Windows 2000 Server

Chapter 7 - Remote Access Server

Microsoft® Windows® 2000 has extensive support for remote access technology to connect remote clients to corporate networks or the Internet. This chapter describes how remote access works and details how to troubleshoot remote access problems.

This chapter is intended for network engineers and support professionals who are already familiar with TCP/IP, IP routing, IPX routing, and wide area network technology, and assumes that you have read the section about remote access in Windows 2000 Server Help.

In This Chapter

Related Information in the Resource Kit

- For more information about TCP/IP routing, see "Introduction to TCP/IP" in the *Microsoft® Windows® 2000 Server Resource Kit TCP/IP Core Networking Guide*.
- For more information about unicast IP routing, see "Unicast IP Routing" in this book.
- For more information about IPX routing, see "IPX Routing" in this book.
- For more information about demand-dial routing, see "Demand-Dial Routing" in this book.
- For more information about virtual private networking, see "Virtual Private Networking" in this book.

Note This chapter mentions Windows 2000 registry entries. For more information about these registry entries, see the *Technical Reference to the Windows 2000 Registry* on the Windows 2000 Resource Kit CD-ROM.

Remote Access Overview

With Windows 2000 remote access, remote access clients connect to remote access servers and are transparently connected to the remote access server, known as point-to-point remote access connectivity, or transparently connected to the network to which the remote access server is attached, known as point-to-LAN remote access connectivity. This transparent connection allows remote access clients to dial-in from remote locations and access resources as if they were physically attached to the network.

Windows 2000 remote access provides two different types of remote access connectivity:

1. Dial-up remote access

   With dial-up remote access, a remote access client uses the telecommunications infrastructure to create a temporary physical circuit or a virtual circuit to a port on a remote access server. Once the physical or virtual circuit is created, the rest of the connection parameters can be negotiated.

2. Virtual private network (VPN) remote access

   With virtual private network remote access, a VPN client uses an IP internetwork to create a virtual point-to-point connection with a remote access server acting as the VPN server. Once the virtual point-to-point connection is created, the rest of the connection parameters can be negotiated.

Note This chapter is primarily devoted to the discussion of dial-up remote access; however, many topics also apply to VPN remote access. For a complete understanding of VPNs, read this chapter first and then read the chapter "Virtual Private Networking" in this book.

Remote Access Versus Remote Control

The distinctions between remote access and remote control solutions are the following:

- The remote access server is a software-based multi-protocol router; remote control solutions work by sharing screen, keyboard, and mouse over the remote link. In remote access, the applications are run on the remote access client computer.
- In a remote control solution, users share a CPU or multiple CPUs on the server. In remote control, the applications are run on the server. The remote access server's CPU is dedicated to facilitating communications between remote access clients and network resources, not to running applications.

Elements of a Dial-Up Remote Access Connection

A dial-up remote access connection consists of a remote access client, a remote access server and a wide area network (WAN) infrastructure as illustrated in Figure 7.1.

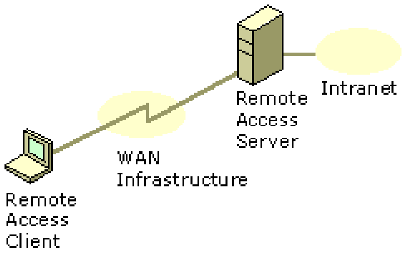


Figure 7.1 Elements of a Dial-Up Remote Access Connection

Remote Access Client

Windows 2000, Microsoft® Windows NT® 3.5 or later, Microsoft® Windows® 95, Microsoft® Windows® 98, Microsoft® Windows® for Workgroups, Microsoft® MS-DOS®, and Microsoft® LAN Manager remote access clients can all connect to a Windows 2000 remote access server. Almost any third-party Point-to-Point Protocol (PPP) remote access clients including UNIX and Apple Macintosh can also connect to a Windows 2000 remote access server.

Remote Access Server

The Windows 2000 remote access server accepts dial-up connections and forwards packets between remote access clients and the network to which the remote access server is attached.

Note The term remote access server as it is used in this chapter refers to a Windows 2000 Server computer running the Routing and Remote Access service and configured to provide remote access.

Dial-Up Equipment and WAN Infrastructure

The physical or logical connection between the remote access server and the remote access client is facilitated by dial-up equipment installed at the remote access client, the remote access server, and the telecommunications infrastructure. The nature of the dial-up equipment and telecommunications infrastructure varies depending on the type of connection being made.

PSTN

The Public Switched Telephone Network (PSTN), also known as Plain Old Telephone Service (POTS), is the analog phone system designed to carry the minimal frequencies to distinguish human voices. Because the PSTN was not designed for data transmissions, there are limits to the maximum bit rate of a PSTN connection. Dial-up equipment consists of an analog modem for the remote access client and the remote access server. For large organizations, the remote access server is attached to a modem bank containing up to hundreds of modems. With analog modems at both the remote access server and the remote access client, the maximum bit rate supported by PSTN connections is 33,600 bits per second, or 33.6 kilobits per second (Kbps).

Figure 7.2 illustrates a PSTN connection.

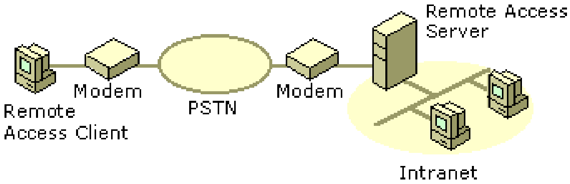


Figure 7.2 Dial-Up Equipment and WAN Infrastructure for PSTN Connections

Digital Links and V.90

The maximum bit rate of the PSTN is a function of the range of frequencies being passed by PSTN switches and the signal-to-noise ratio of the connection. The modern-day analog phone system is only analog on the local loop, the set of wires that connects the customer to the central office (CO) PSTN switch. Once the analog signal reaches the PSTN switch, it is converted to a digital signal. The analog-to-digital conversion introduces noise on the connection known as quantization noise.

When a remote access server is connected to a CO using a digital switch based on T-Carrier or ISDN rather than an analog PSTN switch, there is no analog-to-digital conversion when the remote access server sends information to the remote access client. There is no quantization noise in the downstream path to the remote access client, and therefore, there is a higher signal-to-noise ratio and a higher maximum bit rate.

With this new technology, called V.90, remote access clients can send data at 33.6 Kbps and receive data at 56 Kbps. In North America, the maximum receive bit rate is 53 Kbps due to Federal Communications Commission (FCC) power rules.

To obtain V.90 speeds, the following must be true:

- The remote access client must call using a V.90 modem.
- The remote access server must be using a V.90 digital switch and be connected to the PSTN using a digital link, such as T-Carrier or ISDN.
- There cannot be any analog-to-digital conversions in the path from the remote access server to the remote access client.

Figure 7.3 illustrates a V.90-based PSTN connection.

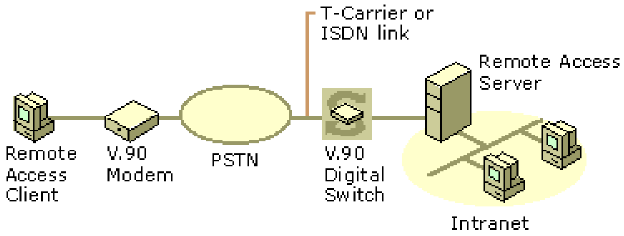


Figure 7.3 Dial-Up Equipment and WAN Infrastructure for V.90 Connections

ISDN

The Integrated Services Digital Network (ISDN) is a set of international specifications for a digital replacement of the PSTN providing a single digital network to handle voice, data, fax, and other services over existing local loop wiring. ISDN behaves like an analog phone line except that it is a digital technology at higher data rates with a much lower connection time. ISDN offers multiple channels; each channel operates at 64 Kbps and because the network is digital end-to-end, there are no analog to digital conversions.

Dial-up equipment consists of an ISDN adapter for the remote access client and the remote access server. Remote access clients typically use Basic Rate ISDN (BRI) with two 64-Kbps channels, and large organizations typically use Primary Rate ISDN (PRI) with 23 64-Kbps channels.

Figure 7.4 illustrates an ISDN connection.

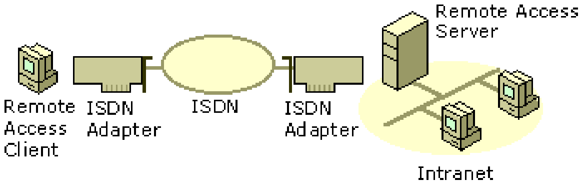


Figure 7.4 Dial-Up Equipment and WAN Infrastructure for ISDN Connections

X.25

X.25 is an international standard for sending data across public packet switching networks. Windows 2000 remote access supports X.25 in two ways:

1. The remote access client supports the use of X.25 smart cards, which can connect directly to the X.25 data network and use the X.25 protocol to establish connections and send and receive data. The remote access client also supports dialing into a packet assembler/disassembler (PAD) of an X.25 carrier using an analog modem.
2. The remote access server only supports the use of X.25 smart cards.

For more information about the configuration of X.25 and PADs, see Windows 2000 Server Help.

Note X.25 smart cards are adapters that use the X.25 protocol and can directly connect to an X.25 public data network. X.25 smart cards are not related to smart cards used for authentication and secure communications.

Figure 7.5 illustrates an X.25 connection.

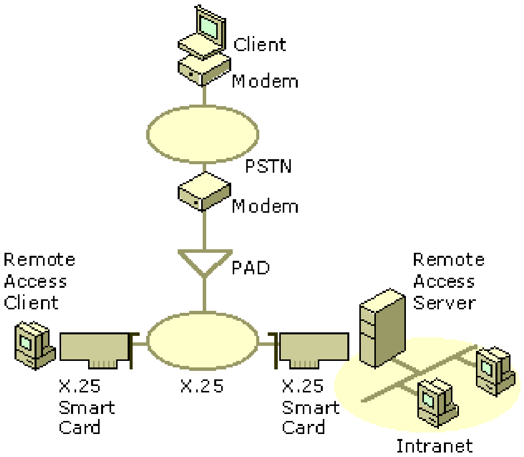


Figure 7.5 Dial-Up Equipment and WAN Infrastructure for X.25 Connections

ATM over ADSL

Asymmetric Digital Subscriber Line (ADSL) is a new local loop technology for small business and residential customers. Although ADSL provides higher bit rates than PSTN and ISDN connections, the bit rate is not the same in the upstream and downstream directions. Typical ADSL connections offer 64 Kbps from the customer and 1.544 megabits per second (Mbps) to the customer. The asymmetric nature of the connection fits well with typical Internet use. Most Internet users receive a lot more information than they send.

ADSL equipment can appear to Windows 2000 as either an Ethernet interface or a dial-up interface. When an ADSL adapter appears as an Ethernet interface, the ADSL connection operates in the same way as an Ethernet connection to the Internet.

When an ADSL adapter appears as a dial-up interface, ADSL provides a physical connection and the individual LAN protocol packets are sent using Asynchronous Transfer Mode (ATM). An ATM adapter with an ADSL port is installed in both the remote access client and remote access server.

Figure 7.6 illustrates an ATM over ADSL connection.

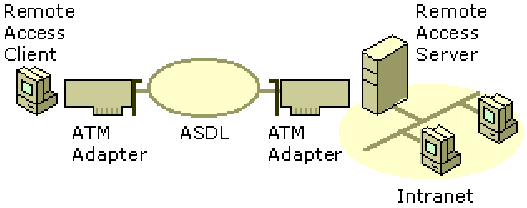


Figure 7.6 Dial-Up Equipment and WAN Infrastructure for ATM over ADSL Connections

Remote Access Protocols

Remote access protocols control the connection establishment and transmission of data over wide area network (WAN) links. The operating system and LAN protocols used on remote access clients and servers dictate which remote access protocol your clients can use.

There are three types of remote access protocols supported by Windows 2000 remote access:

1. Point-to-Point Protocol (PPP) is an industry-standard set of protocols providing the best security, multi-protocol support, and interoperability.
2. Serial Line Internet Protocol (SLIP) is used by older remote access servers.
3. Microsoft RAS protocol, also known as Asynchronous NetBEUI or AsyBEUI, is a remote access protocol used by legacy remote access clients running Microsoft operating systems, such as Microsoft® Windows NT® 3.1, Windows for Workgroups, MS-DOS, and LAN Manager.

Table 7.1 summarizes the remote access protocols and their use in Windows 2000.

Table 7.1 Remote Access Protocols and Their Use in Windows 2000

Remote Access Protocols	**Remote Access Client**	**Remote Access Server**
PPP	X	X
SLIP	X	
AsyBEUI	X	X

LAN Protocols

LAN protocols are the protocols used by the remote access client to access resources on the network connected to the remote access server. Windows 2000 remote access supports TCP/IP, IPX, AppleTalk, and NetBEUI. For more information, see "Remote Access and TCP/IP and IPX" later in this chapter.

Elements of Secure Remote Access

Because remote access is designed to transparently connect a remote access client to a network and its potentially sensitive data, security of remote access connections is an important consideration. Windows 2000 remote access offers a wide range of security features including secure user authentication, mutual authentication, data encryption, callback, and caller ID.

Secure User Authentication

Secure user authentication is obtained through the encrypted exchange of user credentials. This is possible with the PPP remote access protocol using either the Extensible Authentication Protocol (EAP), Microsoft Challenge Handshake Authentication Protocol (MS-CHAP) version 1 and version 2, Challenge Handshake Authentication Protocol (CHAP), or Shiva Password Authentication Protocol (SPAP) authentication protocols. The remote access server can be configured to require a secure authentication method. If the remote access client cannot perform the required secure authentication, the connection is denied.

Mutual Authentication

Mutual authentication is obtained by authenticating both ends of the connection through the encrypted exchange of user credentials. This is possible with the PPP remote access protocol using either the EAP-Transport Level Security (EAP-TLS) or MS-CHAP version 2 authentication protocols. During mutual authentication, the remote access client authenticates itself to the remote access server, and then the remote access server authenticates itself to the remote access client.

It is possible for a remote access server to not request authentication from the remote access client. However, in the case of a Windows 2000 remote access client configured for only MS-CHAP version 2 or only EAP-TLS, the remote access client will force the mutual authentication of the client and server. If the remote access server does not respond to the authentication request, the connection is terminated by the client.

Data Encryption

Data encryption encrypts the data sent between the remote access client and the remote access server. Remote access data encryption only provides data encryption on the communications link between the remote access client and the remote access server. If end-to-end encryption is needed, use IPSec to create an encrypted end-to-end connection after the remote access connection has been made.

Note IPSec can also be used for encrypting a Layer Two Tunneling Protocol (L2TP) virtual private network connection. For more information, see "Virtual Private Networking" in this book.

Data encryption on a remote access connection is based on a secret encryption key known to the remote access server and remote access client. This shared secret key is generated during the user authentication process.

Data encryption is possible over dial-up remote access links when using the PPP remote access protocol and the EAP-TLS or MS-CHAP authentication protocols. The remote access server can be configured to require data encryption. If the remote access client cannot perform the required encryption, the connection attempt is rejected.

Windows 2000, Microsoft® Windows NT® 4.0, Windows 98, and Windows 95 remote access clients and remote access servers support the Microsoft Point-to-Point Encryption Protocol (MPPE). MPPE uses the Rivest-Shamir-Adleman (RSA) RC4 stream cipher and either 40-bit, 56-bit, or 128-bit secret keys. MPPE keys are generated from the MS-CHAP and EAP-TLS user authentication process.

Callback

With callback, the remote access server calls the remote access client after the user credentials have been verified. Callback can be configured on the server to call the remote access client back at a number specified by the user of the remote access client during the time of the call. This allows a traveling user to dial-in and have the remote access server call them back at their current location, saving phone charges. Callback can also be configured to always call the remote access client back at a specific location, which is the secure form of callback.

Caller-ID

Caller-ID can be used to verify that the incoming call is coming from a specified phone number. Caller-ID is configured as part of the dial-in properties of the user account. If the Caller-ID number of the incoming connection for that user does not match the configured Caller-ID, the connection is denied.

Caller-ID requires that the caller's phone line, the phone system, the remote access server's phone line, and the Windows 2000 driver for the dial-up equipment all support Caller-ID. If a Caller-ID is configured for a user account and the Caller-ID is not being passed from the caller to the Routing and Remote Access service, then the connection is denied.

Caller-ID is a feature designed to provide a higher degree of security for network that support telecommuters. The disadvantage of configuring Caller-ID is that the user can only dial-in from a single phone line.

Remote Access Account Lockout

The remote access account lockout feature is used to specify how many times an remote access authentication fails against a valid user account before the user is denied remote access. Remote access account lockout is especially important for remote access virtual private network (VPN) connections over the Internet. Malicious users on the Internet can attempt to access an organization intranet by sending credentials (valid user name, guessed password) during the VPN connection authentication process. During a dictionary attack, the malicious user sends hundreds or thousands of credentials by using a list of passwords based on common words or phrases. With remote access account lockout enabled, a dictionary attack is thwarted after a specified number of failed attempts.

The remote access account lockout feature does not distinguish between malicious users who attempt to access your intranet and authentic users who attempt remote access but have forgotten their current passwords. Users who have forgotten their current password typically try the passwords that they remember and, depending on the number of attempts and the MaxDenials setting, may have their accounts locked out.

