realtimepublishers.com™

Tips and Tricks Guide™ To

Active Directory Troubleshooting

NETPRO
The Directory Experts

Don Jones

Note to Reader: This book presents tips and tricks for Active Directory troubleshooting topics. For ease of use and for cross referencing, the questions are numbered.

Copyright Statement

Q.33: How does Active Directory communicate?

A: Active Directory (AD) relies on several communications services to communicate with client computers and between domain controllers. The variety of communications protocols used reflects the complex nature both of AD and of the industry-standard protocols that AD implements, such as Kerberos and the Lightweight Directory Access Protocol (LDAP). Understanding how AD communicates can be critical when you're working with domain controllers or clients that are separated from domain controllers by firewalls or other port-filtering devices (such as routers).

Basic Communications

AD needs only a few basic services to be available for normal operations:

- User Datagram Protocol (UDP) port 88 is used for Kerberos authentication. Transmission Control Protocol (TCP) port 88 can also be used, although it's less common.
- TCP and UDP ports 135 are needed for remote procedure call (RPC) endpoint mapping. RPCs are used for a number of domain controller-to-domain controller and client-to-domain controller operations. Unfortunately, not all communications take place over port 135, as I'll discuss later.
- TCP port 139 and UDP port 138 are needed for file replication between domain controllers. This port combination is the standard NetBIOS session service port set.
- UDP port 389 handles LDAP queries and is used for normal domain controller operations.
- TCP and UDP ports 445 are used for file replication and are the standard Windows file sharing ports.
- TCP and UDP ports 464 are the Kerberos password change protocol ports.
- TCP port 593 is used by the RPC over HTTP transport. Although you don't technically need this port for normal operations, I'll discuss later how this feature can make working with domain controllers through firewalls a bit easier.
- TCP port 636 is for LDAP over Secure Sockets Layer (SSL), which is the default LDAP methodology for Windows Server 2003 and later.
- TCP port 3268 and 3269 handle Global Catalog (GC) queries. Port 3269 handles secure queries. Any domain controller that needs access to a GC or that is acting as a GC server will use these ports.
- TCP and UDP ports 53 are used to communicate with Domain Name System (DNS), which is a vital part of AD communications.

Generally, opening these ports between clients and domain controllers, or between domain controllers, will enable AD to function normally. One exception is RPC traffic.

RPC Endpoint Mapping

Most RPC communications first start on TCP port 135. However, that's merely the RPC endpoint mapper service. Its function is to select a new destination port for further communications in that RPC session. Exchange Server is a major user of RPC endpoint mapping, and it's very difficult to get Exchange traffic through a firewall as a result. The range of potential endpoint addresses used by RPC communications is huge, essentially requiring the entire firewall to be opened to allow all the possibilities. The ports selected by the endpoint mapper can range from TCP 1024 to TCP 65535.

Fortunately, you can force AD to always map endpoints to specific ports. The Internet Assigned Numbers Authority (IANA) has set aside ports 49152 to 65535 for private port assignments, so choose ports from this range and force AD to always use them. You'll then be able to open a much smaller range of ports in your firewalls.

To force port selection, modify the registry key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters. You'll need to create or modify a DWORD value named TCP/IP Port, and set it to whichever port you're going to use. However, there are some downsides to this modification:

- You'll need to modify every domain controller on your network; otherwise, they won't be able to communicate properly. You're effectively disabling endpoint mapping, so all domain controllers will have to be manually told which port to use.
- Your domain controllers will have to work a bit harder to handle the same number of connections. The servers communicate less efficiently when forced to use a single port for all communications because they can't rely on the port number to identify individual "conversations."

RPC over HTTP

Windows Server 2003 offers an exciting new communications protocol: RPC packets embedded within easily transported HTTP packets. This protocol is called RPC over HTTP, and it's handled by an RPC proxy DLL that's installed as an optional IIS 6.0 component.

Unfortunately, the computer initiating a conversation must choose to use RPC over HTTP, and Windows isn't currently designed to do so for domain communications. The only practical use for RPC over HTTP at the moment is Outlook 2003 communications with Exchange Server 2003; RPC over HTTP is invaluable there because it allows an RPC-heavy client such as Outlook to communicate through easy-to-manage HTTP ports. Hopefully, in the future, RPC over HTTP will become a more widespread means of communication.

Choosing Your Battles

If you're in a situation in which you have to have AD communications passing through a firewall, try to choose the path of least resistance. For example, domain controller-to-domain controller communications are amongst the most difficult as a result of the wide range of protocols in use and the need for constant RPC connectivity. However, client-to-domain controller communications are significantly less complicated, so placing a domain member in a perimeter network, for example, will be easier to deal with than placing a domain controller there.

If you absolutely must have a firewall between domain controllers, you'll need to restrict the ports they use. The File Replication Service (FRS) will need to be restricted, as will general communications. I explained earlier how to force an RPC port for general communications; Microsoft can help you with other types of traffic.

> 🕮 See the Microsoft articles "Restricting Active Directory Replication Traffic to a Specific Port" for information about restricting AD replication traffic and "How to Configure a Firewall for Domains and Trusts" for information about configuring firewalls to support domain and trust communications. In addition, see "How to Restrict FRS Replication Traffic to a Specific Static Port" for information about restricting FRS traffic.

Q.34: How do I troubleshoot Active Directory communications issues?

A: Active Directory (AD) is a complex product and requires a complex chain of communications for proper operation. There are really a few basic, common sequences of communication:

- Logon traffic—This category includes computers logging on to the domain in addition to user logons.
- Ticket request traffic—This type of communication occurs whenever clients need to access a new domain resource.
- Replication traffic—This type of communication occurs periodically between domain controllers and involves both intra-site and inter-site replication.

> 🕮 For more information about AD communications, see Question 10, Question 19, and Question 33.

Additional traffic can occur when clients attempt to look up cross-domain references by contacting a Global Catalog (GC) server. Clients might contact GCs for other reasons, such as during the logon process, to resolve Exchange 200x Address Book lookups, and more.

Troubleshooting this traffic can be difficult. Kerberos traffic, for example, is always encrypted using temporary encryption keys established between the Key Distribution Center (KDC—domain controller) and clients. Even simple Lightweight Directory Access Protocol (LDAP) queries to GC servers or domain controllers are generally encrypted by Secure Sockets Layer (SSL) encryption, particularly in Windows Server 2003, in which LDAP over SSL is the default.

