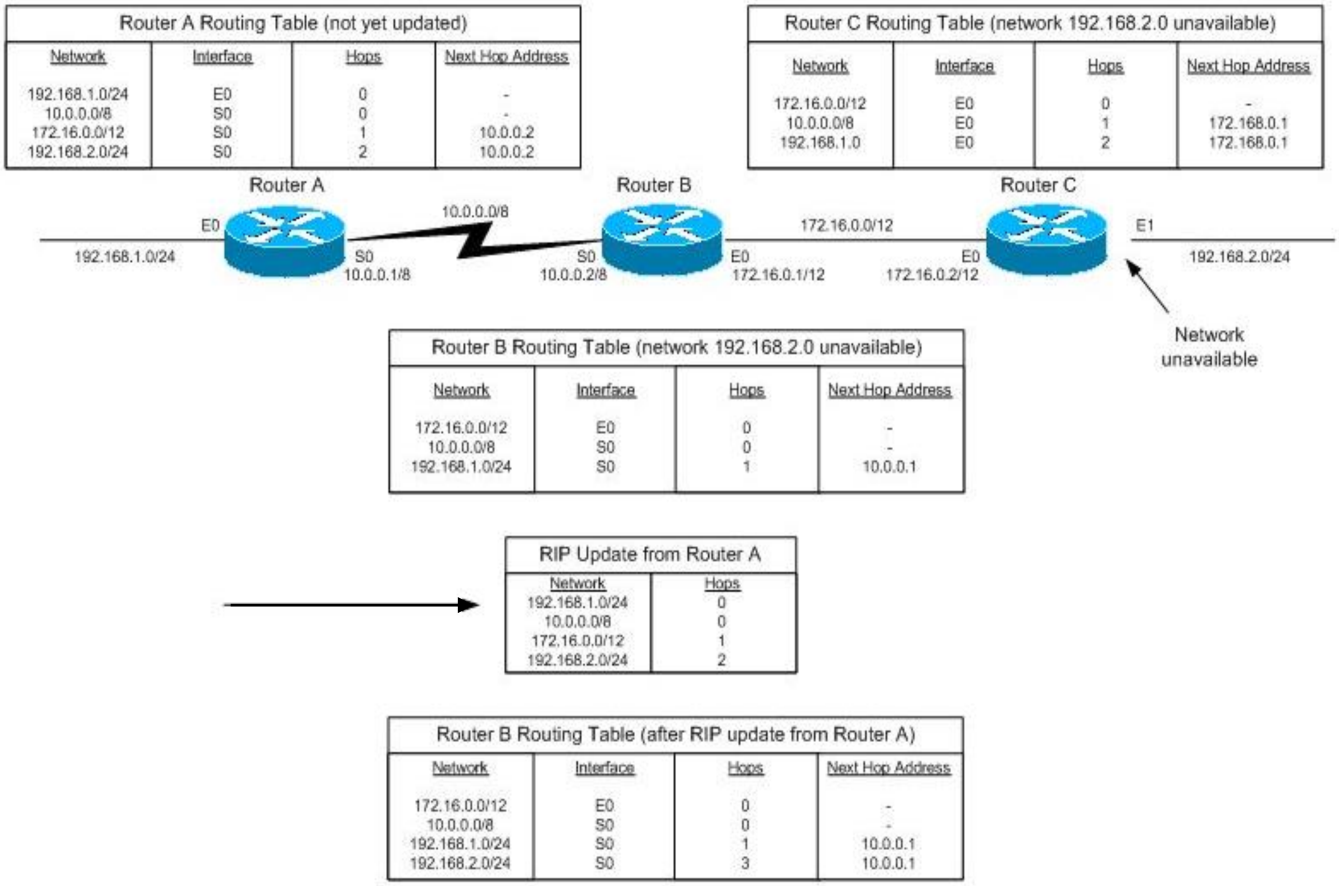
Router B Routing Table (fully converged)

Network	Interface	Hops	Next Hop Address
172.16.0.0/12	E0	0	-
10.0.0.0/8	S0	0	-
192.168.1.0/24	S0	1	10.0.0.1
192.168.2.0/24	E0	1	172.168.0.2

Routing loops become a potential issue when our network experiences a problem. For example, imagine that network 192.168.2.0 experiences a failure – maybe the switch malfunctioned, or somebody simply disconnected the cable. At any rate, Router C recognizes that network 192.168.2.0 is unavailable, and passes this information to Router B in its next routing table update. Once the update arrives, Router B removes the entry for network 192.168.2.0 from its routing table. So far, things are going well. However, there is one little problem – Router A is also sending out routing table updates, and is telling Router B that network 192.168.0.0 is available through it, with a hop count of two, as shown in the figure below. See a problem developing?

Now we have a network where Router B thinks it can get to network 192.168.2.0 via Router A. Without anything else occurring, think about what happens here. When Router B gets a packet destined for network 192.168.2.0, it will send it to Router A. Router A will look at the packet, and will send it back to Router B – that is the direction of network 192.168.2.0 as far as Router A is concerned, after all. The packet will actually end up passed back and forth forever and ever. This is certainly not a very comforting thought.

The problem just described is a directly related to slow convergence. Router C did its job in getting information about the unavailable network to Router B, but unfortunately, Router A also sent an update to Router B prior to the "correct" information arriving at Router A. On a larger network, the problem would be even worse.



Router A Routing Table (not yet updated)			
Network	Interface	Hops	Next Hop Address
192.168.1.0/24	E0	0	-
10.0.0.0/8	S0	0	-
172.16.0.0/12	S0	1	10.0.0.2
192.168.2.0/24	S0	2	10.0.0.2

Router C Routing Table (network 192.168.2.0 unavailable)			
Network	Interface	Hops	Next Hop Address
172.16.0.0/12	E0	0	-
10.0.0.0/8	E0	1	172.168.0.1
192.168.1.0	E0	2	172.168.0.1

Router B Routing Table (network 192.168.2.0 unavailable)			
Network	Interface	Hops	Next Hop Address
172.16.0.0/12	E0	0	-
10.0.0.0/8	S0	0	-
192.168.1.0/24	S0	1	10.0.0.1

RIP Update from Router A	
Network	Hops
192.168.1.0/24	0
10.0.0.0/8	0
172.16.0.0/12	1
192.168.2.0/24	2

Router B Routing Table (after RIP update from Router A)			
Network	Interface	Hops	Next Hop Address
172.16.0.0/12	E0	0	-
10.0.0.0/8	S0	0	-
192.168.1.0/24	S0	1	10.0.0.1
192.168.2.0/24	S0	3	10.0.0.1

Switching/Bridging Loops

When designing a switched or bridged network, you'll almost certainly need to consider redundancy. While network redundancy is a great idea in principle, there are issues that you'll need to take under consideration. The biggest issue is that bridging redundancy exposes networks to a loop, and loops cause major problems if not dealt with properly.

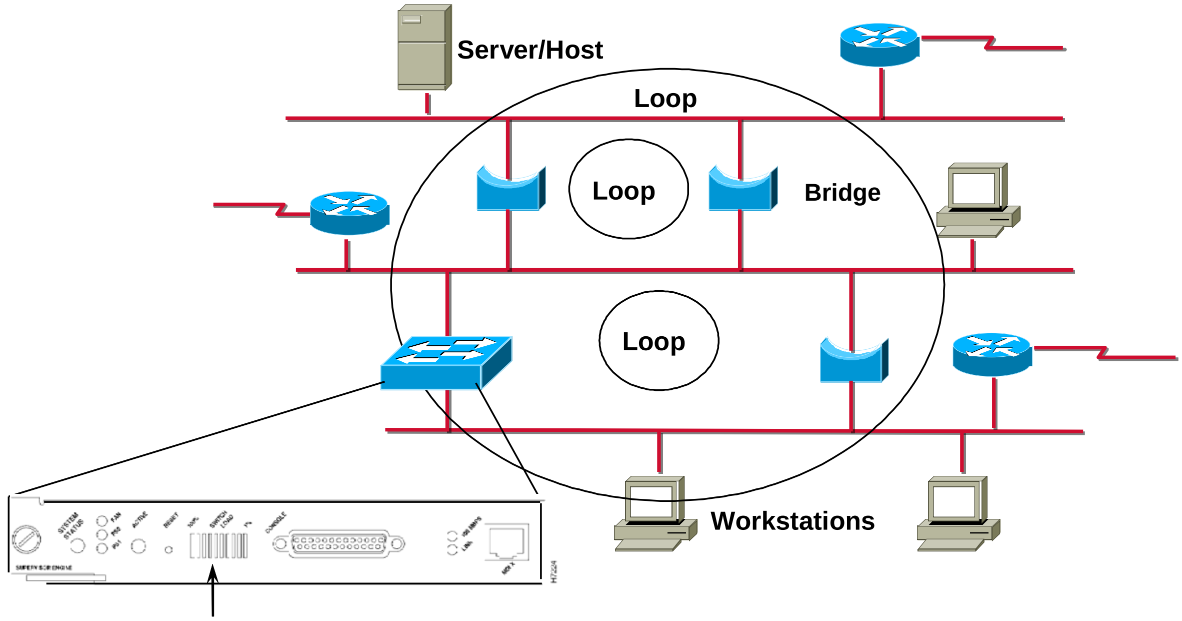
Switching loops occur when there is more than one path between two switches in a computer network. A physical topology that contains switching or bridging loops is necessary for reliability, yet a switched network cannot have loops.

At the Datalink Layer (Layer 2), there is no Time To Live (TTL) mechanism. If a frame enters into a Layer 2 looped topology of switches, it can loop forever.

The problems associated with network loops go back to the days when bridging was the primary way of segmenting a LAN. The idea was to have more than one bridge connecting two segments, in order to provide a redundant path should a link or bridge fail. The problem with having this redundancy in a bridged environment is that may create loops, and network loops are capable of causing communication problems. In the case of a bridging loop, a network becomes susceptible to broadcast storms.

The solution is to allow physical loops, but create a loop free logical topology with the STP (Spanning tree protocol) on the switches.