If you enable the remote access account lockout feature, a malicious user can deliberately force an account to be locked out by attempting multiple authentications with the user account until the account is locked out, thereby preventing the authentic user from being able to log on.

As the network administrator, you must decide on two remote access account lockout variables:

1. The number of failed attempts before future attempts are denied.

   After each failed attempt, a failed attempts counter for the user account is incremented. If the user account's failed attempts counter reaches the configured maximum, future attempts to connect are denied.

   A successful authentication resets the failed attempts counter when its value is less than the configured maximum. In other words, the failed attempts counter does not accumulate beyond a successful authentication.

2. How often the failed attempts counter is reset.

   You must periodically reset the failed attempts counter to prevent inadvertent lockouts due to normal mistakes by users when typing in their passwords.

Remote access account lockout feature is configured by changing settings in the Windows 2000 registry on the computer that provides the authentication. If the remote access server is configured for Windows authentication, modify the registry on the remote access server computer. If the remote access server is configured for RADIUS authentication and Windows 2000 Internet Authentication Service (IAS) is being used, modify the registry on the IAS server computer.

To enable account lockout, you must set the MaxDenials entry in the registry (HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \RemoteAccess\Parameters\AccountLockout) to 1 or greater. MaxDenials is the maximum number of failed attempts before the account is locked out. By default, MaxDenials is set to 0, which means that account lockout is disabled.

To modify the amount of time before the failed attempts counter is reset, you must set the ResetTime (mins) entry in the registry (HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \RemoteAccess\Parameters\AccountLockout) to the required number of minutes. By default, ResetTime (mins) is set to 0xb40, or 2,880 minutes (48 hours).

To manually reset a user account that has been locked out before the failed attempts counter is automatically reset, delete the following registry subkey that corresponds to the user's account name:

HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \RemoteAccess \Parameters \AccountLockout \domain name:*user name*

Note The remote access account lockout feature is not related to the **Account locked out** setting on the **Account** tab on the properties of a user account and the administration of account lockout policies using Windows 2000 group policies.

Managing Remote Access

Remote access has the following management issues:

- Where is the user account data to be stored?
- How are addresses assigned to remote access clients?
- Who is allowed to create remote access connections?
- How does the remote access server verify the identity of the user attempting the remote access connection?
- How does the remote access server record the remote access activity?
- How can the remote access server be managed using industry-standard network management protocols and infrastructure?

Managing Users

Because it is administratively unsupportable to have separate user accounts for the same user on separate servers and to try to keep them all simultaneously current, most administrators set up a master account database at a domain controller (PDC) or on a Remote Authentication Dial-in User Service (RADIUS) server. This allows the remote access server to send the authentication credentials to a central authenticating device.

Managing Addresses

For PPP connections, IP, IPX, and AppleTalk addressing information must be allocated to remote access clients during the connection establishment process. The Windows 2000 remote access server must be configured to allocate IP addresses, IPX network and node addresses, and AppleTalk network and node addresses.

More information about address allocation for IP and IPX can be found later in this chapter.

Managing Access

In Windows NT versions 3.5*x* and 4.0, authorization was based on a simple **Grant dial-in permission to user** option in User Manager or the Remote Access Admin utility. Callback options were also configured on a per-user basis. In Windows 2000, authorization is granted based on the dial-in properties of a user account and remote access policies.

Remote access policies are a set of conditions and connection settings that give network administrators more flexibility in authorizing connection attempts. The Windows 2000 Routing and Remote Access service and Windows 2000 Internet Authentication Service (IAS) both use remote access policies to determine whether to accept or reject connection attempts. For more information about remote access policies, see "Internet Authentication Service" in this book.

With remote access policies, you can grant remote access by individual user account or through the configuration of specific remote access policies.

Access by User Account

Access by user account is the administrative model used in Windows NT version 3.5*x* and 4.0. In Windows 2000, if you wish to manage remote access on an individual per-user basis, set the remote access permission on those user accounts that are allowed to create remote access connections to **Allow access** and modify the profile properties of the default remote access policy called **Allow access if dial-in permission is enabled** for the needed connection parameters.

If the remote access server is only providing dial-up remote access connections and no VPN connections, then delete the default remote access policy called **Allow access if dial-in permission is enabled** and create a new remote access policy with a descriptive name, such as **Dial-up remote access if dial-in permission is enabled**.

As an example of typical settings to allow dial-up remote access connections, configure the remote access policy permission to **Deny remote access permission** and set the conditions and profile settings as listed in Tables 7.2 and 7.3. For detailed information about configuring these settings, see Windows 2000 Server Help.

Table 7.2 Remote Access Policy Conditions for Dial-Up Access by User Account

Conditions	Setting
NAS-Port-Type	Select all except **Virtual**. (example)

Table 7.3 Remote Access Policy Profile Settings for Dial-Up Access by User Account

Profile Tab	Setting
Authentication tab	Enable **Microsoft encrypted authentication version 2 (MS-CHAP v2)** and **Microsoft encrypted authentication (MS-CHAP)**. (example)

Access by Policy

The access by policy administrative model is intended for Windows 2000 remote access servers that are either standalone or a member of a Windows 2000 native mode domain. To manage remote access by policy, set the remote access permission on all user accounts to **Control access through Remote Access Policy**. Then define the new remote access policies that allow or deny access based on your needs. If the remote access server computer is a member of a Windows NT 4.0 domain or a Windows 2000 mixed domain and you want to manage access by policy, set the remote access permission on all user accounts to **Allow access**. Then, remove the default policy called **Allow access if dial-in permission is enabled** and create new policies that allow or deny access. A connection that does not match any configured remote access policy is denied, even if the remote access permission on the user account is set to **Allow access**.

A typical use of policy-based access is to allow access through group membership. For example, create a Windows 2000 group with a name, such as DialUpUsers, whose members are those users who are allowed to create dial-up remote access connections.

To create a remote access server that only allows dial-up remote access connections, delete the default remote access policy called **Allow access if dial-in permission is enabled** and then create a new remote access policy with a descriptive name, such as **Dial-up remote access if member of DialUpUsers group**.

As an example of typical settings to allow dial-up remote access for only members of a specific group, configure the remote access policy permission to **Grant remote access permission** and set the conditions and profile settings as listed in Tables 7.4 and 7.5. For detailed information about configuring these settings, see Windows 2000 Server Help.

Table 7.4 Remote Access Policy Conditions for Dial-Up Access by User Account

Conditions	Setting
NAS-Port-Type	Select all except **Virtual**.
Windows-Groups	DialUpUsers (example)

Table 7.5 Remote Access Policy Profile Settings for Dial-Up Access by User Account

Profile Tab	Setting
Authentication tab	Enable **Microsoft encrypted authentication version 2 (MS-CHAP v2)** and **Microsoft encrypted authentication (MS-CHAP)** (example)

Managing Authentication

The remote access server can be configured to use either Windows or RADIUS as an authentication provider.

Windows Authentication

If Windows is selected as the authentication provider, then the user credentials sent by users attempting remote access connections are authenticated using normal Windows authentication mechanisms.

If the remote access server is a member server in mixed or native Windows 2000 domain and is configured for Windows authentication, the computer account of the remote access server computer must be a member of the RAS and IAS Servers security group. This can be done by a domain administrator with the Active Directory User and Groups snap-in or with the **netsh ras add registeredserver** command before the installation of the Routing and Remote Access server. If the user installing the Routing and Remote Access service is a domain administrator, then the computer account is automatically added to the RAS and IAS Servers security group during the installation of the Routing and Remote Access service.

RADIUS Authentication

If RADIUS is selected and configured as the authentication provider on the remote access server, then user credentials and parameters of the connection request are sent as a series of RADIUS request messages to a RADIUS server such as a computer running Windows 2000 Server and the Internet Authentication Service (IAS).

The RADIUS server receives a user-connection request from the remote access server and authenticates the client against its authentication database. A RADIUS server can also maintain a central storage database of other relevant user properties. In addition to the simple yes or no response to an authentication request, RADIUS can inform the remote access server of other applicable connection parameters for this user - such as maximum session time, static IP address assignment, and so on.

RADIUS can respond to authentication requests based upon its own database, or it can be a front end to another database server such as a generic Open Database Connectivity (ODBC) server or a Windows 2000 PDC. The latter server could be located on the same machine as the RADIUS server, or could be centralized elsewhere. In addition, a RADIUS server can act as a proxy client to a remote RADIUS server.

The RADIUS protocol is described in RFCs 2138 and 2139. For more information about remote access server authentication scenarios and the remote access server as a RADIUS client, see Windows 2000 Server Help. For more information about IAS, see "Internet Authentication Service" in this book.

Note Both the Routing and Remote Access service when configured for Windows authentication and IAS use the same process to provide authentication and authorization of incoming connection requests. For more information on this process, see "Internet Authentication Service" in this book.

Managing Accounting

The remote access server can be configured to use either Windows or RADIUS as an accounting provider. If Windows is selected as the accounting provider, then the accounting information is accumulated in a log file on the remote access server. If RADIUS is selected as the accounting provider, RADIUS accounting messages are sent to the RADIUS server for accumulation and later analysis.

Most RADIUS servers can be configured to place authentication request records into an accounting file. There are also a set of messages (from the remote access server to the RADIUS server) that request accounting records at the start of a call, the end of a call, and at predetermined intervals during a call. A number of third parties have written billing and audit packages that read these RADIUS accounting records and produce various useful reports.

Network Management

The computer acting as the remote access server can participate in a Simple Network Management Protocol (SNMP) environment as an SNMP agent by installing the Windows 2000 SNMP Service. The remote access server records management information in various object identifiers of the Internet Management Information Base (MIB) II that is installed with the Windows 2000 SNMP service. Objects in the Internet MIB II are documented in RFC 1213.

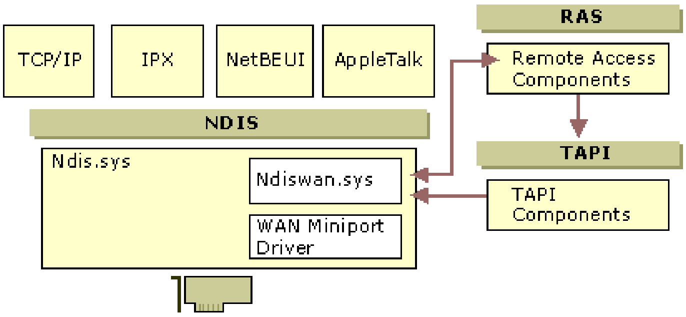
Remote Access Server Architecture

The architecture of the remote access server consists of the following elements, as illustrated in Figure 7.7:

- The NDIS wrapper, Ndis.sys, providing the NDIS packet-level interface to protocols, such as TCP/IP, IPX, NetBEUI, and AppleTalk.
- The NDISWAN driver, Ndiswan.sys, is an intermediate NDIS driver that provides an IEEE 802.3 miniport interface to protocol drivers

and a protocol interface to WAN miniport drivers. NDISWAN provides framing, compression, and encryption services for remote access connections.

- WAN miniport drivers are NDIS miniport drivers that contain the necessary code to operate the dial-up equipment. In order to use an adapter supporting WAN media, such as ISDN or ATM with Windows 2000 remote access, the adapter vendor must create a WAN miniport driver.
- Remote access components are a series of libraries that provide the Remote Access Service (RAS) programming interface for applications, PPP protocols (link control, authentication, and network control protocols), and so on. Remote access components can communicate directly with the NDISWAN driver or by accessing the Telephony API (TAPI).
- TAPI components are a series of libraries that provide a call control programming interface for all TAPI-aware applications. TAPI components communicate directly with the NDISWAN driver to manage connections. For more information about TAPI in Windows 2000, see "Telephony Conferencing and Integration" in this book.



If your browser does not support inline frames, click here to view on a separate page.

Figure 7.7 Remote Access Architecture in Windows 2000

Connections are established by remote access clients that call the RAS programming interface, which in turn uses TAPI to pass call connection information to the dial-up equipment. Once the physical connection is made, TAPI is no longer used and additional remote access components negotiate the connection with link, authentication, and network control protocols by communicating directly with NDISWAN.

Once a remote access connection is established, protocol drivers can communicate over that connection using standard NDIS calls like NdisSend(). NdisSend() calls for dial-up connections are forwarded to NDISWAN, which then determines the appropriate device and port, performs compression and encryption, provides PPP framing, and then forwards the completed frame to the WAN miniport driver. The WAN miniport driver then forwards the frame to the dial-up adapter.

All inbound remote access client connections, initiated by remote access clients to the remote access server, are represented as a single adapter called the RAS server interface. For each outbound remote access client connection, initiated by the remote access server, a separate interface is created.

To accept calls, the remote access server instructs each WAN miniport driver to indicate when it goes into a line-up state. When the call is placed, the WAN miniport driver passes the line-up state indicator up through NDISWAN to the TAPI components. TAPI returns a call handle to NDISWAN to be used to refer to the physical connection, and then NDISWAN and the remote access components negotiate the rest of the remote access connection.

IP, IPX, and AppleTalk Router

Once the remote access connection is established, the remote access client can begin sending LAN protocol traffic to the remote access server or to locations beyond the remote access server. When the remote access client sends LAN protocol traffic that is not destined for the remote access server, the remote access server must forward the LAN traffic to its appropriate destination. To accomplish this, the remote access server must have forwarding capabilities enabled on its routable protocols and act as an IP, IPX, and AppleTalk router.

When the Routing and Remote Access service is installed and enabled to provide point-to-LAN remote access connectivity, it enables forwarding between the installed LAN adapters and the WAN miniport interface.

Figure 7.8 illustrates the remote access server architecture as it appears when routing packets. (In an effort to simplify the illustration, only IP routing is shown.) However, IPX and AppleTalk routing work in the same fashion.

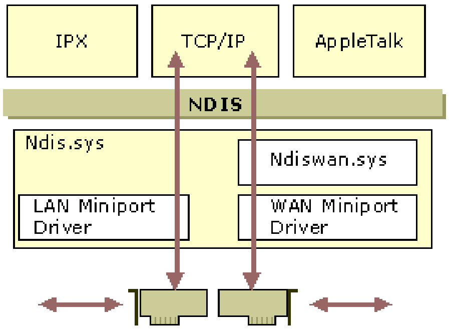


Figure 7.8 IP Routing on the Remote Access Server

Packets from Remote Access Clients

The following process describes how IP packets sent by the remote access client are forwarded by the remote access server.

1. Depending on the dial-up technology, either the entire PPP frame is received by the WAN hardware and passed up as a single frame to the appropriate WAN miniport driver or individual bits of the PPP frame are passed up to the appropriate WAN miniport driver.
2. The WAN miniport driver passes the PPP frame to Ndiswan.sys.
3. Ndiswan.sys verifies the PPP checksum and uses the PPP protocol ID to determine that it is an IP datagram. For more information about PPP, see "Point-to-Point Protocol" later in this chapter.
4. The IP datagram is passed to the TCP/IP protocol driver.
5. The TCP/IP protocol driver, which is enabled for IP forwarding, determines a forwarding interface and an IP address based on the

destination IP address in the IP datagram and the contents of its routing table.

6. To forward the IP datagram using the LAN adapter, the TCP/IP protocol calls NDIS with an NdisSend(), along with instructions to send it using the LAN adapter.
7. NDIS forwards the IP datagram to the appropriate LAN miniport driver.
8. The LAN miniport forwards the IP datagram to the LAN adapter through NDIS.

The end result is that packets from the remote access client are forwarded using the same IP routing process used for all IP routing. The success of the IP forwarding process depends on whether the remote access server can find a suitable entry in the IP routing table. Therefore, either the remote access server is configured with a default gateway, or the remote access server has specific routes to all the locations on the intranet to which the remote access server is attached. Specific routes can be added through static routes, or by enabling a routing protocol on the remote access server.

Packets to Remote Access Clients

The following process describes how IP packets sent by intranet hosts to the remote access client are forwarded by the remote access server.

1. The LAN adapter passes a frame to its appropriate LAN miniport driver through NDIS. The details of how an IP datagram is forwarded to the MAC address of the remote access server can be found in the next section, "TCP/IP On-Subnet and Off-Subnet Addressing."
2. The LAN miniport driver passes the IP datagram to the TCP/IP protocol driver through NDIS.
3. The TCP/IP protocol driver, which is enabled for IP forwarding, determines a forwarding interface and IP address based on the destination IP address in the IP datagram and the contents of its routing table. When the remote access client connects, a host route is created in the IP routing table for the IP address allocated to the remote access client that points to the RAS server interface.
4. To forward the IP datagram using the WAN adapter, the TCP/IP protocol calls NDIS with an NdisSend() with instructions to send it using NDISWAN and a specific connection handle.
5. NDISWAN resolves the connection handle to a specific device and port, adds a PPP header and trailer, and forwards the IP datagram to the appropriate WAN miniport driver through NDIS.
6. The WAN miniport driver forwards the IP datagram to the WAN adapter through NDIS.

The end result is that packets from intranet hosts are forwarded using the same IP routing process used for all IP routing. The success of the IP forwarding process depends on whether the IP addresses of remote access clients are reachable from the hosts on the intranet.