Know the Sequence

The only way to effectively troubleshoot AD communications is to use Network Monitor. Although other utilities are very good at troubleshooting AD's operations, they don't get into low-level communications. You'll need to know what to expect, then you can see what traffic is actually being transmitted.

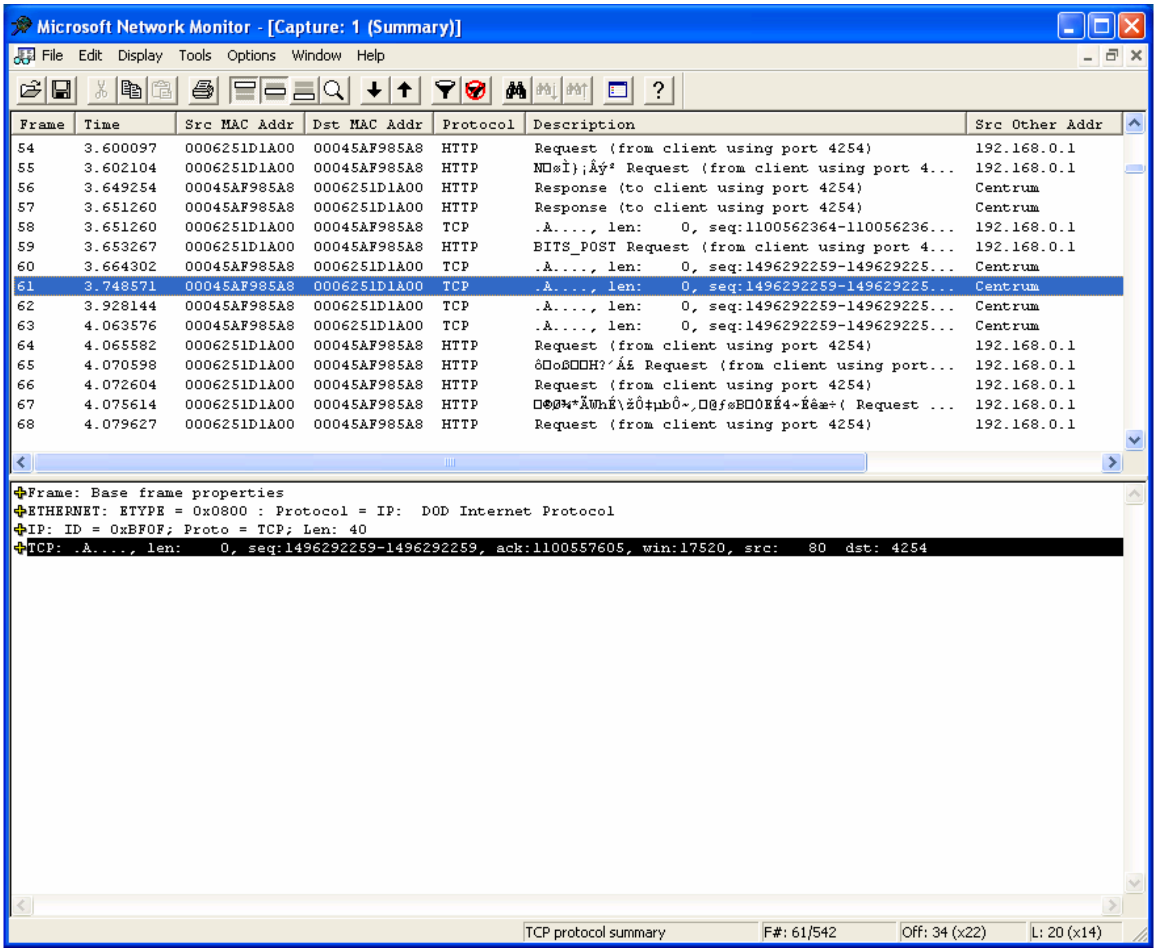
Network Monitor Versions

Keep in mind that Microsoft makes two versions of Network Monitor. The one that comes with Windows is restricted and can only capture traffic coming to and going from the local computer. If this version is the only one you have access to, you'll need to run it on the domain controller that you're troubleshooting.

The other version comes with Systems Management Server (SMS) and can capture any traffic on the local segment, even if the computer running Network Monitor isn't involved in the communication.

The one caveat you'll need to be aware of with Network Monitor is that it doesn't have built-in parsers for most of the traffic AD uses, including Kerberos. That's OK; you don't need a parser to troubleshoot traffic, you just need to know what you're doing. Network Monitor might not, for example, recognize Kerberos traffic when it sees it (because the tool lacks a parser), but I'll show you what to look for so you'll recognize the traffic.

Figure 34.1 shows a typical Network Monitor capture. In the top pane are all the capture frames of traffic; the bottom pane shows a breakdown of the frame selected in the top pane. Notice that the protocol in this case is listed as TCP. Network Monitor will show the Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) for most AD traffic. To determine the actual traffic type, you'll need to look at the packet's details.



```
Frame  Time      Src MAC Addr  Dst MAC Addr  Protocol  Description                                   Src Other Addr
54     3.600097  0006251D1A00  00045AF985A8  HTTP      Request (from client using port 4254)         192.168.0.1
55     3.602104  0006251D1A00  00045AF985A8  HTTP      N□❀Ì};Åý² Request (from client using port 4...  192.168.0.1
56     3.649254  00045AF985A8  0006251D1A00  HTTP      Response (to client using port 4254)          Centrum
57     3.651260  00045AF985A8  0006251D1A00  HTTP      Response (to client using port 4254)          Centrum
58     3.651260  0006251D1A00  00045AF985A8  TCP       .A...., len:    0, seq:1100562364-110056236...  192.168.0.1
59     3.653267  0006251D1A00  00045AF985A8  HTTP      BITS_POST Request (from client using port 4...  192.168.0.1
60     3.664302  00045AF985A8  0006251D1A00  TCP       .A...., len:    0, seq:1496292259-149629225...  Centrum
61     3.748571  00045AF985A8  0006251D1A00  TCP       .A...., len:    0, seq:1496292259-149629225...  Centrum
62     3.928144  00045AF985A8  0006251D1A00  TCP       .A...., len:    0, seq:1496292259-149629225...  Centrum
63     4.063576  00045AF985A8  0006251D1A00  TCP       .A...., len:    0, seq:1496292259-149629225...  Centrum
64     4.065582  0006251D1A00  00045AF985A8  HTTP      Request (from client using port 4254)         192.168.0.1
65     4.070598  0006251D1A00  00045AF985A8  HTTP      ô□oß□□H?´Á£ Request (from client using port...  192.168.0.1
66     4.072604  0006251D1A00  00045AF985A8  HTTP      Request (from client using port 4254)         192.168.0.1
67     4.075614  0006251D1A00  00045AF985A8  HTTP      □®Ø¾*ÃWhÉ\žÔ‡µbÔ~,□@ƒ❀B□OEÉ4~Éêæ÷( Request ...  192.168.0.1
68     4.079627  0006251D1A00  00045AF985A8  HTTP      Request (from client using port 4254)         192.168.0.1

+Frame: Base frame properties
+ETHERNET: ETYPE = 0x0800 : Protocol = IP:  DOD Internet Protocol
+IP: ID = 0xBF0F; Proto = TCP; Len: 40
+TCP: .A...., len:    0, seq:1496292259-1496292259, ack:1100557605, win:17520, src:   80  dst: 4254

TCP protocol summary    F#: 61/542    Off: 34 (x22)    L: 20 (x14)
```