Avoid Bridging Loops!



When looped, the supervisor engine module switch load LEDs may show

Spyware, Worms and other Malware.

Here is another source of bandwidth consumption. Spyware, worms, Trojans and the like perform scanning operations, install additional protocols and services and communicate with their remote counterparts. These actions frequently account for a significant and sometimes large amount of bandwidth utilization. You should be vigilant in keeping your network free of these threats and bandwidth consumers with antivirus software, firewalls and other protective and/or detection systems.

Misconfiguration.

Even the most seemingly insignificant misconfiguration can cause major problems. Incorrect IP address or subnet mask can render a client unable to communicate. Incorrect DNS, depending on if it is a local DNS server or and internet DNS server can cause unreliable communication. A mis-match in duplex and/or speed settings between a router, switch, server or workstation causes slowness for the client device or the whole network due to dropped packets.

Increased Number of Connections (Network Growth)

As the number of devices (workstations, servers, printers, switches, audio/video equipment, etc.) connected to the network increases so goes the demand for bandwidth. You can balance the load through strategic placement of progressively higher capacity infrastructure devices and segmenting to control areas of high traffic. Prioritizing traffic (also called traffic shaping) according to type, source, destination, etc., can help make better, more efficient use of available bandwidth.

Excessive Protocols Running on the Network.

Protocols that run on the network add additional traffic load as they provide the services and perform their designed tasks. Most protocols running on a network are part of the default installation to insure that those needed are available. In an environment were bandwidth is restricted or in short supply you should determine which protocols are really necessary and which ones are nonessential and disable or remove all nonessential protocols.

Poor Typology Design

The physical connections used to create the networks are sometimes at the root of a network connectivity error. Troubleshooting wiring involves knowing what wiring your network uses and where it is used. When troubleshooting network media consider:

- **Media range (attenuation).** All cables used in networking have certain limitations, in terms of distance. It might be that the network problems are a result of trying to use a cable in an environment or a way for which exceed its designed capabilities. For example, you might find that a network is connecting two workstations that are 130 meters apart with Category 5 UTP cabling. Category 5 UTP specifies distances up to 100 meters, so exceeding the maximum cable length can be a potential cause of the problem. The first step in determining the allowable cable distance is to identify the type of cable used. Determining the cable type is often as easy as reading the cable. Cable from quality manufacturers have stamped on the outer jacket the type, whether it is UTP Category 5, RG-58, or something else.
- **Throughout limitations**. A problem with a particular media may be simply that it cannot accommodate the throughout required by the network. This would create network-wide bottlenecks. It may be necessary to update the network media to correct the problem, for instance, upgrading the network backbone to fiber optic media.
- **Media connectors**. Troubleshooting media requires verifying correct installation of the connectors. In the case of UTP or coaxial, sometimes it may be necessary to swap out a

cable with a known working one to test. For fiber optic cabling, ensure the type of connectors used match the switches and routers used.

Connectivity Problems

Connectivity problems may exists when end stations cannot communicate with other areas of your local area network (LAN) or wide area network (WAN). Using management tools, you can often fix a connectivity problem before users even notice it. Connectivity problems include:

Loss of connectivity

When users cannot initialize access areas of your network or when initial connectivity is lost. Some causes of connectivity loss are:

- **Hardware Failure**

  Hardware Failures such as the network adapter, broken wire in a patch cable, patch cable not connected, patch cable not wired correctly or a failure of a port in a network switch or hub.

- **Misconfiguration**

  Configuration errors prevent communication from being successful even if the hardware is functioning properly. Examples of configuration errors include, incorrect IP address, subnet mask, DNS server IP, wrong version of TCP/IP software (IPv4 vs. IPv6), IPX on a network supporting only IP, incorrect frame type for the network platform/transport protocol, incorrect driver for network adapter, duplex mis-match between the switch and the network adapter, etc.

- **Routing Loops**

  Loops in the network topology can cause a plethora of symptoms, such as slowness that progressively gets worse, lack of communication, IP address conflicts, etc.

Intermittent Connectivity

Broadcast Storms

A broadcast storm means that your network is overwhelmed with constant broadcast or multicast traffic. Broadcast storms can eventually lead to a complete loss of network connectivity as the packets proliferate.

“Broadcast Packets" and "Multicast Packets" are a normal part of your network's operation. To recognize a storm, you must be able to identify when broadcast and multicast traffic is abnormal for your network.

When your network is operating normally, monitor the percentage of broadcast and multicast traffic. You can then use this data as a baseline to determine when broadcast and multicast traffic is too high.

You may suspect that a broadcast storm is occurring when your network response times become extremely slow and network operations are timing out. As a broadcast storm progresses, users cannot log in to servers or access e-mail. As the storm worsens, the network becomes unusable and nearly completely paralyzed.

Outside Interference

Outside interference is something other than internal traffic reducing or preventing communication.

Copper-based media is subject to the effects of Electro-Magnetic Interference and crosstalk interference. UTP cables are particularly susceptible to EMI caused by devices such as power lines, electric motors, fluorescent lighting and so on. Consider using STP cable in environments where cables run through areas where EMI may occur. This includes heating ducts, elevator shafts and through ceilings around lighting fixtures. Crosstalk occurs when cables run in close proximity and the signals from one interfere with the signals on the other. This can be hard to troubleshoot and isolate, so when designing a network be sure to implement measures preventing crosstalk.

Timeout

Timeouts can occur because of high bandwidth utilization or excessive number of hops to the destination. Data packets receive a header value that determines when the packet expires and/or the maximum number of hops it can traverse. This is to prevent packets from bouncing around the internet indefinitely when they don’t reach their destination. If this were to happen, it would not be long before the entire internet would come to a halt.

Tools For Troubleshooting

Ping

Verifies connections to a remote computer or computers. This command is available only if the TCP/IP protocol has been installed.

ping [**-t**] [**-a**] [**-n** *count*] [**-l** *length*] [**-f**] [**-i** *ttl*] [**-v** *tos*] [**-r** *count*] [**-s** *count*] [[**-j** *computer-list*] | [**-k** *computer-list*]] [**-w** *timeout*] *destination-list*

Parameters

-t : Pings the specified computer until interrupted.

-a : Resolves addresses to computer names.

-n *count* **:** Sends the number of ECHO packets specified by *count.* The default is 4.

-l *length* **:** Sends ECHO packets containing the amount of data specified by *length*. The default is 32 bytes; the maximum is 65,527.

-f : Sends a Do not Fragment flag in the packet. The packet will not be fragmented by gateways on the route.

-i *ttl* **:** Sets the Time To Live field to the value specified by *ttl*.

-v *tos* **:** Sets the Type Of Service field to the value specified by *tos*.

-r *count* **:** Records the route of the outgoing packet and the returning packet in the Record Route field. A minimum of 1 and a maximum of 9 computers can be specified by *count*.

-s *count* **:** Specifies the timestamp for the number of hops specified by *count*.

-j *computer-list* **:** Routes packets by way of the list of computers specified by *computer-list*. Consecutive computers can be separated by intermediate gateways (loose source routed). The maximum number allowed by IP is 9.

-k *computer-list* **:** Routes packets by way of the list of computers specified by *computer-list*. Consecutive computers cannot be separated by intermediate gateways (strict source routed). The maximum number allowed by IP is 9.

-w *timeout* **:** Specifies a time-out interval in milliseconds.

destination-list **:** Specifies the remote computers to ping.

Notes

The **ping** command does the following:

- Verifies connections to one or more remote computers by sending ICMP echo packets to the computer and listening for echo reply packets.
- Waits for up to one second for each packet sent.
- Prints the number of packets transmitted and received.

Each received packet is validated against the transmitted message. By default, four echo packets containing 32 bytes of data (a periodic uppercase sequence of alphabetic characters) are transmitted.

You can use **ping** to test both the computer name and the IP address of the computer. If the IP address is verified but the computer name is not, you may have a name resolution problem. In this case, be sure that the computer name you are querying is in either the local Hosts file or in the DNS database.

The following example shows output for **ping**:

```
C:\>ping ds.internic.net
Pinging ds.internic.net [192.20.239.132] with 32 bytes of data:
Reply from 192.20.239.132: bytes=32 time=101ms TTL=243
Reply from 192.20.239.132: bytes=32 time=100ms TTL=243
Reply from 192.20.239.132: bytes=32 time=120ms TTL=243
Reply from 192.20.239.132: bytes=32 time=120ms TTL=243
```

PathPing

A route tracing tool that combines features of the **ping** and **tracert** commands with additional information that neither of those commands provides. The **pathping** command sends packets to each router on the way to a final destination over a period of time, and then computes results based on the packets returned from each hop. Since **pathping** shows the degree of packet loss at any given router or link, you can determine which routers or links might be causing network problems.

If you're having delays on the route to some destination on the Internet you can use the pathping command to attempt to see where the problem is. Pathping combines some of the functionality of both the ping and traceroute (tracert) commands.

Pathping output will first show the hops (routers) you go through to get to the destination IP address (the equivalent of the traceroute functionality) . Then it will send packets to every router on a path for a set period of time and finally will compute statistics based on the packets returned from each router.

Pathping is useful because it allows you to see the amount of loss that occurs at any router and along any link. So you can determine which routers or links might be having network problems.

pathping [**-n**] [**-h** *maximum_hops*] [**-g** *host-list*] [**-p** *period*] [**-q** *num_queries* [**-w** *timeout*] [**-T**] [**-R**] *target_name*

Parameters

-n : Does not resolve addresses to host names.

-h *maximum_hops* **:** Specifies maximum number of hops to search for the target. Default is 30 hops.

-g *host-list* **:** Allows consecutive computers to be separated by intermediate gateways (loose source route) along *host-list*.

-p *period* **:** Specifies number of milliseconds to wait between consecutive pings. Default is 250 milliseconds (1/4 second).

-q *num_queries* **:** Specifies number of queries to each computer along the route. Default is 100.

-w *timeout* **:** Specifies number of milliseconds to wait for each reply. Default is 3000 milliseconds (3 seconds).