TCP/IP On-Subnet and Off-Subnet Addressing

The exact mechanism of how an IP node on a subnet to which the remote access server is attached resolves the media access control (MAC) address of the LAN interface of the remote access server depends on whether the remote access server is configured for on-subnet or off-subnet addressing:

- On-subnet addressing is the allocation of IP addresses to remote access clients that are in a range defined by a subnet to which the remote access server is attached. On-subnet addressing uses a subset of addresses of an attached subnet.
- Off-subnet addressing is the allocation of IP addresses to remote access clients that are not in a range defined by a subnet to which the remote access server is attached. Off-subnet addressing uses a separate subnet address space that is unique to the intranet.

On-Subnet Addressing and Proxy ARP

With on-subnet addressing, remote access clients are logically on the same subnet as a subnet attached to the remote access server. Proxy ARP is used by the remote access server to receive IP datagrams being forwarded to remote access clients.

There are two cases where Proxy ARP is used:

1. When the remote access server is configured to use DHCP to obtain addresses for IP-based remote access clients
2. When the remote access server is configured with a static IP address pool consisting of address ranges that are a subset of the addresses for a subnet to which the remote access server is attached.

In either case, the remote access clients are logically on the same subnet as the remote access server. Therefore, IP nodes on that subnet forwarding IP datagrams to a remote access client perform a direct delivery by sending a broadcast Address Resolution Protocol (ARP) Request frame for the remote access client's IP address.

The remote access client cannot respond to the ARP Request because the remote access server does not forward the ARP Request frame to the remote access client, and the remote access client does not have a media access control (MAC) address corresponding to the remote access connection.

Therefore, the remote access server responds with an ARP Reply frame with its own MAC address. The node forwarding the packet then sends the IP datagram to the remote access server's MAC address. The remote access server then uses the IP routing process to forward the IP datagram across the dial-up connection to the remote access client.

Off-Subnet Addressing and IP Routing

With off-subnet addressing, remote access clients are logically on a separate subnet reachable across the remote access server. In this case, Proxy ARP is not used. The remote access server is acting as a router between the subnet of the remote access clients and the subnets to which the remote access server is attached. IP nodes on the LAN-based subnets attached to the remote access server forwarding IP datagrams to a remote access client perform an indirect delivery by sending a broadcast Address Resolution Protocol (ARP) Request frame for the remote access server's IP address.

In order for the remote access clients to be reachable from IP nodes on the intranet, routes representing the address ranges of the IP address pool and pointing to the LAN interface of the remote access server must be present in intranet routers.

When the first TCP/IP-based remote access client connects, routes corresponding to the off-subnet address ranges pointing to the RAS server interface are added to the IP routing table of the remote access server. If the remote access server is configured with an IP routing protocol, the new routes are advertised to neighboring routers using the normal advertising process of the configured routing protocol. If the remote access server is not configured with an IP routing protocol, routes corresponding to the off-subnet address ranges pointing to the remote access server's LAN interface must be added to the routers of the intranet.

NetBIOS Gateway

Windows 2000 includes the NetBIOS gateway for remote access clients that use either the NetBEUI protocol with the PPP remote access protocol or the AsyBEUI remote access protocol. With the NetBIOS gateway, a remote access client using NetBEUI can access any NetBIOS-based network resource that is reachable from the remote access server.

With the NetBIOS gateway, the remote access client can access any of the following resources:

- Network resources available from the remote access server using NetBEUI
- Network resources available from the remote access server using NetBIOS over TCP/IP
- Network resources available from the remote access server using NetBIOS over IPX

The NetBIOS gateway component is responsible for:

- Managing NetBIOS names.

  When the initial connection is made, the remote access client passes its NetBIOS name, which is then added to the NetBIOS name table at the remote access server.
- Passing NetBIOS packets from the remote access client to the LAN.

  When the remote access client sends NetBEUI packets across the phone line, those packets are submitted to the NetBIOS gateway and sent over the NetBIOS providing protocols.
- Passing NetBIOS packets from the LAN to the remote access client.

  NetBIOS packets from the LAN, from any of the NetBIOS-providing protocols, are inspected for the NetBIOS computer name of the remote access client and sent back to the remote access client using NetBEUI.

Note The NetBIOS gateway cannot be used to access non-NetBIOS resources, such as Web servers, FTP servers, and other types of Windows Sockets-based resources.

Figure 7.9 illustrates the NetBIOS gateway architecture of the remote access server.

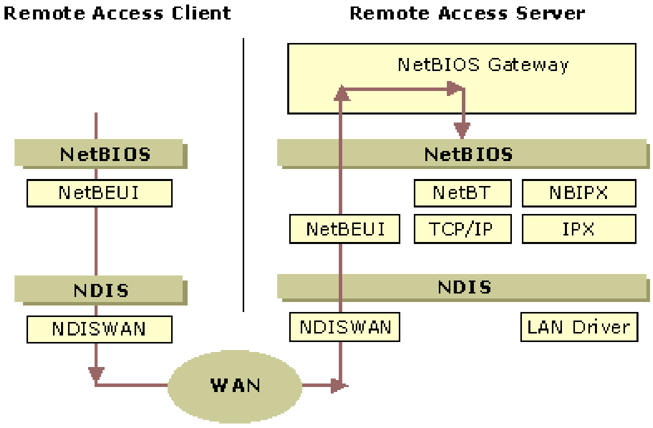


Figure 7.9 NetBIOS Gateway

Point-to-Point Protocol

The Point-to-Point Protocol (PPP) is an industry standard method of utilizing point-to-point links to transport multi-protocol datagrams. PPP is documented in RFC 1661. The Routing and Remote Access service stores PPP settings in the Windows 2000 registry under HKLM\System\CurrentControlSet\Services\RASMan\PPP.

PPP performs the following functions:

- Provides multi-protocol data-link layer encapsulation

  PPP creates frames that contain separate IP datagrams, IPX datagrams, or NetBEUI frames.
- Establishes, maintains, and ends the logical link

  The PPP protocol uses the Link Control Protocol (LCP) to establish and configure the parameters of the data-link connection. Part of the LCP negotiation is authenticating the credentials of the remote access client.
- Provides protocol configuration

  After the data-link connection has been negotiated, network layer protocols such as IP, IPX, and AppleTalk are configured. For example, for TCP/IP, an IP address is allocated to the remote access client by the remote access server. Compression and encryption are also negotiated.

PPP Encapsulation

PPP encapsulation uses a variant of the ISO High Level Data Link Control (HDLC) protocol to encapsulate multi-protocol datagrams as the payload of PPP frames. The PPP header and trailer is shown in Figure 7.10 and contains the following fields:

- **Flag** - Set to 0x7E (bit sequence 011111110) to signify the start and end of a PPP frame. In successive PPP frames only a single Flag character is used.
- **Address** - In HDLC environments, the Address field is used to address the frame to the destination node. On a point-to-point link, the destination node does not need to be addressed. Therefore, for PPP, the Address field is set to 0xFF, the broadcast address. If both PPP peers agree to perform address and control field compression during LCP negotiation, the Address field is not included.
- **Control** - In HDLC environments, the Control field is used for data-link layer sequencing and acknowledgments. PPP does not provided link-to-link reliable data transfer. Therefore, for all PPP frames, the Control field is set to 0x03 to indicate an unnumbered information (UI) frame. If both PPP peers agree to perform address and control field compression during LCP negotiation, the Control field is not included.
- **Protocol ID** - The 2-byte Protocol ID field identifies the protocol of the PPP payload. If both PPP peers agree to perform protocol field compression during LCP negotiation, the Protocol ID field is one byte for Protocol IDs in the range 0x00-00 to 0x00-FF.
- **Frame Check Sequence (FCS)** - A 16-bit checksum that is used to check for bit level errors in the PPP frame. If the receiver's calculation of the FCS does not match the FCS in the PPP frame, the PPP frame is silently discarded.

The maximum size of a PPP frame, known as the maximum receive unit (MRU), is determined during the negotiation of the logical link. The default MRU is 1,500 bytes. If negotiated lower, a PPP host must still have the ability to receive 1,500-byte frames in the event of link synchronization failure.

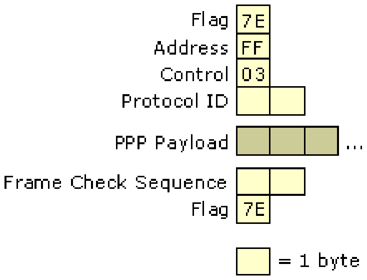


Figure 7.10 PPP Encapsulation

Typical values for the PPP protocol ID are listed in Table 7.6.

Table 7.6 PPP Protocol IDs

Protocol	**Protocol ID/Compressed Value**
Internet Protocol (IP)	0x00-21 / 0x21
AppleTalk	0x00-29 / 0x29
IPX	0x00-2B / 0x2B
Van Jacobsen Compressed TCP/IP	0x00-2D / 0x2D
Multilink	0x00-3D / 0x3D
NetBEUI	0x00-3F / 0x3F
Microsoft Point-to-Point Compression Protocol (MPPC)	0x00-FD / 0xFD
Microsoft Point-to-Point Encryption Protocol (MPPE)	0x00-FD / 0xFD

If MPPE or MPPC are negotiated, then the PPP protocol ID is set to 0x00-FD. With MPPE and MPPC both using the same PPP Protocol ID, each peer must know that the resulting PPP payload either is encrypted or compressed, or both.

- If only MPPC is negotiated, then the PPP Protocol ID is set to 0x00-FD and the PPP payload is compressed.
- If only MPPE is negotiated, then the PPP Protocol ID is set to 0x00-FD and the PPP payload is encrypted.
- If both MPPC and MPPE are negotiated, then compression always occurs before encryption. The compressed PPP frame, consisting of the PPP protocol ID field set to 0xFD and the compressed data, is then encrypted and encapsulated with another PPP header consisting of the protocol ID field set to 0xFD and a 2-byte MPPE header.

Preventing the Occurrence of the Flag Character

The use of the Flag character introduces a problem. What if the Flag character (0x7E) appears elsewhere in the PPP frame besides the beginning or end of the PPP frame? PPP employs two different methods to prevent the occurrence of the Flag character depending on whether PPP is being used on an asynchronous link or a synchronous link.

PPP on Asynchronous Links

On asynchronous links, such as analog phone lines, PPP uses a technique called character stuffing to prevent the occurrence of the Flag character within the PPP frame. If the Flag character (0x7E) occurs elsewhere in the PPP frame, the sender replaces it with the sequence 0x7D-5E. The 0x7D character is known as the PPP Escape character. If the PPP Escape character occurs, the sender replaces it with the sequence 0x7D-5D. The receiver translates 0x7D-5E sequences back to 0x7E and 0x7D-5D sequences back to 0x7D.

Additionally, character values less than 0x20 can be modified to prevent the serial drivers from interpreting them as control characters. If negotiated by LCP, characters below 0x20 are modified by sending the sequence: 0x7D-[Original character with the 6th bit complemented]. For example, the byte 0x01 would be transmitted as 0x7D-0x21.

PPP on Synchronous Links

With synchronous links, such as T-Carrier, ISDN, or other digital links, a technique called bit stuffing is used to prevent the occurrence of the Flag character within the PPP frame. Recall that the Flag character is the bit sequence 01111110. Bit stuffing ensures that six 1 bits in a row occur only when the Flag character is sent. To accomplish this, bit stuffing sends the Flag character unmodified and elsewhere inserts a 0 bit whenever a sequence of five 1 bits occurs. Therefore, the bit sequence 111111 is encoded on the medium as 1111101 and the bit sequence 111110 is encoded as 1111100 (the stuffed bits are underlined). Bit stuffing means that a byte can be encoded on the medium as more than eight bits, but the stuffed bits are added and removed by the synchronous link hardware.

PPP Link Negotiation with LCP

The PPP Link Control Protocol (LCP) is documented in RFC 1661. LPC negotiates link and PPP parameters to dynamically configure the data link layer of a PPP connection. Common LCP options include the PPP MRU, the authentication protocol, compression of PPP header fields, callback, and multilink options.

LCP Packet Structure

LCP uses the PPP Protocol ID of 0xC0-21. The packet structure of LCP is illustrated in Figure 7.11. Each LCP packet is a single LCP message consisting of an LCP Code field identifying the type of LCP packet, an Identifier field so that requests and replies can be matched, a Length field indicating the size of the LCP packet and LCP packet type-specific data.

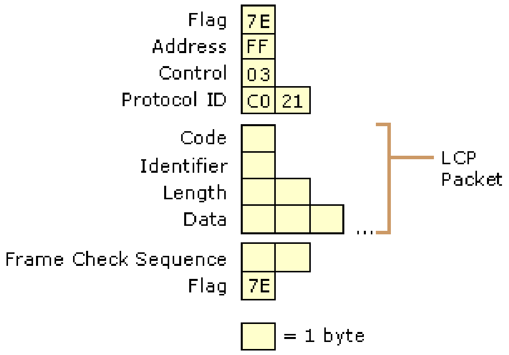


Figure 7.11 LCP Packet Structure

Table 7.7 lists the LCP packet types documented in RFC 1661.

Table 7.7 LCP Packet Types

LCP Code	LCP Packet Type	Description
1	Configure-Request	Sent to open or reset a PPP connection. Configure-Request contains a list of LCP options with changes to default option values.
2	Configure-Ack	Sent when all of the values of all of the LCP options in the last Configure-Request received are recognized and acceptable. When both PPP peers send and receive Configure-Acks, the LCP negotiation is complete.
3	Configure-Nack	Sent when all the LCP options are recognized, but the values of some options are not acceptable. Configure-Nack includes the offending options and their acceptable values.
4	Configure-Reject	Sent when LCP options are not recognized or not acceptable for negotiation. Configure-Reject includes the unrecognized or non-negotiable options.
5	Terminate-Request	Optionally sent to close the PPP connection.
6	Terminate-Ack	Sent in response to the Terminate-Request.
7	Code-Reject	Sent when the LCP code is unknown. The Code-Reject message includes the offending LCP packet.
8	Protocol-Reject	Sent when the PPP frame contains an unknown Protocol ID. The Protocol-Reject message includes the offending LCP packet. Protocol-Reject is typically sent by a PPP peer in response to a PPP NCP for a LAN protocol not enabled on the PPP peer.
9	Echo-Request	Optionally sent to test the PPP connection.
10	Echo-Reply	Sent in response to an Echo-Request. The PPP Echo-Request and Echo-Reply are not related to the ICMP Echo Request and Echo Reply messages.
11	Discard-Request	Optionally sent to exercise the link in the outbound direction.

LCP Options

When using the Configure-Request, Configure-Ack, Configure-Nack, and Configure-Reject LCP packet types, the LCP data portion of the LCP packet consists of one or more LCP options as illustrated in Figure 7.12. Each LCP option consists of an option Type field, a Length field indicating the total length in bytes of the option and the data associated with the option.

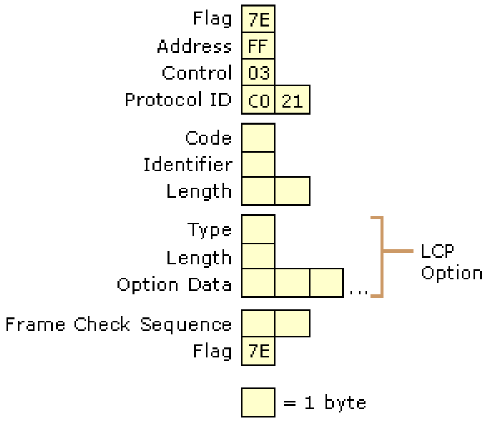


Figure 7.12 LCP Options

Table 7.8 lists common LCP options negotiated by Microsoft PPP peers. For information about other LCP options, see RFC 1661.

Table 7.8 LCP Options

Option Name	Option Type	Option Length	Description
Maximum Receive Unit (MRU)	1	4	The maximum size (up to 65,535) of the PPP frame. The default MRU is 1,500. If neither peer is changing the default, this option is not negotiated.
Asynchronous Control	2	6	A bit map that enables (bit set to 1) or disables (bit set to 0) the use of character

Character Map (ACCM)			escapes for asynchronous links for the 32 ASCII control characters from 0x00 to 0x20. By default, character escapes are used. The ACCM bit map is set to 0x00-00-00-00 for links with XON/XOFF software flow control.
Authentication Protocol	3	5 or 6	Indicates the authentication protocol used during the authentication phase of the connection. Values for this field for Microsoft PPP peers are 0xC2-27 for EAP, 0xC2-23-80 for MS-CHAP version 1, 0xC2-23-81 for MS-CHAP version 2, 0xC2-23-05 for MD5-CHAP, 0xC0-27 for SPAP, and 0xC0-23 for PAP.
Magic Number	5	6	A random number chosen to distinguish a peer and detect looped back lines.
Protocol Compression	7	2	A flag indicating that the PPP protocol ID be compressed to a single octet when the 2-byte protocol ID is in the range 0x00-00 to 0x00-FF.
Address and Control Field Compression	8	2	A flag indicating that the PPP Address field (always set to 0xFF) and the PPP Control field (always set to 0x03) be removed from the PPP header.
Callback	13 or 0x0D	3	A 1-octet indicator of how callback is to be determined. For remote access clients and server running Microsoft® Windows 32-bit operating systems, the callback option octet is set to 0x06, indicating that the callback is determined during Callback Control Protocol (CBCP) negotiation.