Figure 34.1: Typical Network Monitor capture.

In this example, the packet has a source (src) port of 80 and a destination (dst) port of 4254. I can see that the source address was a server and the destination was a client, so I can presume that this is a reply. That means the server's source port is the same port that the client originally used to contact the server: port 80. Port 80 is HTTP traffic, although Network Monitor obviously didn't recognize this frame as a traditional HTTP packet. The destination port was randomly chosen by the client when the client initiated the communication session.

Most AD communications will appear as TCP, and you'll need to look at the source port used by the server or the destination port used by the client to determine the traffic. Expect to see the following TCP and/or UDP ports:

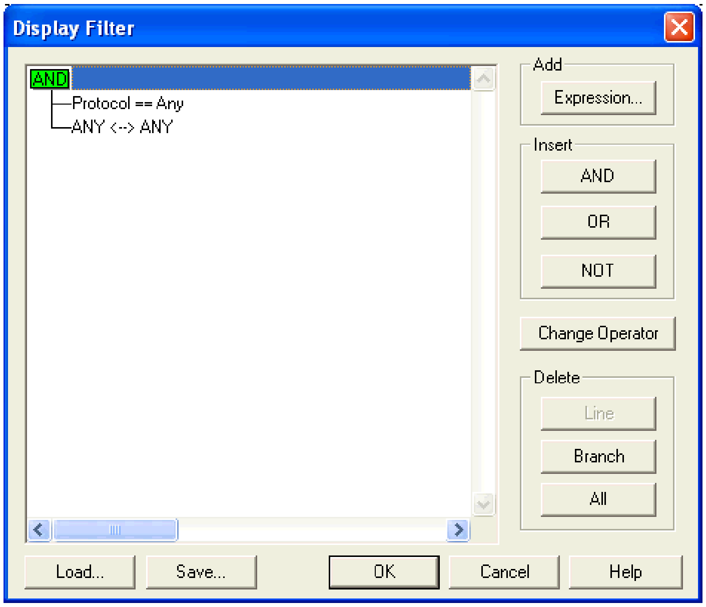
- 88—Kerberos
- 135—Remote procedure call (RPC) endpoint mapper
- 53—Domain Name System (DNS)
- 137—NetBIOS name server
- 139—NetBIOS session service
- 389—LDAP query
- 445—Server Message Blocks (SMBs)
- 636—Secure LDAP (LDAP over SSL)
- 3268—GC LDAP
- 3269—Secure GC (LDAP over SSL)

What to Expect

The exact sequence of packets will differ from environment to environment, especially because AD traffic is usually interleaved with many other, unrelated forms of communication. If you're experiencing any form of problem, check the following items (these don't require you to memorize complex sequences of communications):

- If traffic is passing through a firewall or router, check the traffic on both sides of the device to make sure it's all getting through.
- Make sure DNS queries are working. Network Monitor will parse these and display them as DNS queries; ensure that queries are receiving appropriate responses and that clients are querying the correct servers.
- You can't actually read most exchanges between domain controllers or clients, but you can ensure that there's an actual conversation. In other words, make sure you're seeing the client make a request, then see the domain controller send something back.
- Watch for patterns. For example, if some operation seems to be taking too long or repeatedly times out, you'll likely see an identical sequence of packets repeat over and over. This repetition generally means something internal to the traffic has gone wrong, such as a user password being wrong or an incorrect configuration parameter.

One way that Network Monitor can help draw out the AD traffic from everything else on your network is with filters. When viewing a capture, click Filter. You'll see a dialog box similar to the one that Figure 34.2 shows.

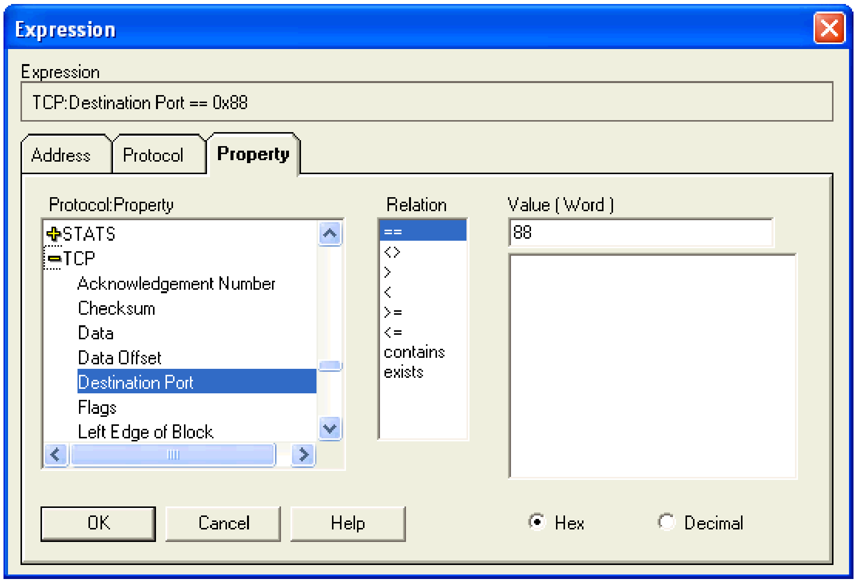


Display Filter

```
AND
├─Protocol == Any
└─ANY <--> ANY
```

Figure 34.2: Viewing filters in Network Monitor.