-T : Attaches a layer-2 priority tag (for example, 802.1p) to the ping packets that it sends to each of the network devices along the route. This helps identify network devices that do not have layer-2 priority configured. This parameter must be capitalized.

-R : Checks to see if each network device along the route supports the Resource Reservation Setup Protocol (RSVP), which allows the host computer to reserve a certain amount of bandwidth for a data stream. This parameter must be capitalized.

target_name **:** Specifies the destination endpoint, identified either by IP address or host name.

Notes

The **pathping** command performs the equivalent of a traceroute to identify which routers are on the path. It then sends pings periodically to all of the routers over a given time period, and computes statistics based on the number returned from each.

To avoid congestion, pings should be sent at a sufficiently slow interval.

To minimize the effects of burst losses, do not send pings too close together.

-p *period* parameter : Pings are sent to each intermediate hop, one at a time. Therefore, the interval between two pings sent to the same hop is (*period*) x (number of hops).

-w *timeout* parameter : Multiple pings can be done in parallel, so the amount of time specified in the timeout parameter is not bounded by the amount of time specified for the period parameter for waiting between pings.

-T parameter : Enabling layer-2 priority on the host computer allows packets to be sent with a layer-2 priority tag, which can be used by layer-2 devices to assign a priority to the packet. Legacy devices that do not understand layer-2 priority will toss tagged packets, since they will appear as malformed packets. Therefore, a switch that connects to a legacy network should be configured to strip the tag before forwarding the packets. This option helps identify the network elements that are tossing the tagged packets.

The **pathping** command is case-sensitive. This parameter must be capitalized.

-R parameter : An RSVP reservation message for a non-existent session is sent to each network device along the route. If the device is not configured to support RSVP, it returns an Internet Control Message Protocol (ICMP) unreachable message. If it is configured to do RSVP, it returns a Reservation Error. Some devices may not return either of these messages. If this happens, **pathping** returns a timeout message.

The **pathping** command is case-sensitive. This parameter must be capitalized.

The following example shows output for **pathping**:

```
D:\>pathping -n msw

Tracing route to msw [7.54.1.196]
```

```
over a maximum of 30 hops:

  0  172.16.87.35
  1  172.16.87.218
  2  192.68.52.1
  3  192.68.80.1
  4  7.54.247.14
  5  7.54.1.196


Computing statistics for 125 seconds...

Source to Here This Node/Link

Hop  RTT   Lost/Sent = Pct Lost/Sent = Pct  Address
  0 172.16.87.35
  0/ 100 =  0%   |
  1 41ms  0/ 100 = 0%  0/ 100 = 0%  172.16.87.218
  13/ 100 = 13%   |
  2 22ms 16/ 100 = 16% 3/ 100 = 3%  192.68.52.1
  0/ 100 =  0%   |
  3 24ms 13/ 100 = 13% 0/ 100 = 0%  192.68.80.1
  0/ 100 =  0%   |
  4 21ms 14/ 100 = 14% 1/ 100 = 1%  7.54.247.14
  0/ 100 =  0%   |
  5 24ms 13/ 100 = 13% 0/ 100 = 0%  7.54.1.196

Trace complete.
```

Example:

```
pathping -n erasmus.terena.nl
Tracing route to erasmus.terena.nl [192.87.30.2]
over a maximum of 30 hops:
0 128.2.4.1
1 128.2.255.12
2 128.2.33.233
3 192.88.115.185
4 192.88.115.18
5 192.88.115.124
6 198.32.8.84
7 198.32.8.105
8 145.145.166.61
9 145.145.162.2
10 145.145.18.46
11 192.87.30.2
Computing statistics for 125 seconds...

Source to Here This Node/Link
Hop RTT Lost/Sent = Pct Lost/Sent = Pct Address
0 128.2.4.1
0/ 100 = 0% |
1 0.92ms 0/ 100 = 0% 0/ 100 = 0% 128.2.255.12
0/ 100 = 0% |
2 1.08ms 0/ 100 = 0% 0/ 100 = 0% 128.2.33.233
0/ 100 = 0% |
3 1.21ms 0/ 100 = 0% 0/ 100 = 0% 192.88.115.185
0/ 100 = 0% |
4 0.87ms 0/ 100 = 0% 0/ 100 = 0% 192.88.115.18
0/ 100 = 0% |
5 10.36ms 2/ 100 = 2% 0/ 100 = 0% 192.88.115.124
0/ 100 = 0% |

6 27.17ms 0/ 100 = 0% 0/ 100 = 0% 198.32.8.84
0/ 100 = 0% |
7 103.96ms 5/ 100 = 5% 0/ 100 = 0% 198.32.8.105
22/ 100 = 22% |
8 104.46ms 12/ 100 = 12% 0/ 100 = 0% 145.145.166.61
0/ 100 = 0% |
9 183.25ms 10/ 100 = 10% 1/ 100 = 1% 145.145.162.2
0/ 100 = 0% |
10 104.57ms 16/ 100 = 16% 0/ 100 = 0% 145.145.18.46
0/ 100 = 0% |
11 103.73ms 14/ 100 = 14% 0/ 100 = 0% 192.87.30.2
Trace complete.
```

In the sample report above, the This Node/Link, Lost/Sent = Pct and Address columns show that the link between 198.32.8.105 and 145.145.166.61 is dropping 22 percent of the packets. The router at hops 9 is also a dropping packets addressed to it, but this loss does not affect its ability to forward traffic that is not addressed to it.

The loss rates displayed for the links, identified as a vertical bar (|) in the Address column, indicate link congestion that is causing the loss of packets that are being forwarded on that path. The loss rates displayed for routers indicate that these routers might be overloaded.

When **pathping** is run, the first results list the route. This is the same path that is shown using **tracert**. Next, **pathping** displays a busy message for approximately the next minute and a half (the exact time varies by the hop count). During this time, **pathping** gathers information from all the routers previously listed and from the links between them. At the end of this period, it displays the test results.

In the sample report above, the **This Node/Link Lost/Sent = Pct** and **Address** columns show that the link between 172.16.87.218 and 192.68.52.1 is dropping 13% of the packets. The routers at hops 2 and 4 also are dropping packets addressed to them, but this loss does not affect their forwarding path.

The loss rates displayed for the links (identified as **|** in the **Address** column) indicate link congestion causing the loss of packets being forwarded along the path. The loss rates displayed for routers (identified by their IP addresses) indicate that the CPUs or local packet buffers of those routers might be overloaded.

Tracert

This diagnostic utility determines the route taken to a destination by sending Internet Control Message Protocol (ICMP) echo packets with varying Time-To-Live (TTL) values to the destination. Each router along the path is required to decrement the TTL on a packet by at least 1 before forwarding it, so the TTL is effectively a hop count. When the TTL on a packet reaches 0, the router is supposed to send back an ICMP Time Exceeded message to the source system. **Tracert** determines the route by sending the first echo packet with a TTL of 1 and incrementing the TTL by 1 on each subsequent transmission until the target responds or the maximum TTL is reached. The route is determined by examining the ICMP Time Exceeded messages sent back by intermediate routers. However, some routers silently drop packets with expired TTL values and are invisible to **tracert**.

tracert [**-d**] [**-h** *maximum_hops*] [**-j** *computer-list*] [**-w** *timeout*] *target_name*

Parameters

-d : Specifies not to resolve addresses to computer names.

-h *maximum_hops* **:** Specifies maximum number of hops to search for target.

-j *computer-list* **:** Specifies loose source route along *computer-list*.

-w *timeout* **:** Waits the number of milliseconds specified by *timeout* for each reply.

target_name **:** Name of the target computer.

Netstat

Netstat is a useful tool for checking network and Internet connections. Some useful applications for the average PC user are considered, including checking for malware connections. Netstat displays the active TCP connections, the ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics (for IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols). When used without parameters, **netstat** displays active TCP connections.

Syntax and switches

The command syntax is `netstat [-a] [-b] [-e] [-f] [-n] [-o] [-p proto] [-r] [-s] [-t] [-v] [interval]` A brief description of the switches is given in Table I below. Some switches are only in certain Windows versions, as noted in the table..*Note that switches for Netstat use the dash symbol "-" rather than the slash "/"*.

Table I. Switches for Netstat command

Switch	Description
-a	Displays all connections and listening ports
-b	Displays the executable involved in creating each connection or listening port. (Added in XP SP2.)
-e	Displays Ethernet statistics

-f	Displays Fully Qualified Domain Names for foreign addresses. (In Windows Vista/7 only)
-n	Displays addresses and port numbers in numerical form
-o	Displays the owning process ID associated with each connection
-p proto	Shows connections for the protocol specified by proto; proto may be any of: TCP, UDP, TCPv6, or UDPv6.
-r	Displays the routing table
-s	Displays per-protocol statistics
-t	Displays the current connection offload state, (Windows Vista/7)
-v	When used in conjunction with -b, will display sequence of components involved in creating the connection or listening port for all executables. (Windows XP SP2, SP3)
[interval]	An integer used to display results multiple times with specified number of seconds between displays. Continues until stopped by command *ctrl+c*. Default setting is to display once,