LCP Negotiation Process

The LCP negotiation is a series of LCP packets exchanged between PPP peers to negotiate a set of options and option values when sending data. The LCP negotiation is actually two separate dialogs between two PPP peers (Peer 1 and Peer 2):

1. Peer 1 asks, negotiates, and then receives confirmation of the LCP options that are used when sending data to Peer 2. This dialog starts with Peer 1 sending a Configure-Request message and ends when Peer 2 sends a Configure-Ack message.
2. Peer 2 asks, negotiates, and then receives confirmation of the LCP options that are used when sending data to peer 1. This dialog starts with Peer 2 sending a Configure-Request message and ends when Peer 1 sends a Configure-Ack message.

Peer 1 and Peer 2 do not have to use the same set of LCP options.

When a PPP peer sends its initial Configure-Request, the response is any of the following:

- A Configure-Nack message because one or more options have unacceptable values.
- A Configure-Reject message because one or more of the options are unknown or not negotiable.
- A Configure-Ack message because all of the options have acceptable values.

When a PPP peer receives a Configure-Nack message or Configure-Reject message in response to its Configure-Request message, it sends a new Configure-Request message with modified options or option values. When a Configure-Ack message is received, the PPP peer is ready to send data.

Figure 7.13 shows a hypothetical LCP negotiation process for Peer 1 using the fictional options W, X, Y, Z.

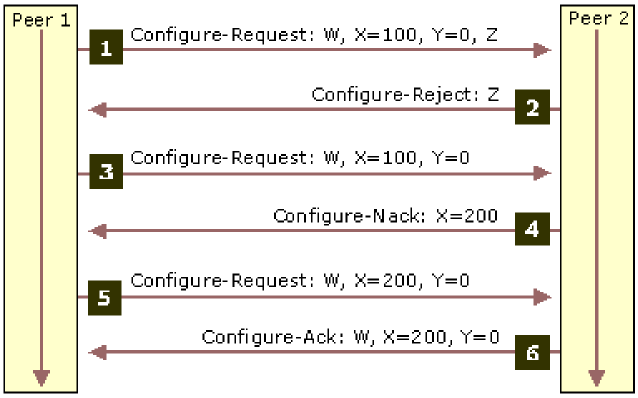

Peer 1
Peer 2
1
Configure-Request: W, X=100, Y=0, Z
2
Configure-Reject: Z
3
Configure-Request: W, X=100, Y=0
4
Configure-Nack: X=200
5
Configure-Request: W, X=200, Y=0
6
Configure-Ack: W, X=200, Y=0

Figure 7.13 Sample LCP Negotiation

From the LCP messages in Figure 7.13:

1. Peer 1 sends a Configure-Request message requesting option W, option X set to 100, option Y set to 0, and option Z. Options W and Z are flag options.
2. Peer 2 does not understand option Z so it sends a Configure-Reject message containing option Z.
3. Peer 1 sends a new Configure-Request message requesting option W, option X set to 100, and option Y set to 0.
4. Peer 2 prefers that option X be set to 200 so it sends a Configure-Nack message containing option X and its preferred value.
5. Peer 1 sends a new Configure-Request message requesting option W, option X set to 200, and option Y set to 0.
6. Peer 2 sends a Configure-Ack message.

Each time Peer 1 sends a new Configure-Request message, it changes the Identifier value in the LCP header so that Configure-Request messages can be matched with their responses.

The previous process only configures how Peer 1 sends data to Peer 2. A separate LCP negotiation must be done so that Peer 2 can be configured to send data to Peer 1. Very often, the LCP packets for the two dialogs are intermixed during the connection process. Peer 1 is configuring the way it sends data at the same time as Peer 2.

Callback Negotiation with the Callback Control Protocol

Callback Control Protocol (CBCP) negotiates the use of callback where the remote access server, after authenticating the remote access client, terminates the physical connection, waits a specified amount of time, and then calls the remote access client back at either a static or dynamically configured phone number. Common CBCP options include the phone number being used by the remote access server to call the remote access client back. For more information about CBCP, see "Proposal for Callback Control Protocol (CBCP)" at the Internet Working Group link on the Web Resources pages at http://windows.microsoft.com/windows2000/reskit/webresources .

Packet Structure

CBCP uses the PPP Protocol ID of 0xC0-29. The packet structure of CBCP is exactly the same for LCP; however, only the Callback-Request (type 1), Callback-Response (type 2), and Callback-Ack (type 3) types are used. For all CBCP packet types, the CBCP data portion of the CBCP packet consists of one or more CBCP options. Each CBCP option consists of an option Type field, an option Length field indicating the total length in bytes of the option, and the data associated with the option.

Negotiated Options

Table 7.9 lists the CBCP options negotiated by Microsoft PPP peers.

Table 7.9 CBCP Options

Option Name	Option Type	Option Length	Description
No callback	1	2	Specifies that no callback is used for the connection.
Callback to a user-specified number	2	variable	The user of the remote access client computer determines the callback number.
Callback to an administrator-defined number	3	variable	Settings on the remote access server determine the callback number.
Callback to any of a list of numbers	4	variable	The remote access server callbacks to one of a list of phone numbers.

PPP Network Layer Negotiation with NCP

Once the link and PPP parameters have been negotiated with LCP, the PPP peers then use a series of Network Control Protocols (NCPs) to negotiate the parameters of individual LAN protocols. Microsoft PPP supports the following NCPs:

- Internet Protocol Control Protocol (IPCP) to negotiate the use of IP.
- Internetwork Packet Exchange Control Protocol (IPXCP) to negotiate the use of IPX.
- AppleTalk Control Protocol (ATCP) to negotiate the use of AppleTalk.
- NetBIOS Frames Control Protocol (NBFCP) to negotiate the use of NetBEUI.

IPCP

Internet Protocol Control Protocol (IPCP) as used by Microsoft PPP peers is documented in RFCs 1332 and 1877. IPCP negotiates IP-based parameters to dynamically configure a TCP/IP-based PPP peer across a point-to-point link. Common IPCP options include an IP address and the IP addresses of DNS and NetBIOS name servers.

Packet Structure

IPCP uses the PPP Protocol ID of 0x80-21. The packet structure of IPCP is exactly the same for LCP, except only packet types 1 through 7 are defined. For Configure-Request, Configure-Ack, Configure-Nack, and Configure-Reject IPCP packet types, the IPCP data portion of the IPCP packet consists of one or more IPCP options. Each IPCP option consists of an Option Type field, an Option Length field indicating the total length in bytes of the option, and the data associated with the option.

Negotiated Options

Table 7.10 lists the IPCP options negotiated by Microsoft PPP peers.

Table 7.10 IPCP Options

Option Name	Option Type	Option Length	Description
IP compression protocol	2	4	Van Jacobsen TCP compression protocol.
IP address	3	6	The IP address to be allocated to the remote access client.
Primary DNS server address	129 or 0x81	6	The primary DNS server for the remote access client.
Primary NBNS server address	130 or 0x82	6	The primary NBNS (WINS) server for the remote access client.
Secondary DNS server address	131 or 0x83	6	The secondary DNS server for the remote access client.
Secondary NBNS server address	132 or 0x84	6	The secondary NBNS (WINS) server for the remote access client.

Notice that there are no IPCP options for these common TCP/IP configuration items:

- Subnet mask

  The subnet mask is assumed by the remote access client to be the class-based subnet mask of the IP address that is allocated to the remote access client.
- Default gateway

  The default gateway IP address is not allocated by the remote access server. However, a default route is created on the remote access client, which points to the remote access connection. If a default route already exists in the routing table, then the metric of the existing default route is increased and a new default route is added with a lower metric. This is the default behavior for remote access clients running Windows 32-bit operating systems and can be modified by disabling the **Use Default Gateway on Remote Network** setting on the TCP/IP properties of a remote access client's phone book entry or dial-up connection object.
- DNS domain name

  The DNS domain name configured from the TCP/IP protocol properties on the remote access server is not negotiated during IPCP. For Windows 2000 remote access clients, the DNS domain name can be obtained through a DHCPInform message. For more information, see "Remote Access and TCP/IP and IPX" later in this chapter.
- NetBIOS Node Type

  If the IP addresses of primary or secondary NetBIOS name servers are negotiated, then the hybrid NetBIOS node type (H-node) is assumed.

IPXCP

Internetwork Packet Exchange Control Protocol (IPXCP) as used by Microsoft PPP peers is documented in RFC 1552. IPXCP negotiates IPX-based parameters to dynamically configure an IPX-based PPP peer across a point-to-point link. Common IPXCP options include IPX network and node addresses.

Packet Structure

IPXCP uses the PPP Protocol ID of 0x80-2B. The packet structure of IPXCP is exactly the same for LCP, except only packet types 1 through 7 are defined. For Configure-Request, Configure-Ack, Configure-Nack, and Configure-Reject IPXCP packet types, the IPXCP data portion of the IPXCP packet consists of one or more IPXCP options. Each IPXCP option consists of an option Type field, an option Length field indicating the total length in bytes of the option, and the data associated with the option.

Negotiated Options

Table 7.11 lists the IPXCP options negotiated by Microsoft PPP peers.

Table 7.11 IPXCP Options

Option Name	Option Type	Option Length	Description
IPX Network Number	1	6	The IPX network number for the remote access client.
IPX Node Number	2	6	The IPX node number for the remote access client.

ATCP

AppleTalk Control Protocol (ATCP) as used by Microsoft PPP peers is documented in RFC 1378. ATCP negotiates AppleTalk-based parameters to dynamically configure an AppleTalk-based PPP peer across a point-to-point link. Common ATCP options include an AppleTalk address and server information.

Packet Structure

ATCP uses the PPP Protocol ID of 0x80-29. The packet structure of ATCP is exactly the same as LCP, except that only packet types 1 through 7 are defined. For Configure-Request, Configure-Ack, Configure-Nack, and Configure-Reject ATCP packet types, the ATCP data portion of the ATCP packet consists of one or more ATCP options. Each ATCP option consists of an option Type field, an option Length field indicating the total length in bytes of the option, and the data associated with the option.

Negotiated Options

Table 7.12 lists the ATCP options negotiated by Microsoft PPP peers.

Table 7.12 ATCP Options

Option Name	Option Type	Option Length	Description
AppleTalk Address	1	6	Negotiates the AppleTalk network and node numbers
Server Information	3	16	Used to convey information about the remote access server

NBFCP

NetBIOS Frames Control Protocol (NBFCP) as used by Microsoft PPP peers is documented in RFC 2097. NBFCP negotiates NetBEUI-based parameters to dynamically configure a NetBEUI-based PPP peer across a point-to-point link. Common NBFCP options include multicast filtering options and peer information.

Packet Structure

NBFCP uses the PPP Protocol ID of 0x80-3F. The packet structure of NBFCP is exactly the same for LCP, except that only packet types 1 through 7 are defined. For Configure-Request, Configure-Ack, Configure-Nack, and Configure-Reject NBFCP packet types, the NBFCP data portion of the NBFCP packet consists of one or more NBFCP options. Each NBFCP option consists of an option Type field, an option Length field indicating the total length in bytes of the option, and the data associated with the option.

Negotiated Options

Table 7.13 lists the NBFCP options negotiated by Microsoft PPP peers.

Table 7.13 NBFCP Options

Option Name	Option Type	Option Length	Description
Multicast filtering	3	5	Negotiates the handling of multicast packets
Peer information	2	17	Used to convey NetBIOS configuration information

Compression Control Protocol

Compression Control Protocol (CCP) is documented in RFC 1962. CCP negotiates parameters to dynamically configure, enable, and disable data compression algorithms between PPP peers across a point-to-point link. Common CCP options include an organization identifier and the use of MPPC.

Packet Structure

CCP uses the PPP Protocol ID of 0x80-FD. The packet structure of CCP is exactly the same for LCP, except only packet types 1 through 7 are defined. For Configure-Request, Configure-Ack, Configure-Nack, and Configure-Reject CCP packet types, the CCP data portion of the CCP packet consists of one or more CCP options. Each CCP option consists of an option Type field, an option Length field indicating the total length in bytes of the option, and the data associated with the option.

Negotiated Options

Table 7.14 lists the CCP options negotiated by Microsoft PPP peers.

Table 7.14 CCP Options

Option Name	Option Type	Option Length	Description
Organization Unique Identifier	0	6 or larger	Used to negotiate an organization's proprietary compression protocol.
MPPC	18 or 0x12	6	Used to indicate the use of MPPC, MPPE, and the encryption strength.

MPPE and MPPC

With CCP option 18, Microsoft PPP peers negotiate both MPPC and MPPE at the same time. The option data field for CCP option 18 is 4 bytes (32 bits) long. Bits within this data field are used as flags to indicate:

- Whether compression is enabled (0x00-00-00-01).
- Whether 40-bit session keys are derived from the LAN Manager version of the user's password (0x00-00-00-10).
- Whether 40-bit session keys are derived from the Windows NT version of the user's password (0x00-00-02-00).
- Whether 56-bit session keys are derived from the Windows NT version of the user's password (0x00-00-00-80).
- Whether 128-bit session keys are derived from the Windows NT version of the user's password (0x00-00-00-40).
- Whether the encryption keys are refreshed with each PPP frame (0x01-00-00-00).

For multiple choices, the flag values are added together. For example, for compression (0x00-00-00-01) and 128-bit encryption keys (0x00-00-00-40), the resulting 32-bit option data field is set to 0x00-00-00-41.

For more information about MPPE, see the Internet draft, "Microsoft Point-To-Point Encryption (MPPE) Protocol."

ECP

The Encryption Control Protocol (ECP) is used to negotiate a specific encryption method and is documented in RFC 1968. However, for Microsoft PPP peers, the only encryption that is supported is MPPE that is negotiated during CCP with the negotiation of MPPC. Therefore, Microsoft PPP peers do not use ECP.

PPP Connection Process

There are four distinct phases of negotiation of a PPP connection. Each of these four phases must complete successfully before the PPP connection is ready to transfer user data. The four phases of a PPP connection are:

1. PPP configuration
2. Authentication
3. Callback
4. Protocol configuration

Phase 1: PPP Configuration

PPP configures PPP protocol parameters using LCP (Link Control Protocol). During the initial LCP phase, each peer negotiates communication options that are used to send data and include:

- PPP parameters, such as MRU, address and control field compression, and protocol ID compression.
- Which authentication protocols are used to authenticate the remote access client. An authentication protocol is selected but not implemented until the authentication phase.
- Multilink options.

Phase 2: Authentication

After LCP is complete, the authentication protocol agreed upon by the remote access server and the remote access client is implemented. The nature of this traffic is specific to the PPP authentication protocol. For more details, see "PPP Authentication Protocols" later in this chapter.

Phase 3: Callback

The Microsoft implementation of PPP includes an optional callback phase using the Callback Control Protocol (CBCP) immediately after authentication. In order for a remote access client user to get called back, the dial-in properties of the user account must be enabled for callback and either the remote access client can specify the callback number or the remote access server must specify the callback number.

If a connection is implementing callback, both PPP peers hang up and the remote access server calls the remote access client at the negotiated number.

Phase 4: Protocol Configuration

Once the PPP is configured and callback is complete (optional), network layer protocols can be configured. With remote access on Windows 32-bit operating systems, the remote access server sends the remote access client Configuration-Request messages for all of the LAN protocols enabled for remote access on the remote access server. The remote access client either continues the negotiation of the LAN protocols enabled at the remote access client or sends an LCP Protocol-Reject message containing the Configuration-Request message.

IPCP, IPXCP, ATCP, and NBFCP each go through a negotiation process very similar to LCP negotiation to configure their corresponding network layer protocol. CCP packets are exchanged to configure MPPC and MPPE.

A Sample PPP Connection

To actually see the PPP connection establishment process in Windows 2000, there are two tools available:

1. Network Monitor, a packet capture and analysis tool, is used to capture all PPP packets sent over a serial link including connection establishment and PPP-encapsulated user data.
2. PPP tracing is used to create a log of the PPP packets exchanged during the PPP connection establishment process.

Network Monitor

To capture PPP packets with Network Monitor, set the capture network to the network corresponding to the dial-up connection and begin capturing PPP frames as desired. You can use Network Monitor to:

- Troubleshoot the PPP connection establishment process.
- Ensure that PPP payloads are being encrypted.

- Ensure that PPP payloads are being compressed.

Note If compression or encryption is being used, then the PPP payload is not interpreted by Network Monitor. Compressed or encrypted payloads are indicated with the PPP protocol ID of 3D (assuming protocol ID compression). To see the structure of user data within PPP payloads, disable compression and encryption.

When using Network Monitor, keep the following in mind:

- Captured PPP frames do not contain a Flag character but do contain an Ethernet-like source address and destination address. This behavior is due to the fact that Network Monitor receives the packets from the Ndiswan.sys driver. Recall that Ndiswan.sys is an intermediate NDIS driver that looks like an Ethernet adapter to protocols.