Click Expression to add a filter expression. As Figure 34.3 shows, create an expression that filters for packets having a specific destination port. In this example, the filter will eliminate all traffic except Kerberos traffic, which uses UDP and TCP 88.



Expression

Expression: TCP:Destination Port == 0x88

Protocol:Property: TCP — Destination Port

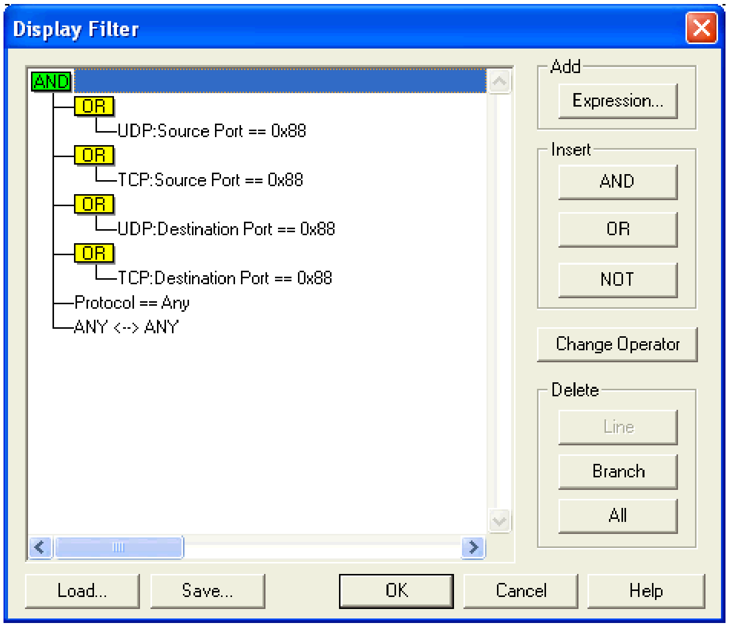
Relation: ==

Value (Word): 88

● Hex ○ Decimal

Figure 34.3: Filtering for TCP port 88.

Be sure to create filters that allow for a source or destination port so that you'll capture both the client and server sides of the exchange. You can use the Or button on the filter dialog box to ensure that any packets meeting any one of your criteria will be included in the display. Figure 34.4 shows an example of the configured filters.



```
AND
 ├OR
 │ └UDP:Source Port == 0x88
 ├OR
 │ └TCP:Source Port == 0x88
 ├OR
 │ └UDP:Destination Port == 0x88
 ├OR
 │ └TCP:Destination Port == 0x88
 ├Protocol == Any
 └ANY <--> ANY
```

Figure 34.4: Configured display filters.

Filtering for specific types of traffic will help you focus on different areas to be sure they're working properly without having to manually wade through all of the packets that Network Monitor may have captured. You can load and save filters for later use, making it easier to quickly look at Kerberos traffic, DNS traffic, LDAP queries, and so forth.

No Fixes, Just Symptoms

Network Monitor won't allow you to directly fix any problems. However, it can point the way to what's broken. For example, if capturing packets from each side of a firewall reveals that Kerberos (port 88) packets aren't making it through, you'll know that you need to reconfigure the firewall. If DNS queries contain incorrect IP addresses, you'll need to fix the DNS zone database. If traffic seems to be repeating itself, there's something wrong or misconfigured within the traffic itself (such as a username or password contained within the traffic), and you'll need to closely examine the configuration of the computers involved.

Q.35: DNS works sometimes, but sometimes it doesn't. What can we do?

A: Troubleshooting any kind of intermittent network problem can be a nightmare. Fortunately, Domain Name System (DNS) is a fairly simple protocol, and there are only so many things that can go wrong.

Low-Hanging Fruit

Start by eliminating obvious problem areas, such as unavailable DNS servers, WAN links that are down, unplugged cables, failed network cards, and so on. These types of problems will almost always manifest in other ways, because DNS won't usually be the only thing affected. Because your client will more often than not be configured with multiple DNS server addresses, use Network Monitor to analyze the network traffic being sent by your clients. That way, you'll know exactly which DNS server they're trying to talk with, and you can focus your troubleshooting efforts there first.

Another common problem root is multihomed servers. Unless specifically instructed to do otherwise, these servers will register all of their IP addresses with DNS. Some of those addresses, however, may be associated with network interfaces that not all clients can access. The result is that some clients will have access to the server and others won't. You may also have clients that switch between having access and not, particularly if DNS round robin is enabled on their DNS server. Round robin may be alternating between an accessible IP address and an inaccessible one, creating intermittent problems for clients.

Replication Issues

Replication issues can cause intermittent problems in Active Directory (AD)-integrated DNS zones. Ensure that AD replication is working properly to start with. Clients that are querying different DNS servers may be receiving different responses if the two servers haven't yet converged.

📖 For more information about how AD replication works, see Question 19.

If replication latency is a problem for your DNS zones, consider upgrading to Windows Server 2003. In Windows Server 2003, the DNS zone is stored in an AD partition, and you can control which domain controllers contain a copy of the partition. By limiting the partition to just those domain controllers that are acting as DNS servers, you'll force a new replication topology to be generated for that partition. The result will be fewer servers replicating the information. Thus, replication will be able to occur more quickly, causing the different copies of the partition to converge more quickly and reducing problems caused by replication latency.

Protocol Problems

Another problem can occur if your network is assuming that DNS uses User Datagram Protocol (UDP) port 53 and blocks access to Transmission Control Protocol (TCP) port 53. The DNS specification requires DNS to use the connectionless UDP transport protocol, but only for small queries. Larger queries—or, more accurately, larger query *responses*—that won't fit into a single UDP packet may be broken into multiple TCP packets instead. This switch to TCP can cause bewildering problems on your network because some DNS queries will work fine and others will simply time out.

> ✎ DNS *queries* will nearly always go out via UDP (the notable exception being the Simple Mail Transfer Protocol—SMTP—service in IIS, which seems to always use TCP); *replies* will come in on UDP or TCP depending upon the number of hosts and IP addresses contained within the replies.