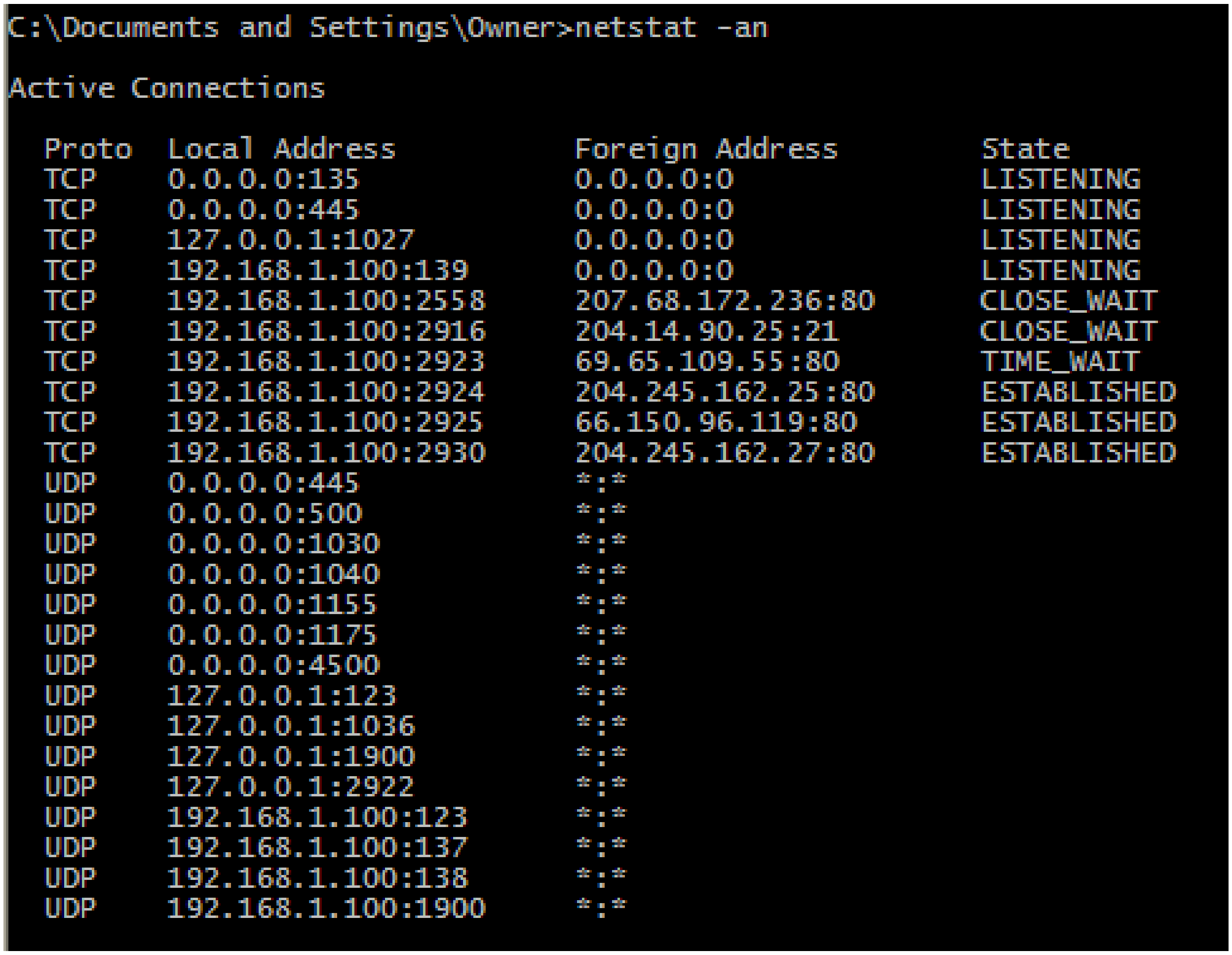
Applications of Netstat

Netstat is one of a number of command-line tools available to check the functioning of a network. (See this page for discussion of other tools.) It provides a way to check if various aspects of TCP/IP are working and what connections are present. In Windows XP SP2, a new switch "-B" was added that allows the actual executable file that has opened a connection to be displayed. This newer capability provides a chance to catch malware that may be phoning home or using your computer in unwanted ways on the Internet. There are various ways that a system administrator might use the assortment of switches but I will give two examples that might be useful to home PC users.

Checking TCP/IP connections

TCP and UDP connections and their IP and port addresses can be seen by entering a command combining two switches: netstat -an An example of the output that is obtained is shown in Figure 1.

Figure 1. Example output for command "netstat -an"



```
C:\Documents and Settings\Owner>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1027         0.0.0.0:0              LISTENING
  TCP    192.168.1.100:139      0.0.0.0:0              LISTENING
  TCP    192.168.1.100:2558     207.68.172.236:80      CLOSE_WAIT
  TCP    192.168.1.100:2916     204.14.90.25:21        CLOSE_WAIT
  TCP    192.168.1.100:2923     69.65.109.55:80        TIME_WAIT
  TCP    192.168.1.100:2924     204.245.162.25:80      ESTABLISHED
  TCP    192.168.1.100:2925     66.150.96.119:80       ESTABLISHED
  TCP    192.168.1.100:2930     204.245.162.27:80      ESTABLISHED
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:500            *:*
  UDP    0.0.0.0:1030           *:*
  UDP    0.0.0.0:1040           *:*
  UDP    0.0.0.0:1155           *:*
  UDP    0.0.0.0:1175           *:*
  UDP    0.0.0.0:4500           *:*
  UDP    127.0.0.1:123          *:*
  UDP    127.0.0.1:1036         *:*
  UDP    127.0.0.1:1900         *:*
  UDP    127.0.0.1:2922         *:*
  UDP    192.168.1.100:123      *:*
  UDP    192.168.1.100:137      *:*
  UDP    192.168.1.100:138      *:*
  UDP    192.168.1.100:1900     *:*
```

The information that is displayed includes the protocol, the local address, the remote (foreign) address, and the connection state. Note that the various IP addresses include port information as well. An explanation of the different connection states is given in Table II>

Table II. Description of various connection states

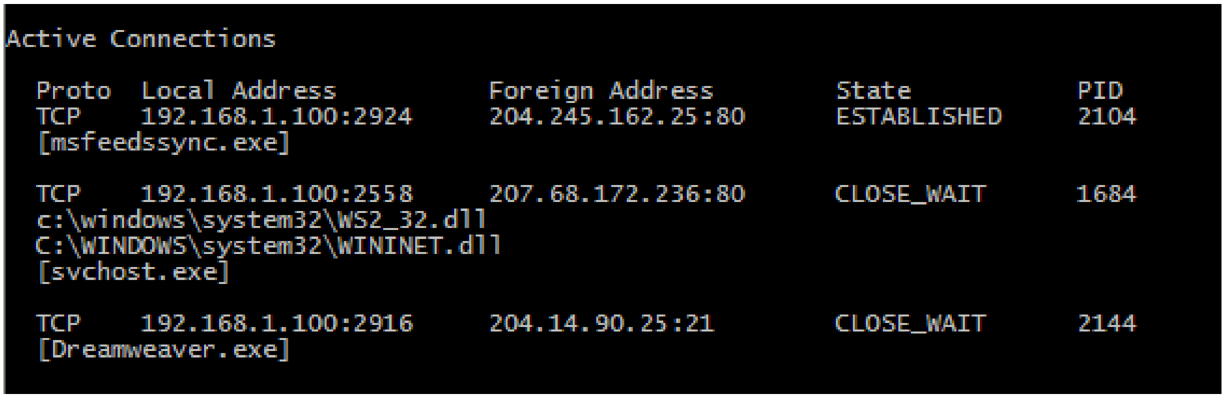
State	Description
CLOSED	Indicates that the server has received an ACK signal from the client and the connection is closed
CLOSE_WAIT	Indicates that the server has received the first FIN signal from the client and the connection is in the process of being closed
ESTABLISHED	Indicates that the server received the SYN signal from the client and the session is established
FIN_WAIT_1	Indicates that the connection is still active but not currently being used

FIN_WAIT_2	Indicates that the client just received acknowledgment of the first FIN signal from the server
LAST_ACK	Indicates that the server is in the process of sending its own FIN signal
LISTENING	Indicates that the server is ready to accept a connection
SYN_RECEIVED	Indicates that the server just received a SYN signal from the client
SYN_SEND	Indicates that this particular connection is open and active
TIME_WAIT	Indicates that the client recognizes the connection as still active but not currently being used

Checking for malware by looking at which programs initiate connections

To find out which programs are making connections with the outside world, we can use the command netstat -b (Note that for Windows Vista/7, this particular switch requires that the command prompt have elevated privileges.) Actually, it is better to check over a period of time and we can add a number that sets the command to run at fixed intervals. Also, it is best to create a written record of the connections that are made over some period of time. The command can then be written netstat -b 5 >> C:\connections.txt Note that as written, this command will run with five-second intervals until stopped by entering "*Ctrl+c*", which is a general command to exit. (Some reports say that this can be fairly CPU intensive so it may cause a slower, single-core machine to run sluggishly. It was not noticeable on my dual-core machine.) A simple example of the type of output is shown in Figure 2. Note that the Process ID (PID) is given when using Windows XP. In Windows Vista/7, the switch "o' has to be added to display PIDs. This command can be combined with other tools such as Task Manager to analyze what executable files and processes are active and are trying to make Internet connections.

Figure 2. Sample output for command "netstat -b" in Windows XP



```
Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    192.168.1.100:2924     204.245.162.25:80      ESTABLISHED     2104
  [msfeedssync.exe]

  TCP    192.168.1.100:2558     207.68.172.236:80      CLOSE_WAIT      1684
  c:\windows\system32\WS2_32.dll
  C:\WINDOWS\system32\WININET.dll
  [svchost.exe]

  TCP    192.168.1.100:2916     204.14.90.25:21        CLOSE_WAIT      2144
  [Dreamweaver.exe]
```

Windows XP batch program to check connections and terminate automatically

The previous example of using "netstat -b" to check connections at intervals has the disadvantage that it requires manual termination. It is also possible to use a batch file that runs a specified number of times with a given time interval and then terminates automatically. In Windows XP we can make use of a command from the Windows 2003 Server Tools called "Sleep". A possible batch file is:

```
@echo off
echo Checking connections
for /L %%X in (1,1,100) do (netstat -b >> C:\connections.txt)&&(sleep 5)
```

This particular example does 100 iterations of the *netstat* command at 30 second intervals and writes the results to a file *C:\connections.txt*. By using different combinations of the switches in Table I, the type of output can be varied

Batch program to check connections in Windows Vista and Windows 7

Windows Vista and Windows 7 do not require installing the "Sleep" file. A command " timeout" has been added to these operating systems that serves a similar purpose. A possible batch file for Windows Vista/7 is:

```
@echo off
echo Checking connections
for /L %%X in (1,1,100) do (netstat -b >> "%USERPROFILE%\connections.txt")&&
((timeout /t 5 /nobreak)>nul)
```

This batch file has to be run with administrator privileges.