  For each PPP frame, the Ethernet-like source and destination addresses are both set to either **SEND** or **RECV** to indicate that the PPP frame was either sent or received by the computer on which the capture was taken. The **SEND** and **RECV** addresses do not necessarily identify the traffic of a remote access server or remote access client. If the capture was taken on the remote access server, then **SEND** frames were sent by the remote access server and **RECV** frames were sent by the remote access client. If the capture was taken on the remote access client, then **SEND** frames were sent by the remote access client and **RECV** frames were sent by the remote access server.
- Captured PPP frames contain an Address or Control field regardless of whether address and control field compression are negotiated.
- Protocol ID compression is usually negotiated with Microsoft PPP peers, making the PPP Protocol ID a single byte when possible.
- Use Network Monitor display to view only the traffic of desired protocols. For example, to view only the IPCP negotiation, set the display filters to disable the display of all protocols except IPCP.

The following printout is an example of a PPP connection establishment process captured with Network Monitor showing only the frame summaries. The entries are indented to improve readability.

```
1 8.726 SEND SEND LCP
Config Req Packet, Ident = 0x00, Length = 36
2 8.796 RECV RECV LCP
Config Req Packet, Ident = 0x00, Length = 25
3 8.796 SEND SEND LCP
Config Ack Packet, Ident = 0x00, Length = 25
4 8.816 RECV RECV LCP
Config Reject Packet, Ident = 0x00, Length = 17
5 8.816 SEND SEND LCP
Config Req Packet, Ident = 0x01, Length = 23
6 8.886 RECV RECV LCP
Config Ack Packet, Ident = 0x01, Length = 23
7 8.886 SEND SEND LCP
Ident Packet, Ident = 0x02, Length = 18
8 8.886 SEND SEND LCP
Ident Packet, Ident = 0x03, Length = 23
9 8.886 RECV RECV PPPCHAP
Challenge, ID = 0x 1: Challenge
10 8.886 SEND SEND PPPCHAP
Challenge, ID = 0x 1: Response, administrator
11 8.976 RECV RECV PPPCHAP
Challenge, ID = 0x 1: Success
12 8.976 RECV RECV CBCP
Callback Request, Ident = 0x01
13 8.976 SEND SEND CBCP
Callback Response, Ident = 0x01
14 8.996 RECV RECV CBCP
Callback Acknowledgement, Ident = 0x01
15 8.996 SEND SEND CCP
Configuration Request, Ident = 0x04
16 8.997 SEND SEND IPCP
Configuration Request, Ident = 0x05
17 8.997 RECV RECV CCP
Configuration Request, Ident = 0x01
18 9.017 RECV RECV IPCP
Configuration Request, Ident = 0x02
19 9.037 RECV RECV IPXCP
Configuration Request, Ident = 0x03
20 9.037 RECV RECV NBFCP
Configuration Request, Ident = 0x04
21 9.117 SEND SEND IPXCP
Configuration Request, Ident = 0x06
22 9.147 SEND SEND CCP
Configuration Acknowledgement, Ident = 0x01
23 9.147 SEND SEND IPCP
Configuration Acknowledgement, Ident = 0x02
24 9.167 SEND SEND IPXCP
Configuration Acknowledgement, Ident = 0x03
25 9.167 SEND SEND LCP
Protocol Reject Packet, Ident = 0x07, Length = 32
26 9.237 RECV RECV CCP
Configuration Reject, Ident = 0x04
27 9.237 RECV RECV IPCP
Configuration Reject, Ident = 0x05
28 9.237 SEND SEND IPCP
Configuration Request, Ident = 0x08
29 9.257 RECV RECV IPXCP
Configuration No Acknowledgement, Ident = 0x06
30 9.257 SEND SEND IPXCP
Configuration Request, Ident = 0x09
31 9.287 RECV RECV IPCP
Configuration No Acknowledgement, Ident = 0x08
32 9.287 SEND SEND IPCP
Configuration Request, Ident = 0x0A
33 9.287 RECV RECV IPXCP
Configuration Acknowledgement, Ident = 0x09
34 9.327 RECV RECV IPCP
Configuration Acknowledgement, Ident = 0x0A
```

```
35 10.729 SEND SEND CCP
Configuration Request, Ident = 0x04
36 10.960 RECV RECV CCP
Configuration Reject, Ident = 0x04
37 10.960 SEND SEND CCP
Configuration Request, Ident = 0x0B
38 10.960 RECV RECV CCP
Configuration Acknowledgement, Ident = 0x0B
```

The trace was captured on the remote access client. Therefore, the SEND frames were sent from the remote access client and the RECV frames were sent from the remote access server. In this trace, you can see the four phases of the PPP connection establishment:

- Phase 1: PPP configuration is done in frames 1 through 8 by using the exchange of LCP configuration packets.
- Phase 2: Authentication is done in frames 9 through 11 where the user's credentials are verified.
- Phase 3: Callback is done in frames 12 through 14.
- Phase 4: Protocol configuration is done in frames 15 through 38 where compression, encryption, IP, and IPX are configured.

In addition to the summary view, Network Monitor can also expand frames for detailed analysis. For example, frame 1 from this trace is displayed as:

```
FRAME: Base frame properties
FRAME: Time of capture = Nov 18, 1998 15:23:6.967
FRAME: Time delta from previous physical frame: 0 milliseconds
FRAME: Frame number: 1
FRAME: Total frame length: 50 bytes
FRAME: Capture frame length: 50 bytes
FRAME: Frame data: Number of data bytes remaining = 50 (0x0032)
PPP: Link Control Protocol Frame (0xC021)
PPP: Destination Address = SEND_
PPP: Source Address = SEND_
PPP: Protocol = Link Control Protocol
LCP: Config Req Packet, Ident = 0x00, Length = 36
LCP: Code = Configuration Request
LCP: Identifier = 0 (0x0)
LCP: Length = 36 (0x24)
LCP: Options: ASYNC.MAP:00 00 00 00-MAGIC#:0x0C05-PROT.COMP- ADR/CF.COMP-CALL.BACK:Unkn---
LCP: ASYNC.MAP:00 00 00 00
LCP: Option Type = Async Control Character Map
LCP: Option Length = 6 (0x6)
LCP: Async Control Character Map = 00 00 00 00
LCP: MAGIC#:0x0C05
LCP: Option Type = Majic Number
LCP: Option Length = 6 (0x6)
LCP: Magic Number = 3077 (0xC05)
LCP: PROT.COMP
LCP: Option Type = Protocol Field Compression
LCP: Option Length = 2 (0x2)
LCP: ADR/CF.COMP
LCP: Option Type = Address and Control Field Compression
LCP: Option Length = 2 (0x2)
LCP: CALL.BACK:Unkn
LCP: Option Type = Callback
LCP: Option Length = 3 (0x3)
LCP: CallBack = 0x06
LCP: Multilink Maximum Receive Reconstructed Unit
LCP: Option Type = 0x11
LCP: Option Length = 4 (0x4)
LCP: Multilink Endpoint Discriminator
LCP: Option Type = 0x13
LCP: Option Length = 9 (0x9)
```

Network Monitor captures can also be saved as files and sent to Microsoft support professionals for analysis.

PPP Tracing

Tracing is a facility of Windows 2000 remote access and routing components that allow you to optionally enable and disable the recording of programming code and network events to a file.

You enable PPP tracing by selecting **Enable Point-to-Point Protocol (PPP) Logging** from the **Event Logging** tab on the properties of a remote access server in the **Routing and Remote Access** snap-in.

The file Ppp.log is created in the %*Systemroot*%\tracing folder and contains information about the PPP connection establishment process. The PPP log generated by PPP tracing contain the programming calls and actual packet contents of PPP packets for PPP control protocols. PPP tracing cannot be used to view PPP user data sent across the connection.

The following printout is an excerpt from a PPP trace of a PPP connection establishment process. The entries are indented to improve readability.

```
[1472] 15:57:50:094: Line up event occurred on port 5
[1472] 15:57:50:104: Starting PPP on link with IfType=0x0,IPIf=0x0,
IPXIf=0x0
[1472] 15:57:50:104: RasGetBuffer returned ae70054 for SendBuf
[1472] 15:57:50:104: FsmInit called for protocol = c021, port = 5
[1472] 15:57:50:104: ConfigInfo = 273e
[1472] 15:57:50:104: APs available = 1
[1472] 15:57:50:104: FsmReset called for protocol = c021, port = 5
[1472] 15:57:50:104: Inserting port in bucket # 5
[1472] 15:57:50:104: Inserting bundle in bucket # 6
[1472] 15:57:50:104: FsmOpen event received for protocol c021 on port 5
[1472] 15:57:50:104: FsmThisLayerStarted called for protocol = c021,
port = 5
[1472] 15:57:50:104: FsmUp event received for protocol c021 on port 5
[1472] 15:57:50:104: <PPP packet sent at 11/04/1998 23:57:50:104
[1472] 15:57:50:104: <Protocol = LCP, Type = Configure-Req, Length =
0x2f, Id = 0x0, Port = 5
[1472] 15:57:50:104: <C0 21 01 00 00 2D 02 06 00 00 00 00 03 05 C2 23
|.!...-.........#|
[1472] 15:57:50:104: <80 05 06 72 5F 50 9A 07 02 08 02 0D 03 06 11 04
|...r_P..........|
[1472] 15:57:50:104: <06 4E 13 09 03 00 60 08 3E 46 07 17 04 00 03 00
|.N....`.>F......|
[1472] 15:57:50:104: InsertInTimerQ called portid=6,Id=0,Protocol=c021,
EventType=0,fAuth=0
[1472] 15:57:50:104: InsertInTimerQ called portid=6,Id=0,Protocol=0,
EventType=3,fAuth=0
[1472] 15:57:50:104: >PPP packet received at 11/04/1998 23:57:50:104
[1472] 15:57:50:104: >Protocol = LCP, Type = Configure-Req, Length =
0x26, Id = 0x0, Port = 5
[1472] 15:57:50:104: >C0 21 01 00 00 24 02 06 00 00 00 00 05 06 00 00
|.!...$..........|
[1472] 15:57:50:104: >C0 05 07 02 08 02 0D 03 06 11 04 06 4E 13 09 03
|._..........N...|
[1472] 15:57:50:104: >00 60 08 52 F9 D8 00 00 00 00 00 00 00 00 00 00
|.`.R............|
```

The last three lines of this trace excerpt is a hexadecimal display of the same LCP packet as frame 1 of the previous Network Monitor trace. To understand this frame, you must manually parse this frame according to the PPP and LCP packet structure. An example of the parsing of this PPP frame is listed in Table 7.15.

Table 7.15 Parsing of the LCP Configuration-Request

Bytes	Meaning
C0 21	PPP Protocol ID for LCP.
01	LCP Code for a Configure-Request.
00	LCP Identifier for this Configure-Request.
00 24	Length in bytes of the LCP packet (36 bytes long).
02	LCP option for Asynchronous Control Character Map (ACCM).
06	Length in bytes of the ACCM option.
00 00 00 00	Data for the ACCM option.
05	LCP option for the magic number.
06	Length in bytes of the magic number option.
00 00 C0 05	Data for the magic number option.
07	LCP option for protocol compression.
02	Length in bytes of the protocol compression option.
08	LCP option for address and control field compression.
02	Length in bytes of the address and control field compression option.
0D	LCP option for callback.
03	Length in bytes of the callback option.
06	Callback option data.
11	LCP option for the Multilink Maximum Receive Reconstructed Unit. Multilink LCP options are discussed in the "Multilink and Bandwidth Allocation Protocol" section of this chapter.
04	Length in bytes of the Multilink Maximum Receive Reconstructed Unit option.
06 4E	Multilink Maximum Receive Reconstructed Unit option data.
13	LCP option for the Multilink Endpoint Discriminator option.
09	Length in bytes of the Multilink Endpoint Discriminator option.
03 00 60 08 52 F9 D8	Multilink Endpoint Discriminator option data.

As you can see, Network Monitor is the easier tool for the interpretation of PPP traffic. However, the PPP trace contains valuable internal component interaction information that can be useful to troubleshoot connection problems and behavior. When in doubt, obtain both a Network Monitor capture and a PPP trace.

Note PPP tracing in Windows 2000 is the same as the PPP log feature found in Windows NT 4.0 and earlier.

PPP Connection Termination

PPP can terminate the link at any time. Termination generally occurs due to carrier loss, authentication failure, link quality failure, time-out, or link closure by the dial-up client or system administrator. When the link is closing, PPP informs the network layer protocols so that they can take appropriate action.

PPP Authentication Protocols

Phase 2 of the PPP connection establishment process is the authentication of the remote access client. Authentication for PPP is accomplished through a PPP authentication protocol. During Phase 1, both PPP peers agree on a single, specific PPP authentication protocol.

Windows 2000 remote access supports Extensible Authentication Protocol (EAP), Challenge Handshake Authentication Protocol (CHAP), Microsoft Challenge Handshake Authentication Protocol (MS-CHAP) version 1 and version 2, Shiva Password Authentication Protocol (SPAP), and Password Authentication Protocol (PAP).

A secure authentication scheme provides protection against replay attacks, remote access client impersonation, and remote access server impersonation.

- A replay attack occurs when a person captures the packets of a successful connection attempt and then replays those packets in an attempt to obtain an authenticated connection.
- Remote access client impersonation occurs when a person takes over an existing authenticated connection. The intruder waits until

the connection is authenticated and then obtains the connection parameters, disconnects the user, and takes control of the authenticated connection.

- Remote server impersonation occurs when a computer appears as the remote access server to the remote access client. The impersonator appears to verify the remote access client credentials and then captures all of the traffic from the remote access client.

PAP

Password Authentication Protocol (PAP) is a simple, plaintext authentication scheme. The user name and password are requested by the remote access server and returned by the remote access client in plaintext. PAP, however, is not a secure authentication protocol. A person capturing the PAP packets between the remote access server and remote access client can easily determine the remote access client's password. PAP offers no protection against replay attacks, remote client impersonation, or remote server impersonation.

The use of PAP is negotiated during LCP negotiation by specifying the authentication protocol LCP option (type 3) and the authentication protocol 0xC0-23. Once LCP negotiation is complete, PAP messages use the PPP protocol ID of 0xC0-23.

PAP is a simple exchange of messages:

1. The remote access client sends a PAP Authenticate-Request message to the remote access server containing the remote access client's user name and clear text.
2. The remote access server checks the user name and password and sends back either a PAP Authenticate-Ack message when the user's credentials are correct, or a PAP Authenticate-Nak message when the user's credentials are not correct.

PAP is included in Windows 2000 so that remote access clients running Windows 32-bit operating systems can connect to older remote access servers that do not support a secure authentication protocol, and remote access clients not running Microsoft operating systems that do not support a secure remote access protocol can connect to a remote access server running Windows 32-bit operating systems.

Note To make your remote access server more secure, ensure that PAP is disabled. However, older remote access clients not running Microsoft operating systems that do not support secure authentication protocols are unable to connect.

SPAP

The Shiva Password Authentication Protocol (SPAP) is a reversible encryption mechanism employed by Shiva remote access servers. A Windows 2000 remote access client can use SPAP to authenticate itself to a Shiva remote access server. A remote access client running Windows 32-bit operating systems can use SPAP to authenticate itself to a Windows 2000 remote access server. SPAP is more secure than PAP but less secure than CHAP or MS-CHAP. SPAP offers no protection against remote server impersonation.

The use of SPAP is negotiated during LCP negotiation by specifying the authentication protocol LCP option (type 3) and the authentication protocol 0xC0-27. Once LCP negotiation is complete, SPAP messages use the PPP protocol ID of 0xC0-27.

Like PAP, SPAP is a simple exchange of messages:

1. The remote access client sends an SPAP Authenticate-Request message to the remote access server containing the remote access client's user name and encrypted password.
2. The remote access server decrypts the password, checks the user name and password, and sends back either an SPAP Authenticate-Ack message when the user's credentials are correct, or an SPAP Authenticate-Nak message with a reason why the user's credentials were not correct.

CHAP

The Challenge Handshake Authentication Protocol (CHAP) is a challenge-response authentication protocol documented in RFC 1994 that uses the industry-standard Message Digest 5 (MD5) one-way encryption scheme to hash the response to a challenge issued by the remote access server.

CHAP is used by various vendors of dial-in servers and clients. CHAP is supported by both the Windows 2000 remote access server and remote access client.

CHAP is an improvement over PAP and SPAP in that the password is never sent over the link. Instead, the password is used to create a one-way hash from a challenge string. The server, knowing the client's password, can duplicate the operation and compare the result with that sent in the client's response.

The use of CHAP is negotiated during Phase 1 by specifying the authentication protocol LCP option (type 3), the authentication protocol 0xC2-23, and the algorithm 0x05. Once LCP negotiation is complete, CHAP messages use the PPP Protocol ID of 0xC2-23.

CHAP authentication is an exchange of three messages:

1. The remote access server sends a CHAP Challenge message containing a session ID and an arbitrary challenge string.
2. The remote access client returns a CHAP Response message containing the user name in cleartext and a hash of the challenge string, session ID, and the client's password using the MD5 one-way hashing algorithm.
3. The remote access server duplicates the hash and compares it to the hash in the CHAP Response. If the hashes are the same, the remote access server sends back a CHAP Success message. If the hashes are different, a CHAP Failure message is sent.

CHAP protects against replay attacks by using an arbitrary challenge string per authentication attempt. However, CHAP does not protect against remote server impersonation.

CHAP requires that local or domain passwords be stored in a reversibly encrypted form. For more information, see Windows 2000 Server Help.