If you're not sure whether this circumstance relates to your problems, try using Network Monitor to capture DNS traffic on both sides of your firewall. If you're not seeing identical traffic on both sides of the firewall, the firewall is obviously blocking some DNS traffic, most likely large replies. To play it safe, I recommend opening your network to incoming DNS traffic on both UDP and TCP ports 53.

Q.36: How can we troubleshoot Group Policy application?

A: Group Policy application seems straightforward enough: Group Policy Objects (GPOs) are linked to organizational units (OUs); users and computers are in OUs. All the GPOs from a user's OU hierarchy filter down to the user. Easy enough.

Things get more complicated, though, when you remember that GPOs can be linked to a domain and to sites—meaning you'll have to open a whole new console to see what's going on. You also have to consider local security policies, which exist solely on the client computer and are applied before any domain-based policies arrive. Throw in options such as Block Policy Inheritance, No Override, and loopback processing, and it's no wonder why there's such a robust market for third-party GPO tools. However, with some patience and a methodology, you can do quite a bit of quality troubleshooting on your own.

Start at the Bottom

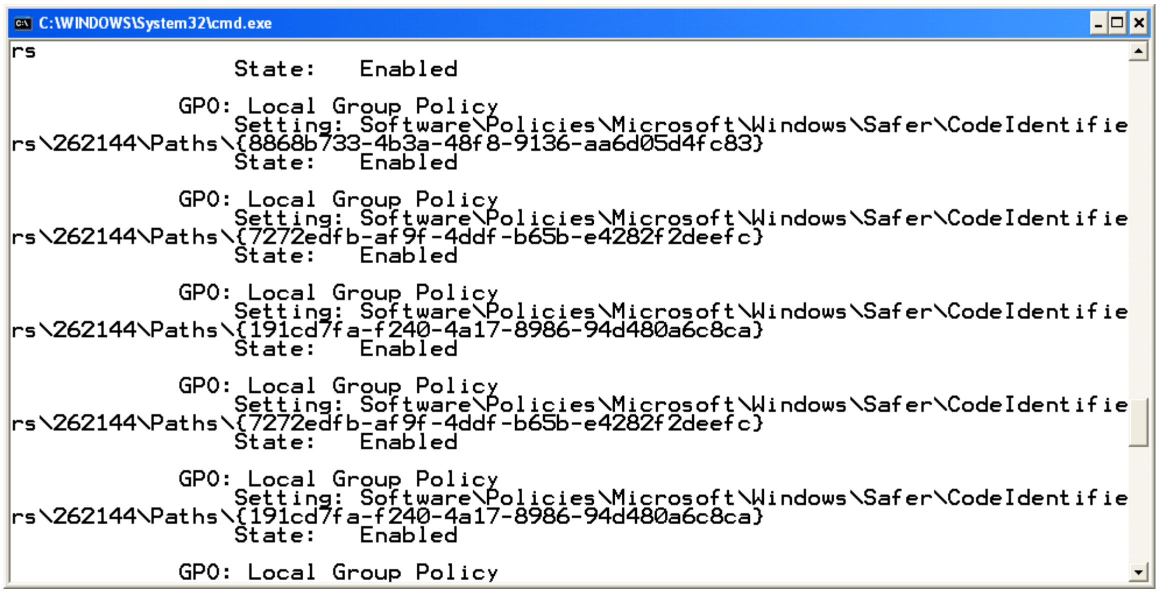
Too many administrators try to start at the top, working their way down the hierarchy of GPOs and figuring out which ones apply. That method is time-consuming, error-prone, and just plain boring. It's a lot easier to start at the bottom—the client—and work your way *up* the tree.

Windows XP's Gpresult tool, for example, is a great troubleshooting tool. Run from the command line, it will tell you which groups the current user is a member of (which can affect GPO application), and give you a list of every GPO that is currently affecting the user. You'll also see the last time that GPOs were applied to the computer. What Gpresult is displaying is called resultant set of policy (RSOP). It sorts through all the blocked inheritance, no overrides, and conflicting policies to sort out exactly which policies are being applied.

By default, Gpresult doesn't show you which individual *policies* are applied or what they are set to; because GPOs successively overwrite one another as they are applied, you can still be left with a troubleshooting task to figure out which of the GPOs listed is responsible for the settings you're seeing. Fortunately, Gpresult has a "superverbose" mode, enabled by running

```
Gpresult /z
```

This mode not only displays which GPOs have been applied, but lists every single policy that's enabled in each GPO, allowing you to see which GPO modified which setting, and which GPO finally won out in the end. Figure 36.1 shows a portion of Gpresult's superverbose output. In this example, the GPO being applied is Local Group Policy, and you can see exactly which registry keys each setting is modifying.



```
rs
                State:   Enabled

            GPO: Local Group Policy
                Setting: Software\Policies\Microsoft\Windows\Safer\CodeIdentifie
rs\262144\Paths\{8868b733-4b3a-48f8-9136-aa6d05d4fc83}
                State:   Enabled

            GPO: Local Group Policy
                Setting: Software\Policies\Microsoft\Windows\Safer\CodeIdentifie
rs\262144\Paths\{7272edfb-af9f-4ddf-b65b-e4282f2deefc}
                State:   Enabled

            GPO: Local Group Policy
                Setting: Software\Policies\Microsoft\Windows\Safer\CodeIdentifie
rs\262144\Paths\{191cd7fa-f240-4a17-8986-94d480a6c8ca}
                State:   Enabled

            GPO: Local Group Policy
                Setting: Software\Policies\Microsoft\Windows\Safer\CodeIdentifie
rs\262144\Paths\{7272edfb-af9f-4ddf-b65b-e4282f2deefc}
                State:   Enabled

            GPO: Local Group Policy
                Setting: Software\Policies\Microsoft\Windows\Safer\CodeIdentifie
rs\262144\Paths\{191cd7fa-f240-4a17-8986-94d480a6c8ca}
                State:   Enabled

            GPO: Local Group Policy
```

Figure 36.1: Gpresult's superverbose mode.

Superverbose mode also breaks down the user and computer policies, allowing you to see every setting that is affecting the current users or their machines.

Centralized Troubleshooting

For Windows Server 2003, Microsoft introduced a sort of server-side Gpresult. It's called RSOP, and it's built into the Active Directory Users and Computers console. With this tool, you can actually view applied policies in a graphical user interface, which can be a bit easier to work with than the text-only Gpresult output.