Examples

To display both the Ethernet statistics and the statistics for all protocols, type the following command:

netstat -e -s

To display the statistics for only the TCP and UDP protocols, type the following command:

netstat -s -p tcp udp

To display active TCP connections and the process IDs every 5 seconds, type the following command:

nbtstat -o 5

To display active TCP connections and the process IDs using numerical form, type the following command:

nbtstat -n -o

Netdiag.exe

This command-line diagnostic tool helps to isolate networking and connectivity problems by performing a series of tests to determine the state of your network client. These tests and the key network status information they expose give network administrators and support personnel a more direct means of identifying and isolating network problems. Moreover, because this tool does not require parameters or switches to be specified, support personnel and network administrators can focus on analyzing the output rather than on training users how to use the tool.

Notes

- You must run NetDiag from the command window.
- TCP/IP must be bound to one or more adapters before using this tool. In most cases, this binding already exists

To download Netdiag.exe, visit the following Microsoft Web site:
http://support.microsoft.com/kb/927229

Netdiag.exe uses the following syntax:

netdiag [/q] [/v] [/l] [/debug] [/d:*domain_name***] [/fix] [/dcaccountenum] [/test**:*test_name***] [/skip:***test_name***]**

You can use the following parameters with Netdiag.exe:

/q: Use this parameter to specify quiet output and display errors only.

/v: Use this parameter to run Netdiag.exe in verbose mode and to display information about the actions that are performed.

/l: Use this parameter to write output to the Netdiag.log file. The Netdiag.log file is created in the same folder in which you run Netdiag.exe.

/debug: Use this parameter to run Netdiag.exe in debug mode. This parameter specifies a more verbose output than when you use the **/v** parameter.

/d:*domain_name*: Use this parameter to locate a domain controller in the specified domain.

/fix: Use this parameter to correct issues with Domain Name System (DNS), domain controller tests and other issues, such as the following:

DNS Test
If the computer is a domain controller, Netdiag.exe verifies all the DNS entries in the Netlogon.dns file to determine whether they are correct. Additionally, Netdiag.exe updates the appropriate entries if there is a problem.

Domain Controller Tests
If the domain GUID that is cached in a local computer which is on your primary domain is different from the domain GUID that is saved in a domain controller, Netdiag.exe will try to update the domain GUID that is cached on the local computer.

/dcaccountenum: Use this parameter to enumerate the computer accounts of the domain controller.

/test:*test_name*: Use this parameter to specify the test or tests that you want to run, where *test_name* can be any one of the following values:

Autonet: Automatic Private IP Addressing (APIPA) address test

Bindings: Bindings test

Browser: Redir and Browser test

DcList: Domain controller list test

DefGw: Default gateway test

DNS: Domain Name Service (DNS) test

DsGetDc: Domain controller discovery test

IpConfig: IP address configuration test

IpLoopBk: IP address loopback ping test

IPSec: Internet Protocol security (IPSec) security test

IPX: Internetwork Packet Exchange (IPX) test

Kerberos: Kerberos Test

Ldap: Lightweight Directory Access Protocol (LDAP) test

Member: Domain membership test

Modem: Modem diagnostics test

NbtNm: NetBIOS over TCP/IP (NetBT) name test

Ndis: Netcard queries test

NetBTTransports: NetBT transports test

Netstat: Netstat information test

NetWare: NetWare test

Route: Routing table test

Trust: Trust relationship test

WAN: Wide Area Network (WAN) configuration test

WINS: Windows Internet Naming Services (WINS) service test

Winsock: Winsock test

To specify two or more tests, separate each **/test:*test_name*** item with a space. Note that the tests that you cannot skip will still be run.

/skip:*test_name*: Use this parameter to specify the test or tests that you do not want to run, where *test_name* can be any one of the tests listed earlier in the **/test:*test_name*** list.

To specify two or more tests, separate each **/skip:*test_name*** item with a space.

Examples

To run Netdiag.exe in verbose mode, type the following line at the command prompt and then press <Enter>:

netdiag /v

To use Netdiag.exe to display information about the domain controller that is in your domain, type the following line and then press <Enter>:

netdiag /v /l /test:dsgetdc

Write the information about the domain controller that is in your domain to the Netdiag.log file. The Netdiag.log file is located in the folder in which Netdiag.exe is run.

To use Netdiag.exe to display the currently active IPSec policy, type the following line and then press <Enter>:

netdiag /test:ipsec /debug

1. To test the domain controller in your domain, type the following line, and then press ENTER: netdiag /v /l /test:dsgetdc

2. To display the IPSec policy, type the following line, and then press <Enter>: netdiag /test:ipsec /debug

"[Fatal] Failed to Get System Information" Error Message Occurs When You Run Netdiag

The procedure entry point DnsGetMaxNumberOfAddressToRegister could not be located in the dynamic link library DNSAPI.dll." error when running netdiag on XP.

Symptom: When you run the Netdiag tool, you receive the following error message: [Fatal] Failed to get system information of this machine.

Resolution: To resolve this issue, start the Computer Management console, click **Services**, and then start the **Remote Registry service**.

"The procedure entry point DnsGetMaxNumberOfAddressToRegister could not belocated in the dynamiclink library DNSAPI.dll." error when running netdiag on XP

Make sure you are running the version form XP CD. There is a possibility that you have a W2K version of netdiag installed explicitly - there was a download from KB. Or you may just upgrade the OS from w2k to XP. To fix this problem, you may want to re-install it. To re-install, run XP CD at \Support\Tools\SUPTOOLS.MSI and select "Remove all". After it is done, launch it again to install appropriate version of support tools.

What does netdiag /fix do

Netdiag /fix switch is very useful tool to correct issues with DNS and domain controller tests. 1. DNS Test: If the computer is a domain controller, Netdiag verifies all the DNS entries in the Netlogon.dns file to determine if they are correct and updates the appropriate entries if there is **a problem. 2. Domain Controller Test: If the domain GUID cached in a local computer on your** primary domain is different than the domain GUID saved in a domain controller, Netdiag tries to update the domain GUID on the local computer.

Nslookup

The **nslookup** command can be used in Windows and Unix to find the IP addresses of a particular computer, using DNS lookup. The acronym stands for "name server lookup". Before using this tool, you should be familiar with how DNS works. The Nslookup command-line tool is available only if you have installed the TCP/IP protocol.

Syntax

nslookup [**-***SubCommand ...*] [{*ComputerToFind*| [**-***Server*]}]

Parameters

-***SubCommand ...* :** Specifies one or more **nslookup** subcommands as a command-line option. For a list of subcommands, see Related Topics.

***ComputerToFind* :** Looks up information for *ComputerToFind* using the current default DNS name server, if no other server is specified. To look up a computer not in the current DNS domain, append a period to the name.

***-Server* :** Specifies to use this server as the DNS name server. If you omit *-Server*, the default DNS name server is used.

{help|?} : Displays a short summary of **nslookup** subcommands.

Remarks

- If *ComputerToFind* is an IP address and the query is for an A or PTR resource record type, the name of the computer is returned. If *ComputerToFind* is a name and does not have a trailing period, the default DNS domain name is appended to the name. This behavior depends on the state of the following **set** subcommands: **domain**, **srchlist**, **defname**, and **search**.
- If you type a hyphen (-) instead of *ComputerToFind*, the command prompt changes to **nslookup** interactive mode.
- The command-line length must be less than 256 characters.
- **Nslookup** has two modes: interactive and noninteractive.

  If you need to look up only a single piece of data, use noninteractive mode. For the first parameter, type the name or IP address of the computer that you want to look up. For the second parameter, type the name or IP address of a DNS name server. If you omit the second argument, **nslookup** uses the default DNS name server.

  If you need to look up more than one piece of data, you can use interactive mode. Type a hyphen (-) for the first parameter and the name or IP address of a DNS name server for the second parameter. Or, omit both parameters and **nslookup** uses the default DNS name server. Following are some tips about working in interactive mode:

  To start Nslookup.exe in interactive mode, simply type "nslookup" at the command prompt:

  ```
  C:\> nslookup Default Server:

  nameserver1.domain.com

  Address: 10.0.0.1 >
  ```

- To interrupt interactive commands at any time, press CTRL+B.
- To exit, type **exit**.
- To treat a built-in command as a computer name, precede it with the escape character (\).
- An unrecognized command is interpreted as a computer name.

If the lookup request fails, **nslookup** prints an error message. The following table lists possible error messages.

Error message	Description
Timed out	The server did not respond to a request after a certain amount of time and a certain number of retries. You can set the time-out period with the **set timeout** subcommand. You can set the number of retries with the **set retry** subcommand.
No response from server	No DNS name server is running on the server computer.
No records	The DNS name server does not have resource records of the current query type for the computer, although the computer name is valid. The query type is specified with the **set querytype** command.
Nonexistent domain	The computer or DNS domain name does not exist.
Connection refused -or- Network is unreachable	The connection to the DNS name server or finger server could not be made. This error commonly occurs with **ls** and **finger** requests.
Server failure	The DNS name server found an internal inconsistency in its database and could not return a valid answer.
Refused	The DNS name server refused to service the request.
Format error	The DNS name server found that the request packet was not in the proper format. It may indicate an error in **nslookup**.

Examples

Each command-line option consists of a hyphen (-) followed immediately by the command name and, in some cases, an equal sign (=) and then a value. For example, to change the default query type to host (computer) information and the initial time-out to 10 seconds, type:

nslookup -querytype=hinfo -timeout=10

```
$ nslookup www.wikipedia.org
```

```
Server:  ns0.southern.edu
Address:  216.229.224.4

Non-authoritative answer:
Name:    www.wikipedia.org
Addresses:        207.142.131.248,    207.142.131.235,    207.142.131.236,
207.142.131.245 207.142.131.246, 207.142.131.247
```

Nbtstat

Displays NetBIOS over TCP/IP (NetBT) protocol statistics, NetBIOS name tables for both the local computer and remote computers, and the NetBIOS name cache. **Nbtstat** allows a refresh of the NetBIOS name cache and the names registered with Windows Internet Name Service (WINS). Used without parameters, **nbtstat** displays help.