MS-CHAP v1

The Microsoft Challenge Handshake Authentication Protocol version 1 (MS-CHAP v1) is an encrypted authentication mechanism very similar to CHAP. As in CHAP, the remote access server sends a challenge to the remote client that consists of a session ID and an arbitrary challenge string. The remote client must return the user name and a Message Digest 4 (MD4) hash of the challenge string, the session ID, and the MD4-hashed password.

One difference between CHAP and MS-CHAP v1 is that, in CHAP, the plaintext version of the password must be available to validate the challenge response. With MS-CHAP v1, the remote access server only requires the MD4 hash of the password to validate the challenge response. In Windows 2000, the user's password is stored as an MD4 hash and in a reversibly encrypted form. When CHAP is used, the remote access server decrypts the reversibly encrypted password to validate the remote access client's response.

MS-CHAP v1 authentication is an exchange of three messages:

1. The remote access server sends an MS-CHAP Challenge message containing a session ID and an arbitrary challenge string.
2. The remote access client returns an MS-CHAP Response message containing the user name in cleartext and a hash of the challenge string, session ID, and the MD4 hash of the client's password using the MD4 one-way hashing algorithm.
3. The remote access server duplicates the hash and compares it to the hash in the MS-CHAP Response. If the hashes are the same, the remote access server sends back an MS-CHAP Success message. If the hashes are different, an MS-CHAP Failure message is sent.

The use of MS-CHAP v1 is negotiated during LCP negotiation by specifying the authentication protocol LCP option (type 3), the authentication protocol 0xC2-23, and the algorithm 0x80. Once LCP negotiation is complete, MS-CHAP v1 messages use the PPP protocol ID of 0xC2-23.

MS-CHAP v1 also allows for error codes including a "password expired" code and password changes. MS-CHAP v1 protects against replay attacks by using an arbitrary challenge string per authentication attempt. MS-CHAP v1 does not provide protection against remote server impersonation.

If MS-CHAP v1 is used as the authentication protocol and MPPE is negotiated, then shared secret encryption keys are generated by each PPP peer. MS-CHAP v1 also provides a set of messages that allows a user to change their password during the user authentication process.

MS-CHAP v2

Windows 2000 includes support for Microsoft Challenge Handshake Authentication Protocol version 2 (MS-CHAP v2) that provides stronger security for remote access connections. MS-CHAP v2 offers the additional security features:

- LAN Manager encoding of responses and password changes is no longer supported.
- Two-way authentication verifies the identity of both sides of the connection. The remote access client authenticates against the remote access server and the remote access server authenticates against the remote access client. Two-way authentication, also known as mutual authentication, ensures that the remote access client is dialing into a remote access server that has access to the user's password. Mutual authentication provides protection against remote server impersonation.
- Separate cryptographic keys are generated for transmitted and received data.
- The cryptographic keys are based on the user's password and the arbitrary challenge string. Each time the user connects with the same password, a different cryptographic key is used.

The use of MS-CHAP v2 is negotiated during LCP negotiation by specifying the authentication protocol LCP option (type 3), the authentication protocol 0xC2-23, and the algorithm 0x81. Once LCP negotiation is complete, MS-CHAP messages use the PPP protocol ID of 0xC2-23.

MS-CHAP v2 authentication is an exchange of three messages:

1. The remote access server sends an MS-CHAP v2 Challenge message to the remote access client that consists of a session identifier and an arbitrary challenge string.
2. The remote access client sends an MS-CHAP v2 Response message that contains:
   - The user name.
   - An arbitrary peer challenge string.
   - An Secure Hash Algorithim (SHA) hash of the received challenge string, the peer challenge string, the session identifier, and the MD4-hashed version of the user's password.
3. The remote access server checks the MS-CHAP v2 Response message from the client and sends back an MS-CHAP v2 Response message containing:
   - An indication of the success or failure of the connection attempt.
   - An authenticated response based on the sent challenge string, the peer challenge string, the client's encrypted response, and the user's password.
4. The remote access client verifies the authentication response and if it is correct, uses the connection. If the authentication response is not correct, the remote access client terminates the connection.

EAP

The Extensible Authentication Protocol (EAP) is an extension to PPP that allows for arbitrary authentication mechanisms to be employed for the validation of a PPP connection. With PPP authentication protocols such as MS-CHAP and SPAP, a specific authentication mechanism is chosen during the link establishment phase. Then, during the connection authentication phase, the negotiated authentication protocol is used to validate the connection. The authentication protocol itself is a fixed series of messages sent in a specific order.

With EAP, the specific authentication mechanism is not chosen during the link establishment phase. Instead, each PPP peer negotiates to perform EAP during the connection authentication phase. Once the connection authentication phase is reached, the PPP peers must first negotiate the use of a specific EAP authentication scheme known as an EAP type. Once the EAP type is agreed upon, EAP allows for an open-ended conversation between the remote access client and the remote access server that can vary based on the parameters of the connection. The conversation consists of requests for authentication information and the responses. The length and detail of the authentication conversation is dependent upon the EAP type.

For example, when EAP is used with security token cards, the remote access server could separately query the remote access client for a name, PIN, and card token value. As each query is asked and answered, the user passes through another level of authentication. When all questions have been answered satisfactorily, the user is authenticated and permitted access to the network.

The use of EAP is negotiated during LCP negotiation by specifying the authentication protocol LCP option (type 3) and the authentication protocol 0xC2-27. Once LCP negotiation is complete, EAP messages use the PPP Protocol ID of 0xC2-27. Windows 2000 includes support for the EAP-MD5 and EAP-TLS EAP types.

Architecturally, EAP is designed to allow authentication plug-in modules at both the client and server ends of a connection. By installing an EAP type library file on both the remote access client and the remote access server, a new EAP type can be supported. This presents vendors with the opportunity to supply a new authentication scheme at any time. EAP provides the highest flexibility in authentication uniqueness and variations.

EAP-MD5

EAP-MD5 is the CHAP authentication mechanism used within the EAP framework. Rather than negotiating to perform MD5 authentication during the link establishment phase, the authenticator and peer negotiate to do EAP during the connection authentication phase.

Once the connection authentication phase is reached, the following process verifies the client:

1. The authenticator sends an EAP-Request message requesting the identity of the client.
2. The client sends its user ID to the authenticator as an EAP-Response message.
3. The authenticator sends an EAP-Request message containing the MD5 challenge string.
4. The client sends the MD5 hash of its user ID and password to the authenticator as an EAP-Response message.
5. If the response is proper, the authenticator sends a Success message to the client.

EAP-MD5 is a required EAP type and can be used to test EAP interoperability. Like, CHAP, EAP-MD5 requires that local or domain passwords be stored in a reversibly encrypted form. For more information, see Windows 2000 Server Help.

EAP-TLS

The Transport Layer Security (TLS) protocol, based on the Secure Sockets Layer, allows applications to communicate securely. TLS provides authentication (user and data), data integrity, and data confidentiality services. To achieve these services, TLS specifies a framework that allows the following:

- Client and two-way authentication using symmetric or asymmetric encryption.
- Negotiation of the specific encryption algorithm (the cipher-suite).
- Secured exchange of encryption keys to be used for encrypting messages.
- Message integrity and user authentication using a message authentication code.

For more information about the details of TLS, see RFC 2246. For more information about EAP-TLS, see RFC 2716.

EAP-TLS is the use of TLS during the establishment of a PPP connection. With EAP-TLS, mutual authentication between the PPP client and the authenticator is done through the exchange and verification of certificates. The client attempting the connection sends a user certificate, and the authenticator sends a machine certificate.

EAP-TLS is only supported on Windows 2000 Server remote access server computers that are a member of a Windows 2000 mixed or native domain. Stand-alone Windows 2000 remote access servers do not support EAP-TLS.

EAP-RADIUS

EAP-RADIUS is not an EAP type, but the passing of EAP messages of any EAP type by the remote access server to a RADIUS server for authentication. The EAP messages sent between the remote access client and remote access server are encapsulated and formatted as RADIUS messages between the remote access server and the RADIUS server. The remote access server becomes a pass-through device passing EAP messages between the remote access client and the RADIUS server. All processing of EAP messages occurs at the remote access client and the RADIUS server.

EAP-RADIUS is used in environments where RADIUS is used as the authentication provider. An advantage of using EAP-RADIUS is that EAP types do not need to be installed at each remote access server, only at the RADIUS server.

In a typical use of EAP-RADIUS, the remote access server is configured to use EAP and to use RADIUS as its authentication provider. When a connection attempt is made, the remote access client negotiates the use of EAP with the remote access server. When the client sends an EAP message to the remote access server, the remote access server encapsulates the EAP message as a RADIUS message and sends it to its configured RADIUS server. The RADIUS server processes the EAP message and sends a RADIUS-encapsulated EAP message back to the remote access server. The remote access server then forwards the EAP message to the remote access client.

Unauthenticated Connections

Windows 2000 also supports unauthenticated PPP connections. In an unauthenticated PPP connection, the authentication phase of the PPP connection establishment is skipped. Neither the remote access client or the remote access server exchange credentials. The use of unauthenticated PPP connections must be carefully considered, as connections are allowed without verifying the identity of the remote access client.

There are two common cases where unauthenticated connections are desired:

1. When using Automatic Number Identification/Calling Line Identification (ANI/CLI) authentication, the authentication of a connection attempt is based on the phone number of the caller. ANI/CLI service returns the number of the caller to the receiver of the call and is provided by most standard telephone companies.

   ANI/CLI authentication is different from caller ID authorization. In caller ID authorization, the caller sends a valid user name and password. The caller ID that is configured for the dial-in property on the user account must match the connection attempt; otherwise, the connection attempt is rejected. In ANI/CLI authentication, a user name and password are not sent.
2. When using guest authentication, the Guest account is used as the identity of the caller.

For information about procedures to implement these common unauthenticated connection scenarios, see Windows 2000 Server Help.

Remote Access and TCP/IP and IPX

The following sections describe how the Windows 2000 remote access server allocates network configuration parameters for TCP/IP and IPX-based remote access clients.

TCP/IP

To configure a TCP/IP-based remote access client with IPCP, the remote access server allocates an IP address and assigns the IP addresses of DNS and WINS servers.

IP Address Allocation

To allocate an IP address to a remote access client, the remote access server is either configured to use Dynamic Host Configuration Protocol (DHCP) to obtain IP addresses, or with a static IP address pool.

DHCP and Automatic Private IP Addressing

When the remote access server is configured to use DHCP to obtain IP addresses, the Routing and Remote Access service instructs the DHCP client component to obtain 10 IP addresses from a DHCP server. The remote access server uses the first IP address obtained from DHCP for the RAS server interface, and subsequent addresses are allocated to TCP/IP-based remote access clients as they connect. IP addresses freed due to remote access clients disconnecting are reused.

When all 10 IP addresses are used, the remote access server uses the DHCP client component to obtain 10 more. You can modify the number of IP addresses obtained at a time by changing the value of the **InitialAddressPoolSize** registry entry:

HKEY_LOCAL_MACHINE \System \CurrentControlSet \Services \RemoteAccess \Parameters\Ip

With the Windows NT 4.0 remote access server, the DHCP allocated addresses are recorded and reused when the remote access service is restarted. The Windows 2000 remote access server now releases all DHCP allocated IP addresses using DHCPRELEASE messages each time the service is stopped.

If the remote access server initially starts using DHCP-allocated addresses and the DHCP server becomes unavailable, then an IP address cannot be allocated to additional TCP/IP-based remote access clients.

If a DHCP server is not available when the Routing and Remote Access service is started, then the DHCP client returns 10 addresses in the range 169.254.0.1 to 169.254.255.254 to allocate to remote access clients. The address range 169.254.0.0/16 is used for Automatic Private IP Addressing (APIPA). APIPA addresses for point-to-LAN remote access connectivity work only if the network to which the remote access server computer is attached is also using APIPA addresses. If the local network is not using APIPA addresses, remote access clients are only able to obtain point-to-point remote access connectivity.

If a DHCP server does become available, the next time IP addresses are needed by the Routing and Remote Access service, DHCP-obtained addresses are then allocated to remote access clients that connect after the DHCP addresses were obtained.

The remote access server uses a specific LAN interface to obtain DHCP-allocated IP addresses for remote access clients. You can select which LAN interface to use from the **IP** tab on the properties of a remote access router in the **Routing and Remote Access** snap-in. By default, **Allow RAS to select adapter** is selected, which means that the Routing and Remote Access service randomly picks a LAN interface to use.

Static IP Address Pool

When a static IP address pool is configured, the remote access server uses the first IP address in the first address range for the RAS server interface, and subsequent addresses are allocated to TCP/IP-based remote access clients as they connect. IP addresses freed due to remote access clients disconnecting are reused.

If an address range in the static IP address pool is for off-subnet addresses, either enable an appropriate routing protocol on the remote access server or add the routes corresponding to the IP address ranges to the routers of your intranet. For more information, see "TCP/IP On-Subnet and Off-Subnet Addressing" earlier in this chapter.

DNS and WINS Address Assignment

As part of the IPCP negotiation, the remote access server assigns the IP addresses of DNS and WINS servers. Exactly which set of DNS and WINS server IP addresses are assigned to the remote access client depends on the following factors:

- Whether DNS and WINS server IP address assignment is prohibited.
- Whether DNS and WINS server IP addresses for remote access clients are globally configured by using data stored in the registry.
- Whether the remote access server has more than one LAN interface.
- Whether the DNS and WINS server IP addresses for the remote access server are configured statically or are obtained through DHCP.

Prohibiting DNS and WINS IP Address Assignment

If you do not want the remote access server to assign DNS and WINS IP addresses, set the values of **SuppressDNSNameServers** and **SuppressWINSNameServers** in:

HKEY_LOCAL_MACHINE \System \CurrentControlSet \Services \RemoteAccess \Parameters\Ip to 1

Configuring Global DNS and WINS IP Address Assignment

To globally configure DNS and WINS server IP addresses for remote access clients, enter the IP addresses in the values of **DNSNameServers** and **WINSNameServer** in:

HKEY_LOCAL_MACHINE \System \CurrentControlSet \Services \RemoteAccess \Parameters\Ip

Multiple LAN Interfaces

If the DNS and WINS server IP address assignment is not prohibited or globally configured, then the remote access server allocates the DNS and WINS server IP addresses of a LAN interface on the remote access server to remote access clients. If there is only one LAN interface, which is the typical configuration of a dial-up remote access server, the remote access server allocates the DNS and WINS server IP addresses of the single LAN interface to remote access clients. If there is more than one LAN interface, the DNS and WINS server IP addresses of a specific LAN interface must be determined.

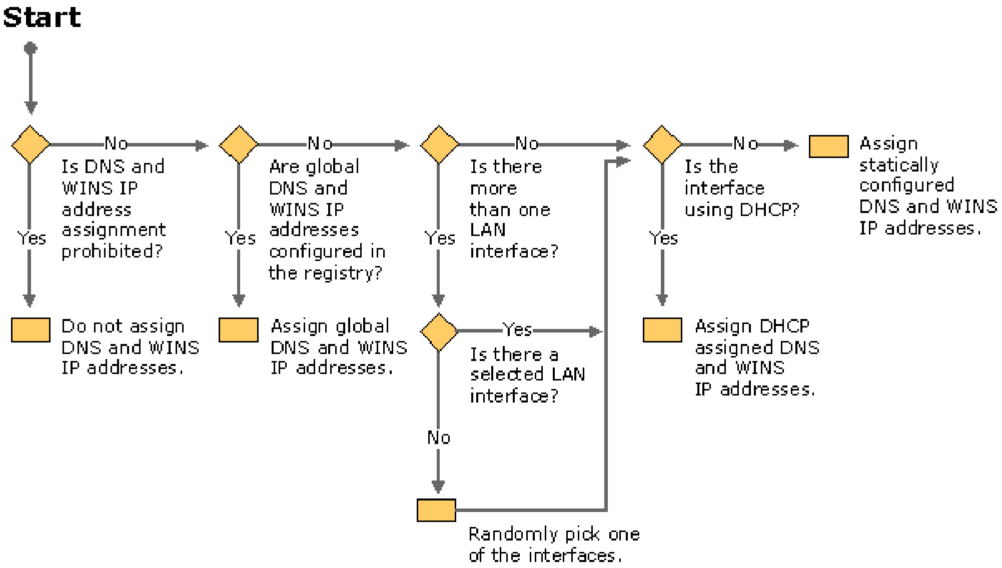
With multiple LAN interfaces, which is the typical configuration of a VPN remote access server, the remote access server by default picks one LAN interface randomly during startup and uses the DNS and WINS server IP addresses of the chosen LAN interface to allocate to remote access clients. To override this behavior, you can select the desired LAN interface through the **IP** tab on the properties of a remote access router in the **Routing and Remote Access** snap-in. By default, **Allow RAS to select adapter** is selected.

Static Configuration or DHCP

Once the LAN adapter for DNS and WINS server IP address assignment has been determined:

- If the LAN adapter has a static IP configuration, then the IP addresses of the statically configured DNS and WINS servers are allocated to remote access clients.
- If the LAN adapter obtained its IP configuration using DHCP, then the IP addresses of the DHCP-obtained DNS and WINS servers are allocated to remote access clients.