> ✎ The Windows Server 2003 version of the Active Directory Users and Computers console will work fine against Windows Server 2000 domains running Service Pack 3 (SP3) and later; however, you might need to purchase a copy of Windows Server 2003 to be able to legally use the new the new Active Directory Users and Computers console.

To launch the new tool, open the Active Directory Users and Computers console and select a user or computer. Right-click the object, and select Resultant Set of Policy (Planning) from the All Tasks menu. The RSOP wizard will step through a number of screens that allow you to specify the user or computer account location, determine whether you want to simulate the effect of a slow network link or loopback processing, choose a specific site, or modify either the user's or computer's security group memberships. The final result, which Figure 36.2 shows, is a console that looks a lot like the Group Policy Object Editor, displaying each of the policies that have been applied to the user. For each policy, you'll see which GPO was responsible for the final application of the policy, making it easy to see where you need to go to make changes.

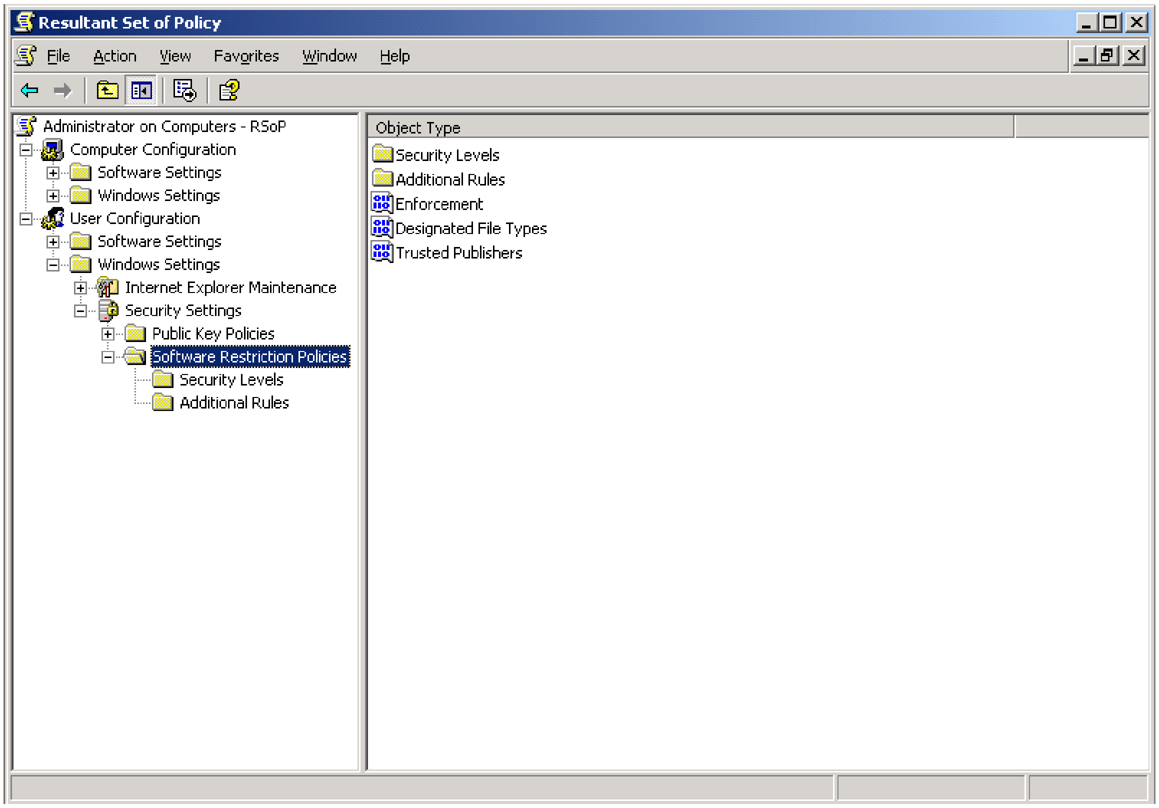


Figure 36.2: Output from the new Active Directory Users and Computers RSOP tool.

Remember the Pecking Order

Remembering the pecking order of GPOs can be helpful when troubleshooting. In general, the *least specific* policies apply first; those which are *more specific* to a user or computer apply last and have the opportunity to overwrite policies applied earlier. That's an important concept: The *last* policy applied remains effective, even if an *earlier* policy had a contradictory setting.

Local policies apply first; they have the advantage of living right on the computer, so they don't have to wait for the user to log on. Next comes the Active Directory (AD)-based policies, and of those, site policies apply first. Domain policies, which are more specific, apply next. Finally, OU policies apply starting at the topmost OU and working down. OUs can block policy inheritance, meaning higher-up policies will not apply from that point on down. However, higher-up policies can also be marked as No Override, meaning they'll break through Block Policy Inheritance and be inherited anyway.

Replication Problems Equals Inconsistency

If users are experiencing inconsistent GPO application, the problem is most likely a failure in Active Directory's (AD's) GPO replication process. Although AD *defines* GPO links in the AD database, the GPOs themselves are contained in normal files, which are replicated from domain controller to domain controller by the File Replication Service (FRS). A failure in the FRS can result in inconsistent GPOs on domain controllers, which results in users having inconsistent GPO application.

> 🕮 For more information about the FRS, see Question 31.

Perhaps the simplest way to verify that the GPOs have replicated consistently is to check the files themselves, located in each domain controller's SYSVOL share. Provided each domain controller has the same files, with the same date and time stamps (which are replicated by FRS, not recreated on each domain controller), then everything should be consistently applied to all users.

Q.37: Are WAN links are being over-utilized by Active Directory replication traffic. If yes, why?

A: Most companies do the right thing when it comes to Active Directory (AD) site design. For example, suppose you have several sites connected by T1 lines, as Figure 37.1 shows. The T1 lines represent your WAN's physical connectivity, and all the AD design guidelines tell you that your site links should reflect that physical connectivity.