Syntax

nbtstat [**-a** *RemoteName*] [**-A** *IPAddress*] [**-c**] [**-n**] [**-r**] [**-R**] [**-RR**] [**-s**] [**-S**] [*Interval*]

Parameters

-a *RemoteName* : Displays the NetBIOS name table of a remote computer, where *RemoteName* is the NetBIOS computer name of the remote computer. The NetBIOS name table is the list of NetBIOS names that corresponds to NetBIOS applications running on that computer.

-A *IPAddress* : Displays the NetBIOS name table of a remote computer, specified by the IP address (in dotted decimal notation) of the remote computer.

-c : Displays the contents of the NetBIOS name cache, the table of NetBIOS names and their resolved IP addresses.

-n : Displays the NetBIOS name table of the local computer. The status of **Registered** indicates that the name is registered either by broadcast or with a WINS server.

-r : Displays NetBIOS name resolution statistics. On a Windows XP computer that is configured to use WINS, this parameter returns the number of names that have been resolved and registered using broadcast and WINS.

-R : Purges the contents of the NetBIOS name cache and then reloads the #PRE-tagged entries from the Lmhosts file.

-RR : Releases and then refreshes NetBIOS names for the local computer that is registered with WINS servers.

-s : Displays NetBIOS client and server sessions, attempting to convert the destination IP address to a name.

-S : Displays NetBIOS client and server sessions, listing the remote computers by destination IP address only.

***Interval* :** Redisplays selected statistics, pausing the number of seconds specified in *Interval* between each display. Press CTRL+C to stop redisplaying statistics. If this parameter is omitted, **nbtstat** prints the current configuration information only once.

/? : Displays help at the command prompt.

Remarks

- **Nbtstat** command-line parameters are case-sensitive.
- The following table describes the column headings that are generated by **nbtstat**.

Heading	Description
Input	The number of bytes received.
Output	The number of bytes sent.
In/Out	Whether the connection is from the computer (outbound) or from another computer to the local computer (inbound).
Life	The remaining time that a name table cache entry will live before it is purged.
Local Name	The local NetBIOS name associated with the connection.
Remote Host	The name or IP address associated with the remote computer.
<03>	The last byte of a NetBIOS name converted to hexadecimal. Each NetBIOS name is 16 characters long. This last byte often has special significance because the same name might be present several times on a computer, differing only in the last byte. For example, <20> is a space in ASCII text.
Type	The type of name. A name can either be a unique name or a group name.
Status	Whether the NetBIOS service on the remote computer is running (Registered) or a duplicate computer name has registered the same service (Conflict).
State	The state of NetBIOS connections.

The following table describes the possible NetBIOS connection states.

State	Description
Connected	A session has been established.
Associated	A connection endpoint has been created and associated with an IP address.
Listening	This endpoint is available for an inbound connection.
Idle	This endpoint has been opened but cannot receive connections.
Connecting	A session is in the connecting phase and the name-to-IP address mapping of the destination is being resolved.

Accepting	An inbound session is currently being accepted and will be connected shortly.
Reconnecting	A session is trying to reconnect (it failed to connect on the first attempt).
Outbound	A session is in the connecting phase and the TCP connection is currently being created.
Inbound	An inbound session is in the connecting phase.
Disconnecting	A session is in the process of disconnecting.
Disconnected	The local computer has issued a disconnect and it is waiting for confirmation from the remote system.

This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections

Examples

To display the NetBIOS name table of the remote computer with the NetBIOS computer name of CORP07, type:

nbtstat -a CORP07

To display the NetBIOS name table of the remote computer assigned the IP address of 10.0.0.99, type:

nbtstat -A 10.0.0.99

To display the NetBIOS name table of the local computer, type:

nbtstat -n

To display the contents of the local computer NetBIOS name cache, type:

nbtstat -c

To purge the NetBIOS name cache and reload the #PRE-tagged entries in the local Lmhosts file, type:

nbtstat -R

To release the NetBIOS names registered with the WINS server and re-register them, type:

nbtstat -RR

To display NetBIOS session statistics by IP address every five seconds, type:

nbtstat -S 5

Packet sniffer

A packet sniffer (also known as a network analyzer or protocol analyzer or, for particular types of networks, an Ethernet sniffer or wireless sniffer) is computer software or computer hardware that can intercept and log traffic passing over a digital network or part of a network. As data streams travel back and forth over the network, the sniffer captures each packet and eventually decodes and analyzes its content according to the appropriate RFC or other specifications.

Capabilities

On wired broadcast LANs, depending on the network structure (hub or switch), one can capture traffic on all or just parts of the traffic from a single machine within the network; however, there are some methods to avoid traffic narrowing by switches to gain access to traffic from other systems on the network (e.g. ARP spoofing). For network monitoring purposes it may also be desirable to monitor all data packets in a LAN by using a network switch with a so-called *monitoring port*, whose purpose is to mirror all packets passing through all ports of the switch.

On wireless LANs, one can capture traffic on a particular channel.

On wired broadcast and wireless LANs, in order to capture traffic other than unicast traffic sent to the machine running the sniffer software, multicast traffic sent to a multicast group to which that machine is listening, and broadcast traffic, the network adapter being used to capture the traffic must be put into promiscuous mode; some sniffers support this, others don't. On wireless LANs, even if the adapter is in promiscuous mode, packets not for the service set for which the adapter is configured will usually be ignored; in order to see those packets, the adapter must be put into monitor mode.

Uses

The versatility of packet sniffers means they can be used to:

- Analyze network problems.
- Detect network intrusion attempts.
- Gain information for effecting a network intrusion.
- Monitor network usage.
- Gather and report network statistics.
- Filter suspect content from network traffic.
- Spy on other network users and collect sensitive information such as passwords (depending on any content encryption methods which may be in use)
- Reverse engineer protocols used over the network.
- Debug client/server communications.

APPENDIX A:

10 tips for troubleshooting slowdowns in small business networks

Network congestion and slowdowns--whether caused by faulty hardware, negligent users, viruses or spyware applications gone wild, or other factors--lead to serious headaches for network administrators and support personnel. By keeping a wary eye tuned for the following 10 items, IT professionals can help prevent the most common causes of network slowdowns.

#1: Bad NICs

Intermittent network errors, particularly those isolated to a specific workstation or server, can often be traced to a failing network interface card. When you believe a network adapter may be failing, visually inspect the card's LED link lights.

A solid green (or amber) LED indicates the NIC has a good active physical connection with another network device, such as a network switch or router (blinking LEDs typically indicate the NIC possesses an active connection and is processing network traffic). If the LED is not lit green, it's likely the network adapter is disabled within Windows or doesn't have an active connection to the network. It's also possible the cable plugged into the NIC is connected to a nonfunctioning wall-jack or faulty network port.

If you can rule out nonfunctioning wall-jacks and faulty network ports (the easiest method of doing so is to connect the same network connection to a laptop known to have a properly functioning network adapter), and if the network adapter is properly enabled and configured in Windows, it's likely the NIC is bad and requires replacement.

#2: Failing switches/routers

Many network slowdowns are foreshadowed by strange occurrences. For example, regular Web traffic may work properly, but e-mail may stop functioning. Or, regular Web traffic may work properly but attempts to connect to any secure (HTTPS) sites may fail. In other cases, Internet access simply ceases across the board.

Often the best remedy for inconsistent network outages and/or slowdowns is to reboot or power cycle the network's routers/switches. If local network connectivity exists (if users can view and access network shares) but they are not receiving e-mail from external users or cannot access the Internet, rebooting or power cycling the WAN modem can often return the network to proper operation.

If you're having to reboot or power cycle a piece of network equipment consistently, make sure that it's connected to a quality uninterruptible power supply. Power fluctuations often result in confused switches and routers. If a network device is connected to a good UPS and still frequently experiences trouble, it may be necessary to replace the failing switch, router, or modem.

#3: Daisy chaining

As organizations grow, particularly small businesses, outside IT contractors often implement simple solutions. Many consultants choose to simply add a five-port router to an existing four-port router/firewall. Small businesses everywhere boast just such a setup.

However, as switches are added to a network, data packets must navigate additional hops to reach their destination. Each hop complicates network routing. Depending upon the amount of traffic a network must support--and even a small dentist's or doctor's office can easily stress 10/100 Mbps systems due to X-ray imagery, patient file information, and other data--the addition of an extra hop or two can spell the difference between a smooth running network and one that frequently slows employee productivity to unacceptable levels.

Resist the urge to daisy chain multiple network switches and routers. Instead, plan for capacity. Or if unforeseen growth has resulted in successive connected switches, eliminate as many devices as possible through consolidation to a more potent and scalable unit.

#4: NetBIOS conflicts

NetBIOS, still in use on many Windows NT 4.0 networks in particular, contains many built-in processes to catch and manage conflicts. Occasionally, however, those processes don't handle conflicts properly. The result can be inaccessible file shares, increased network congestion, and even outages.

Guard against NetBIOS conflicts by ensuring older Windows systems all receive the most recent service packs. In some cases, Windows NT 4.0 systems having different service packs will generate telltale NetBT (ID 4320) errors.

Strange network behavior can also occur when two systems are given the same computer name or when two systems both believe they serve the master browser role. Sometimes the error will log itself as Event ID 8003 in a server's system log. Disabling WINS/NetBT name resolution (only if it's not required) can eliminate such issues.

If disabling NetBT is not an option, such errors can often be eliminated by identifying the second system that has the same computer name within the same domain and giving it a new name or by restarting the Netlogon Service on the domain controller. Yet another option for eliminating legacy NetBT issues is to search a system's LMHOSTS file for inaccurate or outdated entries. Some IT professionals claim they've solved similar errors by disabling and re-enabling the NIC on the offending system.