The way that the remote access server determines the set of DNS and WINS server IP addresses to assign to remote access clients during IPCP negotiation is illustrated in Figure 7.14.


Start
No
Is DNS and WINS IP address assignment prohibited?
Yes
Do not assign DNS and WINS IP addresses.
No
Are global DNS and WINS IP addresses configured in the registry?
Yes
Assign global DNS and WINS IP addresses.
No
Is there more than one LAN interface?
Yes
Yes
Is there a selected LAN interface?
No
Randomly pick one of the interfaces.
No
Is the interface using DHCP?
Yes
Assign DHCP assigned DNS and WINS IP addresses.
Assign statically configured DNS and WINS IP addresses.

If your browser does not support inline frames, click here to view on a separate page.

Figure 7.14 DNS and WINS Server IP Address Determination

Overriding IPCP-Allocated DNS and WINS Server IP Addresses with DHCPInform

After IPCP is completed, Windows 2000 and Windows 98 remote access clients send their remote access servers a DHCPInform message. DHCPInform is a DHCP message used by DHCP clients to obtain DHCP options. While PPP remote access clients do not use DHCP to obtain IP addresses for the remote access connection, Windows 2000 and Windows 98 remote access clients use the DHCPInform message to obtain DNS server IP addresses, WINS server IP addresses, and a DNS domain name. The DHCPInform message is sent after the IPCP negotiation is concluded.

The DHCPInform message received by the remote access server is then forwarded to a DHCP server. The remote access server forwards DHCPInform messages only if it has been configured with the DHCP Relay Agent as discussed in the following section. The response to the DHCPInform message is forwarded back to the requesting remote access client.

If the DHCPInform response contains DNS and WINS server IP address options, then these new values override what was allocated during IPCP. When the remote access client is a Windows 2000 remote access client and the DHCPInform response contains a DNS domain name, the DNS domain name is used as the per-adapter DNS suffix for the remote access connection of the remote access client. For more information on per-adapter DNS suffixes, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.

Remote Access Server and the DHCP Relay Agent

To facilitate the forwarding of DHCPInform messages between remote access clients and DHCP servers, the remote access server uses the DHCP Relay Agent, a component of the Windows 2000 remote access router. To configure the remote access server to use the DHCP Relay Agent, add the **Internal** interface to the DHCP Relay Agent IP routing protocol with the **Routing and Remote Access** snap-in.

If the remote access server is using DHCP to obtain IP addresses for remote access clients, then the remote access server uses the DHCP Relay Agent to forward DHCPInform messages to the DHCP server of the selected LAN interface, on the **IP** tab on the properties of a remote access router in the **Routing and Remote Access** snap-in.

If the remote access server is using a static IP address pool to obtain IP addresses for remote access clients, then the DHCP Relay Agent must be configured with the IP address of at least one DHCP server. Otherwise, DHCPInform messages sent by remote access clients are silently discarded by the remote access server.

IPX

To configure an IPX-based remote access client with IPXCP, the remote access server allocates an IPX network number and an IPX node number. The network number allocation behavior is configured from the **IPX** tab on the properties of a remote access router in the **Routing and Remote Access** snap-in. The main capabilities of IPX configuration are:

- IPX network numbers for remote access clients can be automatically allocated or specified as a range by the network administrator. For automatically allocated IPX network numbers, the remote access server ensures that the IPX network number is not being used on the IPX internetwork by sending a RIP GetLocalTarget packet on its LAN interfaces. If there is a response to the RIP GetLocalTarget, then the IPX network number is in use and another IPX network number is chosen.
- The same IPX network number can be assigned to all remote access clients.
- Specific IPX node numbers can be requested by remote access clients.

You can set the first IPX node number as a 12-digit hexadecimal number to be allocated to IPX remote access clients.To do so, add the **FirstWanNode** (REG_SZ) registry entry to:

HKEY_LOCAL_MACHINE \System \CurrentControlSet \Services \RemoteAccess \Parameters\Ipx

And enter the desired IPX node number in the value of the entry.

Subsequent IPX clients are assigned incrementally increasing node numbers. If this registry entry does not appear in the registry, a random IPX node number, in the form 0x2E-*xx-xx-xx-xx-xx* (where you specify each *x* digit), is assigned when the remote access client does not request a specific IPX node number.

Remote Access Policies

In Windows 2000, remote access connections are accepted based on the dial-in properties of a user account and remote access policies. A remote access policy is a set of conditions and connection parameters that define the characteristics of the incoming connection and the set of constraints imposed on it. Remote access policies can be used to specify allowed connections conditioned by the time of day and day of the week, the Windows 2000 group to which the dial-in user belongs, the type of remote access client (dial-up or VPN), and so on. Remote access policies can be used to impose connection parameters such as maximum session time, idle disconnect time, required secure authentication methods, required encryption, and so on.

With multiple remote access policies, different sets of conditions can be applied to different remote access clients or different requirements can be applied to the same remote access client based on the parameters of the connection attempt. For example, multiple remote access policies can be used to:

- Allow or deny connections if the user account belongs to a specific group.
- Define different days and times for different user accounts based on group membership.
- Configure different authentication methods for dial-up and VPN remote access clients.
- Configure different authentication or encryption settings for Point-to-Point Tunneling Protocol (PPTP) or Layer Two Tunneling Protocol (L2TP) connections.
- Configure different maximum session times for different user accounts based on group membership.
- Send network access server-specific RADIUS attributes to a RADIUS client.

When you have multiple Windows 2000 remote access or VPN server and you want all of the servers to use a centralized set of remote access policies to authorize incoming connections, you must configure a computer to run Windows 2000 and Internet Authentication Service (IAS) and then configure each remote access or VPN server as a RADIUS client to the IAS server computer.

For more information about remote access policies, including common remote access policy scenarios and their configuration, see Windows 2000 Server Help.

Connection Attempt Processing

To process a connection attempt, the parameters of the connection attempt are compared to the user name, password, and dial-in properties of the user account and the configured remote access policies.

Some general characteristics of remote access connection attempt processing are:

- If a connection attempt does not use a valid user name and password, then the connection attempt is denied.
- If there are no configured policies, then all connection attempts are denied.
- If the connection attempt does not match any of the remote access policies, then the connection attempt is denied.

- If the remote access permission of the user account for the remote access user is set to **Deny Access**, the connection attempt is always denied for that remote access user.
- The only time that a connection attempt is allowed is when it matches the conditions of a remote access policy, and remote access permission is enabled either through the dial-in properties of the user account or through the remote access permission of the remote access policy (assuming the user's remote access permission is set to control access through remote access policies), and the parameters of the connection attempt match or conform to the parameters and conditions of the dial-in properties of the user account and the remote access policy profile properties.

Figure 7.15 depicts the specific processing of remote access connection attempts using the dial-in properties of the user account and remote access policies. Figure 7.15 assumes that the user name and password sent during the authentication process match a valid user account.

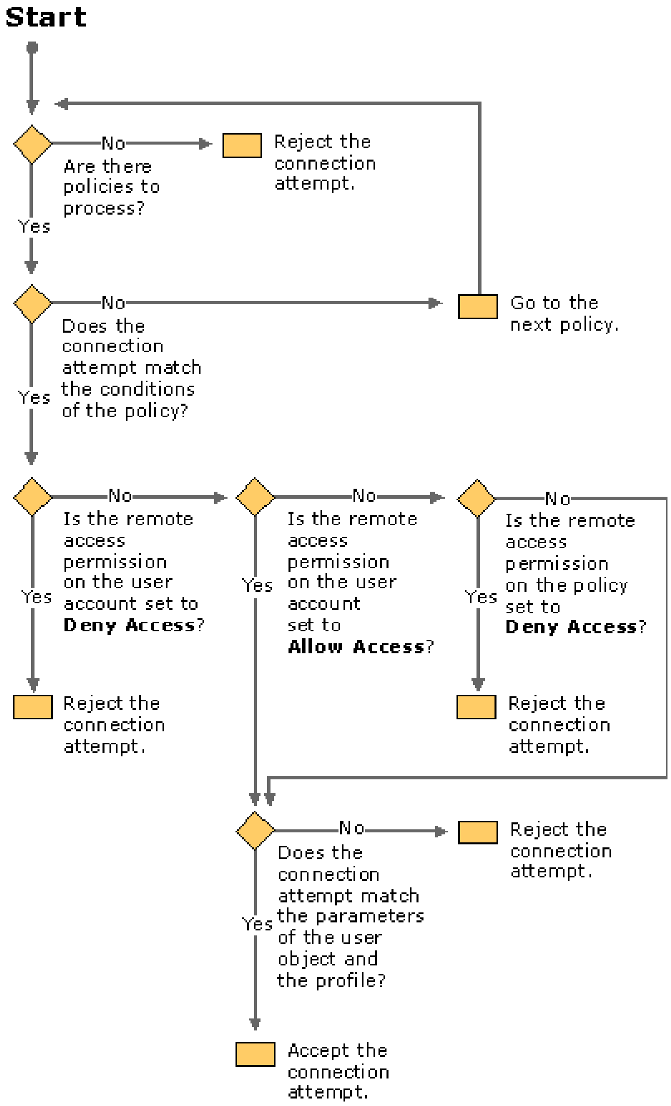

Start
No
Are there policies to process?
Yes
Reject the connection attempt.
No
Does the connection attempt match the conditions of the policy?
Yes
Go to the next policy.
No
Is the remote access permission on the user account set to **Deny Access**?
Yes
Reject the connection attempt.
No
Is the remote access permission on the user account set to **Allow Access**?
Yes
No
Is the remote access permission on the policy set to **Deny Access**?
Yes
Reject the connection attempt.
No
Does the connection attempt match the parameters of the user object and the profile?
Yes
Reject the connection attempt.
Accept the connection attempt.

Figure 7.15 Connection Attempt Processing

Troubleshooting Remote Access Policies

A common problem with remote access policies is that a connection attempt is denied when it should be allowed. When in doubt, apply the logic of Figure 7.7 to the parameters of the connection attempt, the dial-in properties of the user account, and the remote access policies. However, troubleshooting the denial of the connection attempt can be very time consuming when there are multiple remote access policies in place.

When multiple remote access policies are configured and you want to determine which remote access policy is denying the connection attempt, then enable the logging of authentication requests for local files from **Remote Access Logging** in the **Routing and Remote Access** snap-in. Logged authentication requests contain the name of the remote access policy used in either accepting or rejecting the connection attempt.

Multilink and Bandwidth Allocation Protocol

Windows 2000 remote access supports the PPP Multilink Protocol (MP), the Bandwidth Allocation Protocol (BAP), and the Bandwidth Allocation Control Protocol (BACP):

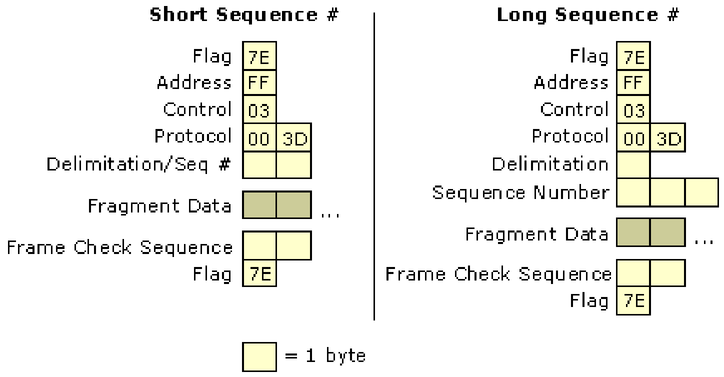
- MP allows multiple physical links to appear as a single logical link over which data can be sent and received.
- BAP is a PPP control protocol that is used to dynamically add or remove additional links to an MP connection.
- BACP is a PPP NCP that elects a favored peer in case both PPP peers request to add or remove a connection at the same time.

Each of these protocols is discussed in greater detail in the following sections.

PPP Multilink Protocol

The PPP Multilink Protocol (MP) protocol is defined in RFC 1990 and used to aggregate multiple physical links into a single logical link. A good example is the aggregation of both B-channels of an ISDN Basic Rate Interface (BRI) connection. MP fragments, sequences, and re-orders alternating packets sent across multiple physical connections so that the end result is a single logical link with the combined bandwidth of all of the aggregated physical links. MP is the recommended method of combining multiple B-channels of a BRI connection because the support for bonding - the combining of ISDN B-channels through hardware support - can be specific to the ISDN adapter. MP can be done for any ISDN adapter. MP must be supported on both sides of the connection.

Figure 7.16 illustrates the structure of an MP frame. The payload of a Multilink PPP packet is either a fragment of a PPP frame or the entire PPP frame. Multilink PPP fragmentation need not occur if the Multilink PPP packet fits within the MRU of the link. To prevent improper ordering of the datagrams or fragments across multiple links, additional fields are used between the PPP Protocol field and the IP datagram. Multilink PPP uses the PPP Protocol ID of 0x00-3D.



If your browser does not support inline frames, click here to view on a separate page.

Figure 7.16 Multilink PPP

RFC 1717 defines two different packet formats for short sequence numbers and long sequence numbers. When using either short or long sequence numbers, the sequence number is used to prevent misordering of frames that are sent across multiple links, not to sequence fragments.

For short sequence numbers, a two byte Delimitation/Sequence # field consists of four bits used for delimitation and 12 bits used for the sequence number. Within the delimitation field are two flags. The first bit (the Beginning bit) is an indicator that this fragment begins a sequence of fragments corresponding to a packet. The second bit (the Ending bit) is an indicator that this fragment ends a sequence of fragments corresponding to a packet. The other bits in the first four bits of the short sequence number header are set to 0.

For a PPP frame that is sent without fragmentation, both the Beginning and Ending bits are set. For a PPP frame that is larger than the MRU of the physical link, the PPP frame is fragmented, and each fragment is sent as a separate PPP packet. MP performs a data-link layer fragmentation that is not related to IP fragmentation.

For long sequence numbers, a four byte Delimitation/Sequence # field consists of eight bits (one byte) used for delimitation, and 24 bits (3 bytes) used for the sequence number. Within the delimitation field, the same bits as the short sequence number header define the Beginning bit and the Ending bit. The other bits in the first byte of the long sequence number header are set to 0. The long sequence number header is used by default, unless the short sequence number is chosen during LCP negotiation.

Table 7.16 lists Multilink LCP options negotiated by Microsoft PPP peers. For information about other Multilink options, see RFC 1990.

Table 7.16 Multilink LCP Options

Option Name	Option Type	Option Length	Description
Multilink Maximum Receive Reconstructed Unit	17 or 0x11	4	Specifies the number of octets that a peer can reconstruct when performing reassembly of fragmented MP frames.
Short Sequence Number Header Format	18 or 0x12	2	Specifies the use of the short sequence number in the MP header.
Multilink Endpoint Discriminator	19 or 0x13	9	A unique system identifier to differentiate links from two PPP peers with the same authenticated name.

Bandwidth Allocation Protocol (BAP)

While MP allows for multiple physical links to be aggregated, MP does not provide a mechanism to adapt to changing conditions by adding extra links when needed or terminating extra links when unneeded. This additional capability is provided by the Bandwidth Allocation Protocol (BAP) and the Bandwidth Allocation Control Protocol (BACP) defined in RFC 2125. BAP is a PPP control protocol that is used on an MP connection to dynamically manage links. BAP uses the PPP Protocol of ID 0xC0-2D.

For example, an MP and BAP-enabled remote access client and remote access server create an MP connection consisting of a single physical link. As the utilization of the single link rises to a configured level, the remote access client uses a BAP Call-Request message to request an additional link. The BAP Call-Request message specifies the type of link desired, such as analog phone, ISDN, or X.25. The remote access server then sends a BAP Call-Response message containing the phone number of an available port on the remote access server of the same type specified by the remote access client in the BAP Call-Request.

When the utilization on the second link drops to a specific level, either the remote access client or the remote access server can send a BAP Link-Drop-Query-Request message to drop the link.

BAP also supports a Callback-Request message where the requesting peer specifies the link type and the number to call back to. For more information about BAP messages, see RFC 2125.

Table 7.17 lists BAP LCP options negotiated by Microsoft PPP peers.

Table 7.17 BAP LCP Options

Option Name	Option Type	Option Length	Description
BAP Link Discriminator	23 or 0x17	4	A unique number used to identify a particular link in a Multilink PPP connection.

Multilink PPP and BAP are enabled on the remote access server through the **PPP** tab on the properties of the remote access router in the **Routing and Remote Access** snap-in. Properties of Multilink and BAP are configured from the **Multilink** tab on the properties of a remote access policy profile.

To set the phone number of a port that is sent in the BAP Call-Response message

1. Use the **Routing and Remote Access** snap-in to obtain properties on the **Ports** object.
2. Select the desired port, and click **Configure**.
3. Type the phone number in the **Phone number of this device** text box.

Bandwidth Allocation Control Protocol (BACP)

The Bandwidth Allocation Control Protocol (BACP) is a PPP NCP that negotiates a single option: the election of a favored peer. If both peers of an MP and BAP-enabled connection send BAP Call-Request or BAP Link-Drop-Query-Request messages at the same time, the favored peer is the peer whose requests are implemented.