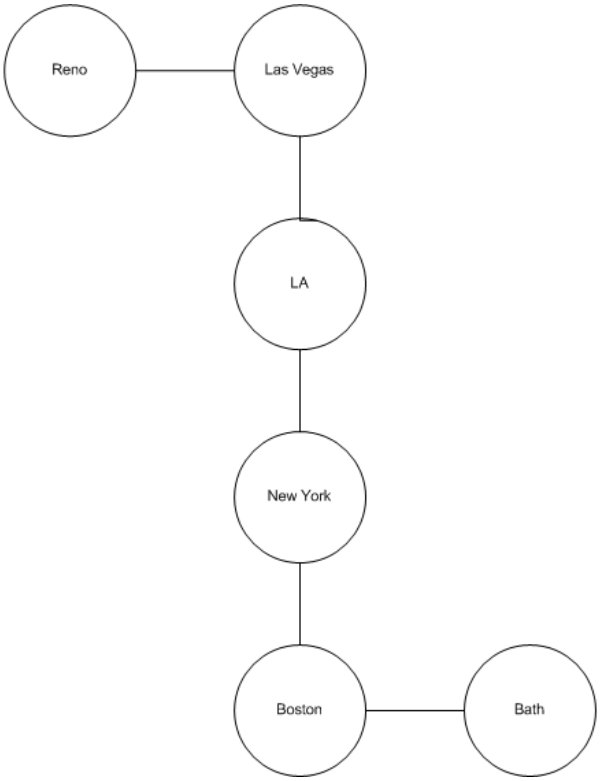


Figure 37.1: Sample physical WAN design.

If you set up one site link per T1 link, you'll wind up with a design somewhat like the one in Figure 37.2. This configuration is straightforward and reflects the way that most companies build their sites and networks. If you have additional backup links between sites, you might even configure separate site links for those, configuring a higher link cost to reflect the link's nature as a backup.

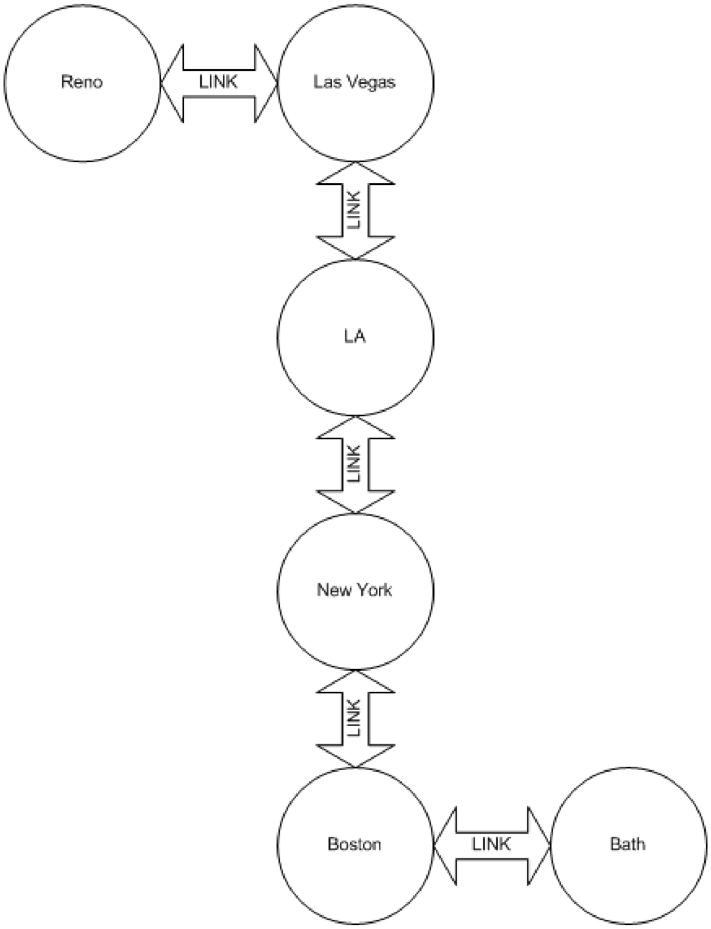


Figure 37.2: Site link topology.

Secret Bridges

What you might not realize is that AD, by default, creates *site link bridges* for every site link. This setup isn't necessarily a bad idea. For example, consider what happens if an administrator locks out a user account in the Bath office. Obeying only the site links, AD will have to replicate that change from a bridgehead in the Bath office to a bridgehead in the Boston office, then to New York, LA, Las Vegas, and finally Reno. Depending upon your replication schedules, it could be quite some time before a Reno domain controller actually locked out the user's account, even though account lockout is a high-priority change for replication. In the meantime, a user could be logging on to a Reno or Las Vegas domain controller, relying on the fact that those domain controllers haven't yet heard about the lockout.

AD's automatic site link bridges create a topology similar to the one in Figure 37.3, in which each site is *logically* linked to the others.

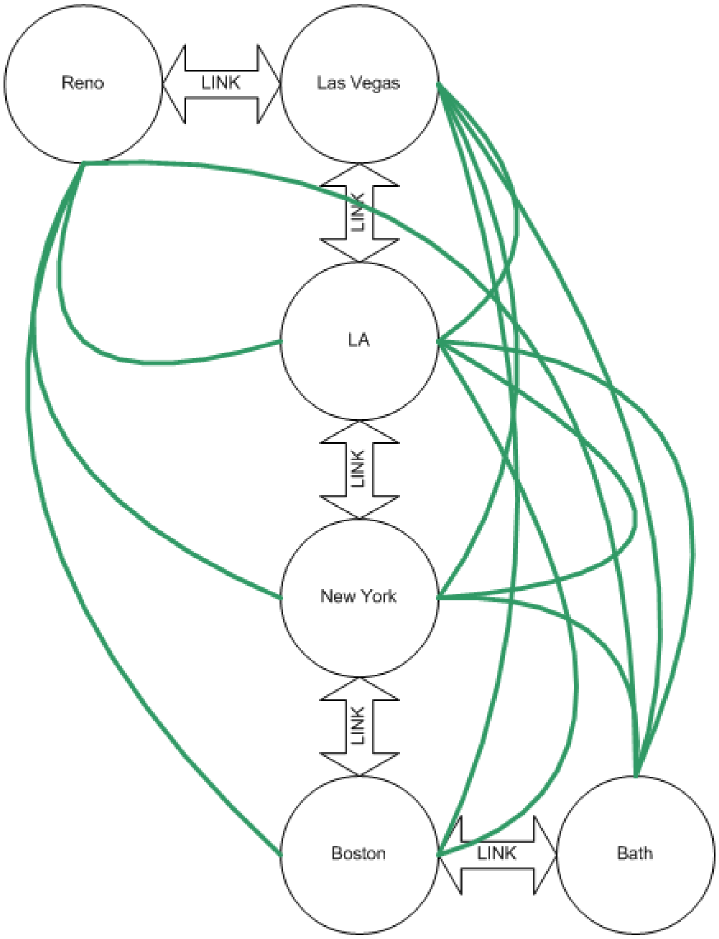


Figure 37.3: Site link bridges are shown in green.