#5: IP conflicts

Windows typically prevents two devices with the same IP address from logging on to the same network (when using DHCP). But occasionally, two systems with the same address wind up on the same network. For example, one system could receive an address automatically, while another computer logs on using a static address specified by a user. When such conflicts occur, network slowdowns result (and the systems sharing the same address frequently experience outages).

Troubleshoot IP address conflicts by ensuring you don't have a rogue DHCP server on the network. Confirm, too, that configured DHCP scopes don't contain overlapping or duplicate entries and that any systems (such as servers and routers) that have been assigned static IP addresses have been excluded from the DHCP pools.

#6: Excessive network-based applications

Occasionally, networks are overrun by the applications they power. For example, a physician's office that uses a Web-based patient and practice application will commonly have every workstation logged on to the program during business hours. Retrieving data from the patient database and consistent monitoring of appointment and scheduling information alone can place stress on even a well-architected network.

Add in the fact that each workstation is likely tuned to e-mail (and many offices are turning to VoIP) and it's easy to see how introducing a few streaming audio/video files to the mix (either in the form of online music services, news videos, or instructional medical presentations and Webinars) can unacceptably slow a 10/100 Mbps network's performance.

Implement policies--and if necessary, hardware-based Web filtering tools--to prevent applications from overwhelming available network bandwidth. Make sure employees understand they're not to stream unnecessary audio and video files. Further, when working with VoIP, be sure adequate data pipes are in place to manage both voice and data traffic.

#7: Spyware infestation

Spyware, the scourge of the last few years, finally appears to be meeting its match in business environments. The development of potent anti-spyware tools, combined with effective end user policies, is reducing the impact of spyware in many organizations. Windows Vista includes Defender, a decent anti-spyware application powered by the popular Giant engine.

However, infestations still occur, particularly on older systems that haven't been properly safeguarded. Implement strong user policies and either gateway-based protection or individual client applications to prevent spyware programs from consuming precious network bandwidth.

#8: Virus infestation

Just as spyware is proving containable within business environments, so too are viruses. That said, despite an administrator's best efforts--including firewall deployment, routine and consistent Windows patching, and the use of regularly updated antivirus programs--viruses do get through. The result can bring a network to a standstill.

For example, many viruses place Trojan programs on Windows systems, where they can wreak havoc. In addition to leveraging a system's ability to send e-mail to forward hundreds (if not thousands) of spam messages an hour, viruses can corrupt network configuration.

Defend against virus threats to network performance by ensuring firewalls, Windows updates, and antivirus programs are properly configured and maintained.

#9: Insufficient bandwidth

Sometimes, a network just doesn't have the throughput it requires. As with # 6--excessive network-based applications--some environments demand more bandwidth than others.

When a network does bog down, several options typically exist for increasing capacity. Besides boosting up- and downstream speeds, some offices may require additional dedicated connections. From multiple T1s to DS3s to even optical carrier-grade connectivity, many potential solutions exist.

Further, some organizations may need to upgrade existing 10/100 Mbps networks to gigabit speeds. By upgrading NICs, cabling, and devices to 10/100/1000 Mbps equipment--and replacing any remaining hubs with switches--many firms can realize significant capacity gains. In other cases, it may be necessary to subnet networks to localize particularly intense traffic to specific network segments.

#10: DNS errors

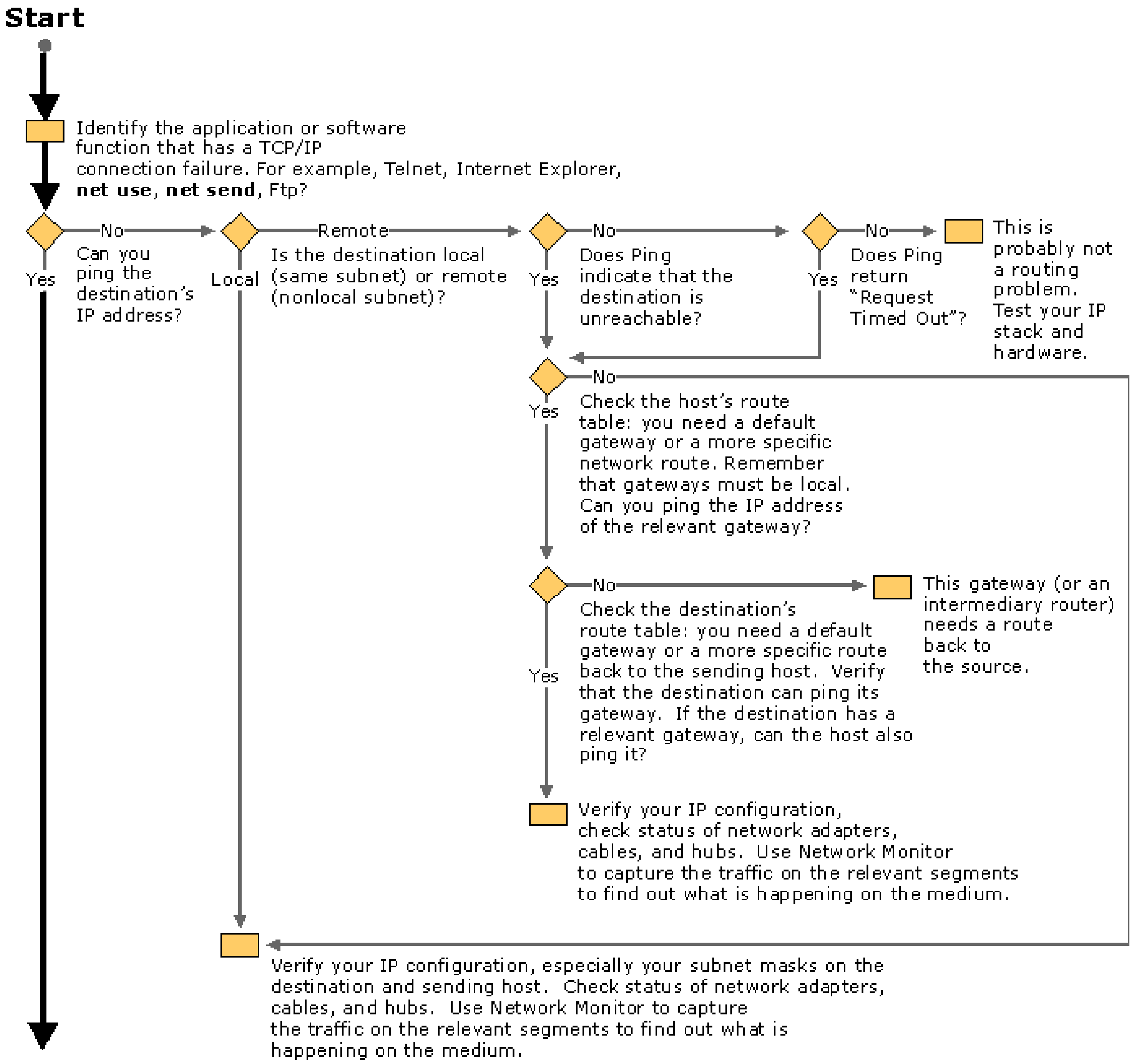
DNS configuration errors can lead to numerous network failures and generalized slow performance. When no DNS server is available on a local LAN, local systems may have trouble finding one another or accessing local resources because they'll have trouble finding service locator records that assist Windows systems in communicating with Active Directory. Worse, systems with no local DNS server or those workstations having DNS servers several hops away may experience delays or flat outages in accessing Web sites and extranets.

Try placing DNS servers as close to network systems as possible. Although adding DNS services to existing servers places greater demand on those boxes, properly configured machines can remain secure and noticeably enhance response times to external resources.

Also, always check to ensure systems are configured to use the proper DNS servers. Network architectures change over time, yet older workstations (particularly those set to use static addressing) occasionally are forgotten and continue operating using outdated DNS settings. As your organization and ISP update DNS systems, be sure workstations and other routing equipment actually receive the updates.

APPENDIX B

Network Troubleshooting Flowchart



Identify the application or software function that has a TCP/IP connection failure. For example, Telnet, Internet Explorer, **net use**, **net send**, Ftp?

Can you ping the destination's IP address?

Is the destination local (same subnet) or remote (nonlocal subnet)?

Does Ping indicate that the destination is unreachable?

Does Ping return "Request Timed Out"?

This is probably not a routing problem. Test your IP stack and hardware.