BACP uses the PPP Protocol ID of 0xC0-2B. The packet structure of BACP is exactly the same for LCP, except that only packet types 1 through 7 are defined. For Configure-Request, Configure-Ack, Configure-Nack, and Configure-Reject BACP packet types, the BACP data portion of the BACP packet consists of the single BACP Favored-Peer option listed in Table 7.18.

Table 7.18 BACP Favored Peer Option

Option Name	Option Type	Option Length	Description
Favored-Peer	1	6	A randomly allocated 4-byte magic number used to elect a favored BAP peer. The favored peer is the peer with the lowest magic number.

Remote Access Server and IP Multicast Support

The Windows 2000 remote access server also supports the forwarding of IP multicast traffic between remote access clients and the networks to which the remote access server is attached.

IP multicast support for remote access clients requires the following three elements, as illustrated in Figure 7.17.

1. Internet Group Management Protocol (IGMP) router mode is enabled on the interface connected to all of the remote access clients. In the **Routing and Remote Access** snap-in, this is the **Internal** interface.
2. IGMP proxy mode is enabled on a single interface.
3. The network corresponding to the interface on which IGMP proxy mode is enabled is part of an IP multicast-enabled network. An IP multicast-enabled network uses multicast routing protocols to propagate IP multicast traffic from multicast sources located on any network, to hosts located on any network. For example, the IP multicast-enabled portion of the Internet is called the Multicast Backbone or *MBone*.

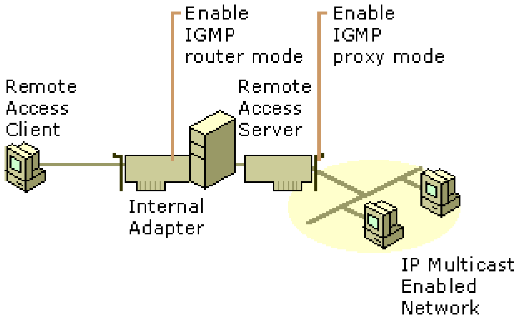


Figure 7.17 Multicast Support for Remote Access

For more information about IP multicasting and its support in Windows 2000 Server, see "IP Multicast Support" in this book.

Note Depending on your choices when running the Routing and Remote Access Server Setup Wizard, IGMP router mode and IGMP proxy mode may already be enabled on the appropriate interfaces.

Multicast Traffic to Remote Access Clients

To facilitate the forwarding of IP multicast traffic from multicast sources on the IP multicast-enabled network to remote access clients:

1. The multicast groups being listened to by remote access clients must be registered with the IP multicast routers of the IP multicast-enabled network.
2. IP multicast traffic from multicast sources must be forwarded to the remote access clients.

Remote Access Client Group Registration

Remote access clients register the IP multicast addresses from which they want to receive traffic by sending IGMP Membership Report messages across the remote access connection. The remote access server records the multicast groups registered by each remote access client and then forwards the IGMP Membership Report message using the interface on which IGMP proxy mode is enabled.

The forwarded IGMP Membership Report message is received by the IP multicast routers attached to the remote access server's network segment. The IP multicast routers of the IP multicast-enabled network use multicast routing protocols to create entries in their multicast forwarding tables, so that multicast traffic sent to the multicast groups - registered by the remote access clients - are forwarded to the network segment of the remote access server.

Forwarding Multicast Traffic

When the multicast source sends multicast traffic to the multicast group registered by the remote access clients, IP multicast routers forward the multicast traffic to the network segment of the remote access server for the interface on which IGMP proxy mode is enabled.

When the remote access server receives multicast traffic on the interface on which IGMP proxy mode is enabled, the multicast traffic is checked to determine whether any connected remote access client has registered an IGMP Membership Report for that multicast group. If the multicast traffic corresponds to a multicast group registered by a remote access client, the multicast traffic is forwarded to the remote access client.

Multicast Traffic from Remote Access Clients

To facilitate the forwarding of IP multicast traffic from remote access clients that are multicast sources:

1. The multicast groups being listened to by hosts must be registered with the IP multicast routers of the IP multicast-enabled network.
2. IP multicast traffic from the remote access clients must be forwarded to the group members.

Host Group Registration

Hosts on the IP multicast-enabled network register the IP multicast addresses for which they want to receive traffic from by sending IGMP Membership Report messages on their local network segments. The IP multicast routers of the IP multicast-enabled network use multicast routing protocols to create entries in their multicast forwarding tables. Therefore, multicast traffic sent to the multicast groups registered by the hosts are forwarded to the host's network segment.

Forwarding Multicast Traffic

When the remote access client sends multicast traffic across the remote access connection, the multicast traffic is forwarded to the network segment of the interface on the remote access server enabled for IGMP proxy mode. IP multicast routers on that network segment receive the forwarded multicast traffic and forward it to the network segments of the group members.

Additionally, the remote access server forwards the IP multicast traffic to other remote access clients that are listening for the IP multicast traffic of the remote access client that is the multicast source.

Internet-Based IP Multicast Traffic

If the remote access server is being used to provide Internet access to dial-up clients, then the following configuration allows IP multicast traffic to and from connected remote access clients:

1. The remote access server has a direct connection to the Internet's MBone or an indirect connection to the MBone through a logical tunnel.
2. The interface corresponding to the direct or indirect connection to the MBone is added to the IGMP routing protocol and enabled for IGMP proxy mode.
3. The **Internal** interface is added to the IGMP routing protocol and enabled for IGMP router mode.

Organization-Based IP Multicast Traffic

If the remote access server is being used to provide a connection to an organization's intranet to dial-up or VPN clients, then the following configuration allows IP multicast traffic to and from connected remote access clients:

1. The remote access server has a LAN interface on the organization's intranet, which is a network segment on the organization's IP multicast-enabled network.
2. The LAN interface connection to the organization intranet is added to the IGMP routing protocol and enabled for IGMP proxy mode.
3. The **Internal** interface is added to the IGMP routing protocol and enabled for IGMP router mode.

Troubleshooting the Remote Access Server

Troubleshooting remote access is a combination of troubleshooting IP connectivity, addressing, routing, and dial-up hardware. A firm understanding of all of these topics is required. The following sections outline common remote access problems and the troubleshooting tools provided with Windows 2000.

To troubleshoot VPN connections, see "Virtual Private Networking" in this book. To troubleshoot demand-dial routing connections, see "Demand-Dial Routing" in this book.

Common Remote Access Problems

Remote access problems typically include the following:

- Connection attempt is rejected when it should be accepted.
- Connection attempt is accepted when it should be rejected.
- Unable to reach locations beyond the remote access server.
- Miscellaneous remote access problems.

The following sections give troubleshooting tips to isolate the configuration or infrastructure problem causing the remote access problem.

Connection Attempt Is Rejected When It Should Be Accepted

- Verify that the Routing and Remote Access Service is running on the remote access server.
- Verify that remote access is enabled on the remote access server.
- Verify that the dial-up ports on the remote access server are configured to allow inbound remote access connections.
- Verify that the remote access client and the remote access server in conjunction with a remote access policy are enabled to use at least one common authentication method.
- Verify that the remote access client and the remote access server in conjunction with a remote access policy are enabled to use at least one common encryption method.
- Verify that the parameters of the connection attempt are accepted by the currently configured dial-in properties of the user account and remote access policies.

  In order for the connection to be established, the parameters of the connection attempt must:

  1. Match all of the conditions of at least one remote access policy.
  2. Be granted remote access permission, either through the remote access permission of the user account (set to **Allow access**), or the user account is set to **Control access through Remote Access Policy** and the remote access permission of the matching remote access policy is set to **Grant remote access permission**.
  3. Match all the settings of the profile.
  4. Match all the settings of the dial-in properties of the user account.
- Verify that the settings of the remote access policy profile are not in conflict with properties of the remote access router.

  The properties of the remote access policy profile and the properties of the remote access router both contain settings for:

  - Multilink
  - Bandwidth allocation protocol
  - Authentication protocols

  If the settings of the profile of the matching remote access policy are in conflict with the settings of the remote access router, then the connection attempt is denied. For example, if the matching remote access policy profile specifies that the EAP-TLS authentication protocol must be used and EAP-TLS is not enabled through the properties of the remote access router, then the remote access server denies the connection attempt.

- For a remote access server that is a member server in a mixed-mode or native-mode Windows 2000 domain that is configured for Windows 2000 authentication, verify that:
  - The **RAS and IAS Servers** security group exists. If not, then create the group and set the group type to **Security** and the group scope to **Domain local**.
  - The **RAS and IAS Servers** security group has **Read** permission to the **RAS and IAS Servers Access Check** object.
  - The computer account of the remote access server computer is a member of the **RAS and IAS Servers** security group. You can use the **netsh ras show registeredserver** command at the Windows 2000 command prompt to view the current registration. You can use the **netsh ras add registeredserver** command to register the server in a specified domain.
- If you add or remove the remote access server computer to the **RAS and IAS Servers** security group, the change does not take effect immediately (due to the way that Windows 2000 caches Active Directory™ directory service information). For the change to take effect immediately, you need to restart the remote access server computer.
- Verify that your dial-up equipment is working properly.
- Verify that all of the dial-up ports on the remote access server are not already connected.
- Verify that the LAN protocols being used by the remote access clients are either enabled for routing or remote access.
- Verify that the remote access client's credentials consisting of user name, password, and domain name are correct and can be validated by the remote access server.
- For connections using MS-CHAP v1 and attampting to negotiate 40-bit MPPE encryption, verify that the user's password is not larger than 14 characters.
- Verify that the user account has not been disabled or is not locked out on the properties of the user account. If the password on the account has expired, verify that the remote access client is using MS-CHAP v1 or MS-CHAP v2. MS-CHAP v1 and MS-CHAP v2 are the only authentication protocols provided with Windows 2000 that allow you to change an expired password during the connection process.

  For an administrator-level account whose password has expired, reset the password using another administrator-level account.
- Verify that the user account has not been locked out due to remote access account lockout.
- If the remote access server is configured with a static IP address pool, verify that there are enough addresses in the pool.

  If all of the addresses in the static pool have been allocated to connected remote access clients, then the remote access server is unable to allocate an IP address. If the remote access client is only configured to use TCP/IP as a LAN protocol, the connection attempt is denied.
- If the remote access client is configured to request its own IPX node number, verify that the server is configured to allow IPX clients to request their own IPX node number.
- If the remote access server is configured with a range of IPX network numbers, verify that the IPX network numbers in the range are not being used elsewhere on your IPX internetwork.
- Verify the configuration of the authentication provider.

  The remote access server can be configured to use either Windows 2000 or RADIUS to authenticate the credentials of the remote access client.
- For a remote access server that is a member of a Windows 2000 native-mode domain, verify that the remote access server has joined the domain.
- For a Windows NT version 4.0 Service Pack 4 and later remote access server that is a member of a Windows 2000 mixed mode domain or a Windows 2000 remote access server that is a member of a Windows NT 4.0 domain that is accessing user account properties for a user account in a trusted Windows 2000 domain, verify that the Everyone group is added to the Pre-Windows 2000 Compatible Access group with the **net localgroup** "**Pre-Windows 2000 Compatible Access**" command. If not, issue the **net localgroup** "**Pre-Windows 2000 Compatible Access**" **everyone /add** command on a domain controller computer and then restart the domain controller computer.
- For a Windows NT version 4.0 Service Pack 3 and earlier remote access server that is a member of a Windows 2000 mixed mode domain, verify that Everyone group has been granted list contents, read all properties, and read permissions to the root node of your domain and all sub-objects of the root domain.
- For RADIUS authentication, verify that the remote access server computer can communicate with the RADIUS server.
- If you are using MS-CHAP v1, verify that you are not using a user password over 14 characters long. If so, either use a different authentication protocol or change the password so that it is 14 characters or less in length.
- Verify that if the Windows 2000 Fax service and the Routing and Remote Access service are sharing the same modem, that the modem supports adaptive answer. If the modem does not support adaptive answer, you must disable fax on the modem to receive incoming remote access connections.

Connection Attempt Is Accepted When It Should Be Rejected

- Verify that the parameters of the connection does not have permission through remote access policies.

  In order for the connection to be rejected, the parameters of the connection attempt must be denied remote access permission one of two ways. Either set the remote access permission of the user account to **Deny access** or set the user account to **Control access through Remote Access Policy,** and then set the remote access permission of the first remote access policy that matches the parameters of the connection attempt to **Deny remote access permission**.

Unable to Reach Locations Beyond the Remote Access Server

- Verify that the LAN protocols being used by the remote access clients are either enabled for routing or enabled to allow access to the network to which the remote access server is attached.
- Verify the IP address allocation settings of the remote access server.

  If the remote access server is configured to use a static IP address pool, verify that the destinations of the address ranges of the static IP address pool are reachable by the hosts and routers of the intranet. If not, then routes corresponding to the address ranges, as defined by the IP address and mask of the range, must be added to the routers of the intranet or enable the routing protocol of your routed infrastructure on the remote access server. If the routes to the remote access client address ranges are not present, remote access clients cannot receive traffic from locations on the intranet. Routes for the address ranges are implemented either through static routing entries or through a routing protocol, such as Open Shortest Path First (OSPF) or Routing Information Protocol (RIP).

  If the remote access server is configured to use DHCP for IP address allocation and no DHCP server is available, the remote access server allocates addresses from the Automatic Private IP Addressing (APIPA) address range from 169.254.0.1 through 169.254.255.254. Allocating APIPA addresses for remote access clients works only if the network to which the remote access server is attached is also using APIPA addresses.

If the remote access server is using APIPA addresses when a DHCP server is available, verify that the proper adapter is selected from which to obtain DHCP-allocated IP addresses. By default, the remote access server randomly chooses the adapter to use to obtain IP addresses through DHCP. If there is more than one LAN adapter, then the Routing and Remote Access service may choose a LAN adapter for which there is no DHCP server available. You can manually choose a LAN adapter from the **IP** tab on the properties of a remote access server in the **Routing and Remote Access** snap-in.

- If the address ranges of the static IP address pool are a subset of the range of IP addresses for the network to which the remote access server is attached, verify that the address ranges of the static IP address pool are not assigned to other TCP/IP nodes, either through static configuration or through DHCP.
- Verify that packet filters on the remote access policy profile are not preventing the flow of needed IP traffic. TCP/IP packet filters can be configured on the profile properties of the remote access policies on the remote access server (or the RADIUS server if Internet Authentication Service is used) that are used to define traffic that is allowed on the remote access connection.
- If Microsoft remote access clients using only the IPX protocol are unable to create file and print sharing connections to servers that are beyond the remote access server, then NetBIOS over IPX broadcast forwarding must be enabled on the **Internal** interface and the appropriate LAN interfaces. For more information on NetBIOS over IPX broadcasts, see "IPX Routing" in this book.

Callback Problems

- Verify that callback is enabled on the dial-in properties of the user account.
- Verify that **Link Control Protocol (LCP) Extensions** is enabled on the **PPP** tab on the properties of a remote access server in the **Routing and Remote Access** snap-in.
- Verify that the callback numbers are not too long. Callback numbers may be truncated when a remote access server running Windows NT 4.0 requests dial-in properties of a user account in a Windows 2000 native-mode domain.

Troubleshooting Tools

The following tools, which enable you to gather additional information about the source of your problem, are included with Windows 2000.

Authentication and Accounting Logging

A remote access server running Windows 2000 supports the logging of authentication and accounting information for remote access connections in local logging files when Windows authentication or Windows accounting is enabled. This logging is separate from the events recorded in the system event log. You can use the information that is logged to track remote access usage and authentication attempts. Authentication and accounting logging is especially useful for troubleshooting remote access policy issues. For each authentication attempt, the name of the remote access policy that either accepted or rejected the connection attempt is recorded.

The authentication and accounting information is stored in a configurable log file or files stored in the %*SystemRoot*%\System32 \LogFiles folder. The log files are saved in Internet Authentication Service (IAS) 1.0 or database format, meaning that any database program can read the log file directly for analysis.

If the remote access server is configured for RADIUS authentication and accounting and the RADIUS server is a Windows 2000 computer running IAS, then the authentication and accounting logs are stored in the %*SystemRoot*%\System32\LogFiles folder on the IAS server computer.

Event Logging

On the **Event logging** tab on the properties of a remote access server, there are four levels of logging. Select **Log the maximum amount of information** and try the connection again. After the connection fails, check the system event log for events logged during the connection process. After you are done viewing remote access events, select the **Log errors and warnings** option on the **Event logging** tab.

Tracing

Tracing records the sequence of programming functions called during a process to a file. Enable tracing for remote access components and try the connection again. After you are done viewing the traced information, reset the tracing settings back to their default values. You can enable PPP tracing from the **Event logging** tab on the properties of a remote access server.

The tracing information can be complex and very detailed. Most of the time this information is useful only to Microsoft support professionals, or to network administrators who are very experienced with the Routing and Remote Access service. The tracing information can be sent to Microsoft support for analysis.

Network Monitor

Network Monitor is a packet capture and analysis tool that you can use to view the traffic sent between a remote access server and remote access client during the remote access connection process and during data transfer. Network Monitor does not interpret the compressed or encrypted portions of remote access traffic.

The proper interpretation of the remote access traffic with Network Monitor requires an understanding of PPP protocols described in this chapter and the referenced RFCs. Network Monitor captures can be saved as files and sent to Microsoft support for analysis.

Send feedback to Microsoft