When a change is made at the Bath office, its bridgehead domain controllers replicate *directly* to bridgehead domain controllers in each of the other offices. Effectively, AD is ignoring the physical layout of your network a bit in order to speed replication. The cost is that your WAN links are going to carry more traffic than you might expect. For example, the link connecting Las Vegas and LA will carry Bath's change *twice*—once as Bath replicates with Las Vegas, and once as Bath replicates with Reno. The link between Bath and Boston will carry the same replication traffic *five* times—once for each of the other offices.

> 🕮 For more information about how site link bridges are created and how the replication topology is generated, see Question 19.

Being Smarter than AD

You don’t have to let AD create site link bridges automatically. In fact, you can disable the behavior entirely. However, doing so puts you right back to a high degree of replication latency, which may not be any more desirable than wasting WAN bandwidth. Fortunately, there’s a happy middle ground. Consider the topology that Figure 37.4 shows, in which two site link bridges have been manually created.

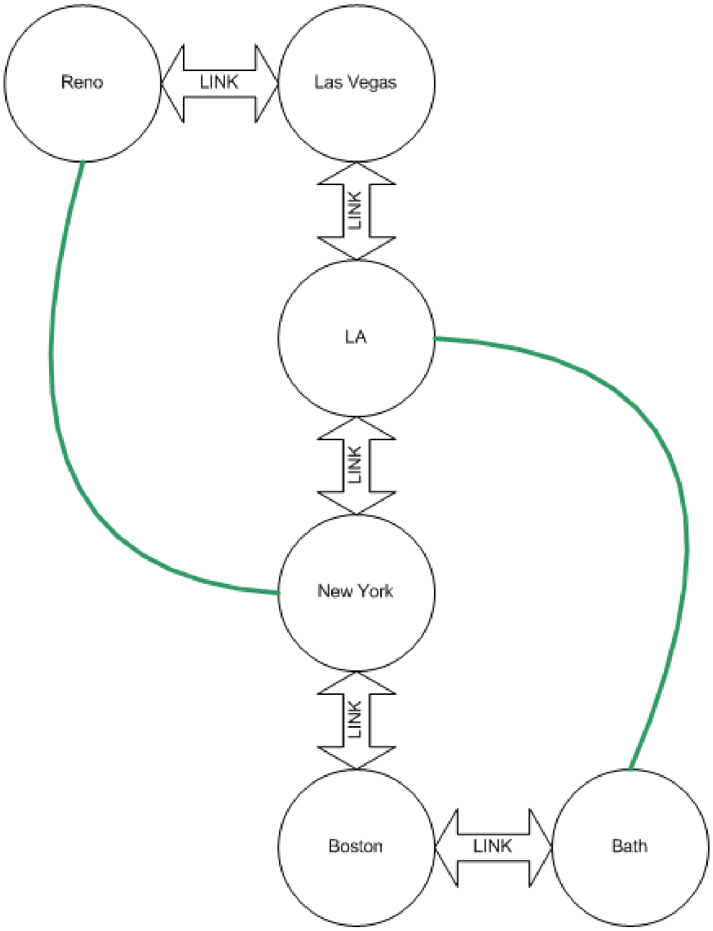
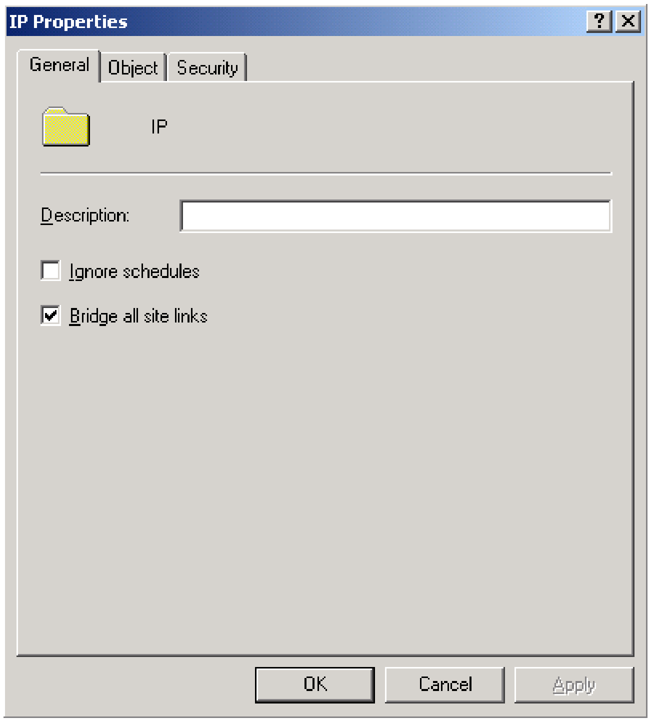


Figure 37.4: Manually created site link bridges.

In this case, a change made at Bath would replicate to Boston and LA, because LA is “virtually” connected to Bath. LA would then replicate to New York and Las Vegas. Reno would receive the information last, either from New York or Las Vegas. You might reconfigure this setup a bit to have the Reno site link bridge connecting to LA rather than New York; doing so would place more burdens on LA-based domain controllers, but would disseminate the information in fewer steps. The site link bridges effectively shortcut the physical topology, wasting a small amount of WAN bandwidth but providing minimal replication latency.

Making the Change

You can make this configuration change in the AD Sites & Services console. You'll need to make the change for each intersite replication transport in use, although most companies will just have IP. Right-click the transport's folder, and select Properties from the context menu. As Figure 37.5 shows, the default behavior is to bridge all site links; you can disable this behavior by clearing the check box.



IP

Description:

☐ Ignore schedules

☑ Bridge all site links

Figure 37.5: Default settings of the IP intersite transport.

If you disable this behavior, you should definitely review your manual site link bridges and create new ones as necessary to provide the desired amount of replication latency. I recommend testing your new topology by making changes in your furthest-out office (Bath, in our example) and measuring the amount of time it takes the change to replicate to the opposite corner of your network (Reno, in our example). Adding site link bridges that bridge from the edges of your WAN to the middle of your WAN is the most effective strategy, as it provides the most efficient shortcut for replication traffic.