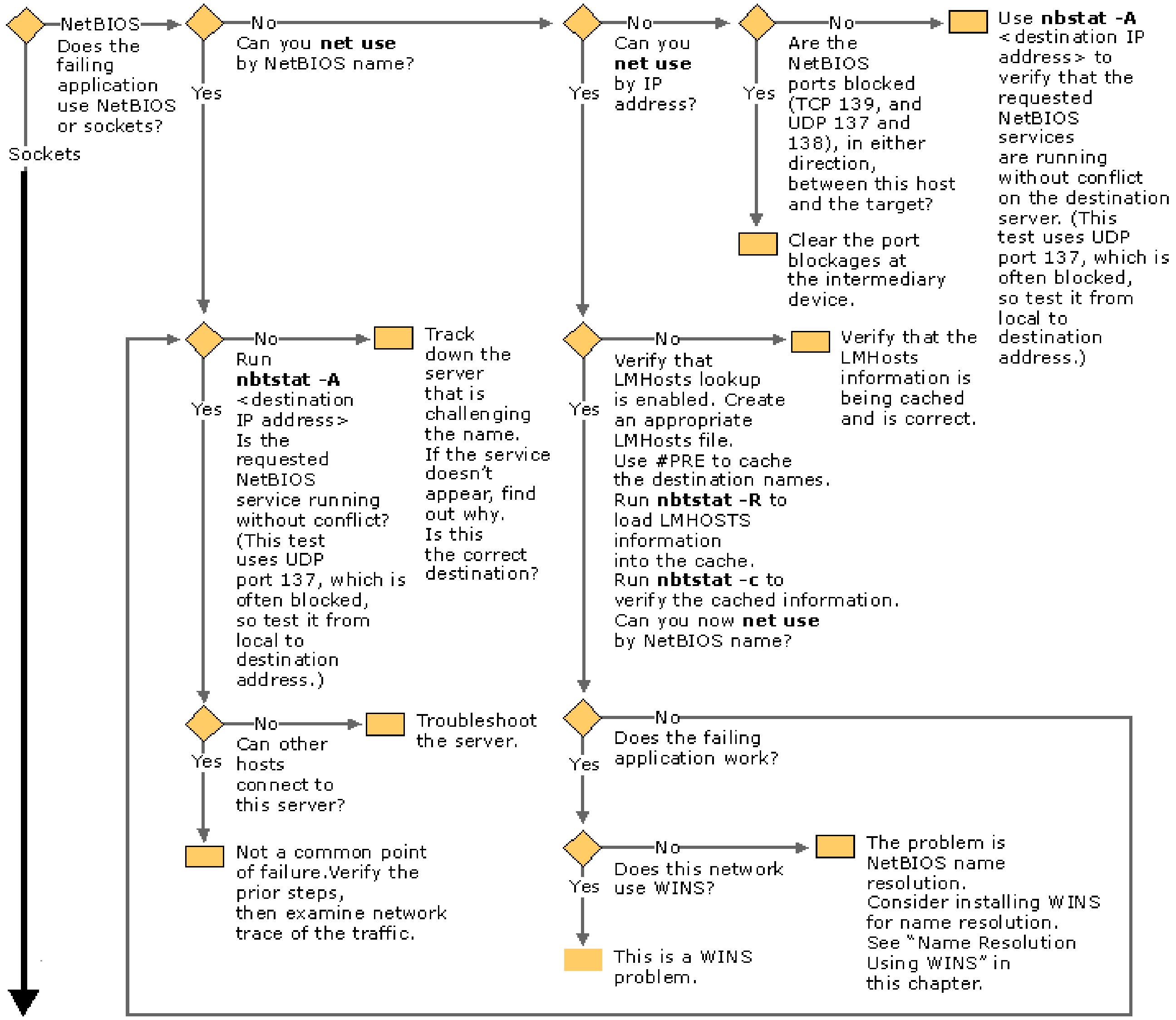
Check the host's route table: you need a default gateway or a more specific network route. Remember that gateways must be local. Can you ping the IP address of the relevant gateway?

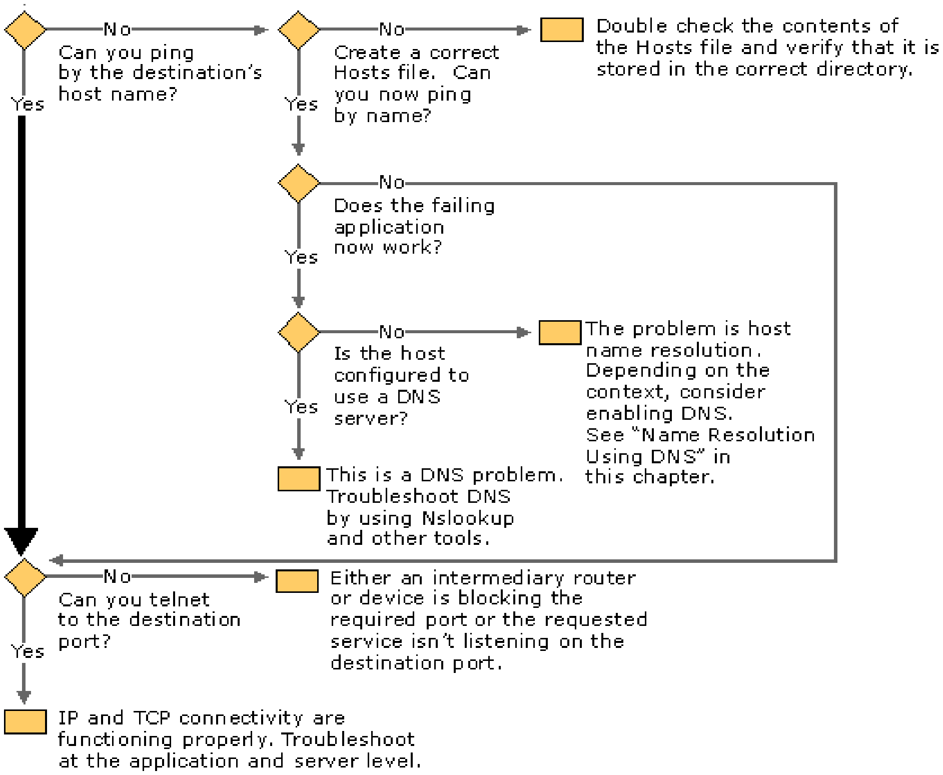
Check the destination's route table: you need a default gateway or a more specific route back to the sending host. Verify that the destination can ping its gateway. If the destination has a relevant gateway, can the host also ping it?

This gateway (or an intermediary router) needs a route back to the source.

Verify your IP configuration, check status of network adapters, cables, and hubs. Use Network Monitor to capture the traffic on the relevant segments to find out what is happening on the medium.

Verify your IP configuration, especially your subnet masks on the destination and sending host. Check status of network adapters, cables, and hubs. Use Network Monitor to capture the traffic on the relevant segments to find out what is happening on the medium.


NetBIOS
Does the failing application use NetBIOS or sockets?
Sockets
No
Yes
Can you **net use** by NetBIOS name?
Can you **net use** by IP address?
Are the NetBIOS ports blocked (TCP 139, and UDP 137 and 138), in either direction, between this host and the target?
Clear the port blockages at the intermediary device.
Use **nbstat -A** <destination IP address> to verify that the requested NetBIOS services are running without conflict on the destination server. (This test uses UDP port 137, which is often blocked, so test it from local to destination address.)
Run **nbtstat -A** <destination IP address> Is the requested NetBIOS service running without conflict? (This test uses UDP port 137, which is often blocked, so test it from local to destination address.)
Track down the server that is challenging the name. If the service doesn't appear, find out why. Is this the correct destination?
Verify that LMHosts lookup is enabled. Create an appropriate LMHosts file. Use #PRE to cache the destination names. Run **nbtstat -R** to load LMHOSTS information into the cache. Run **nbtstat -c** to verify the cached information. Can you now **net use** by NetBIOS name?
Verify that the LMHosts information is being cached and is correct.
Can other hosts connect to this server?
Troubleshoot the server.
Not a common point of failure.Verify the prior steps, then examine network trace of the traffic.
Does the failing application work?
Does this network use WINS?
This is a WINS problem.
The problem is NetBIOS name resolution. Consider installing WINS for name resolution. See "Name Resolution Using WINS" in this chapter.


No
Can you ping by the destination's host name?
Yes
No
Create a correct Hosts file. Can you now ping by name?
Yes
No
Double check the contents of the Hosts file and verify that it is stored in the correct directory.
No
Does the failing application now work?
Yes
No
Is the host configured to use a DNS server?
Yes
The problem is host name resolution. Depending on the context, consider enabling DNS. See "Name Resolution Using DNS" in this chapter.
This is a DNS problem. Troubleshoot DNS by using Nslookup and other tools.
No
Can you telnet to the destination port?
Yes
Either an intermediary router or device is blocking the required port or the requested service isn't listening on the destination port.
IP and TCP connectivity are functioning properly. Troubleshoot at the application and server level.

I can't connect to the network

1. First, make sure the problem is not a physical one. That is, check the cable connection from your network card to the hub or switch.
2. Check the LED/light at the back of your network card; the link or activity LED should be on once you've booted into your operating system, regardless if you're logged on or not.
3. Likewise, if you have access to the hub or switch, check the link LED of whichever port you're attached to, and make sure it's on.
4. If either the network card or hub LED is not on, then replace the network cable with a known good cable. Or move the cable to a known working port on the hub.
5. If you have eliminated all the physical issues, then it's time to look into the network card driver. To do so, go into Device Manager (right click on My Computer, go to Properties, select Device Manager) and look under "Network Adapters"; the model of your network card should appear here without any red "X" or yellow "!" point.
6. Once you have eliminated the network card driver problem, then it's time to check the network card protocol and configuration.
7. Right click on the Network Neighborhood icon on the desktop and select Properties, or go to Control Panel / Network. Check to see if you have the proper protocol(s) installed. Ask your network administrator to determine which protocol(s) are required for your network.

I keep getting disconnected

1. First thing to check is the network cable. If possible, replace the cable or use another cable that you know is working.
2. Check with your network administrator to see if anyone else is having this problem. If you're not the only one being disconnected, then the problem may be network related, not with the system you're using.
3. If all else fails, try replacing the network card.

My network performance is slow

1. Check with your network administrator to see if you're the only one having this problem. If other users are complaining about the network's performance, then the problem lies with the server and/or network itself, not with the system you're using.
2. Another system with a defective network card may slow the entire network down by resending incorrect packets.
3. If you're the only one having this problem, then try replacing the network card.

Check nic and device driver

1. Go into Device Manager (right click on My Computer, go to Properties, select Device Manager) and double click on Network Adapters. Make sure your NIC model is listed here and a yellow "!" or red "X" is not listed next to the device.
2. If you see a red "X" then it means the device is disabled in the hardware profile. To enable it, double click on the NIC and clear the option "Disable in this hardware profile."
3. Also, check to see if you have any entries that read "Unsupported or Other Devices." If you do, double on the entry and see what's under the list. If you see an entry that reads "PCI Ethernet Controller" then you have to remove this entry first and restart your system. The system will detect the network card and prompt you for the driver.

Check link led on nic, switch/hub and cable

1. Always check to make sure the problem you have is not a physical network problem. Which means checking the cable and the hub.
2. Look at the back of the NIC and you should see a couple of lights. Look for the LED that reads "link/active" or "activity"; the link LED should be on and the activity LED should flicker off and on.
3. If the NIC's link/active LED is off, then you need to check the cable and the hub, if you have access to it. The link/active LED on the hub should be on for whichever port you're using. If it isn't on, try plugging the cable to another open port on the hub.
4. If, after trying everything, the link LED on the NIC is still off, you may have a defective NIC.