Preparing to Configure DDR

This chapter presents the decisions and preparations leading to a dial-on-demand routing (DDR) configuration and shows where some advanced features fit into the DDR configuration steps. It distinguishes between the topology decisions and the implementation of the decisions. In the implementation phase, it distinguishes the DDR-independent decisions from the DDR-dependent decisions.

This chapter provides the following information:

- DDR Decision Flowchart—A flowchart of topology and implementation decisions that you will need to make before you configure DDR.
- DDR Topology Decisions, DDR-Independent Implementation Decisions, and DDR-Dependent Implementation Decisions—References to sources of detailed information for the configuration steps associated with each decision.
- Global and Interface Preparations for DDR—Brief description indicating which preparations are global and which are interface-specific.
- Preparations for Routing or Bridging over DDR—A description of the steps required for bridging or routing over DDR.

The section "Configuration Examples for Legacy DDR" at the end of this chapter provides examples of configuring DDR in your network, and includes line configuration and chat script samples.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

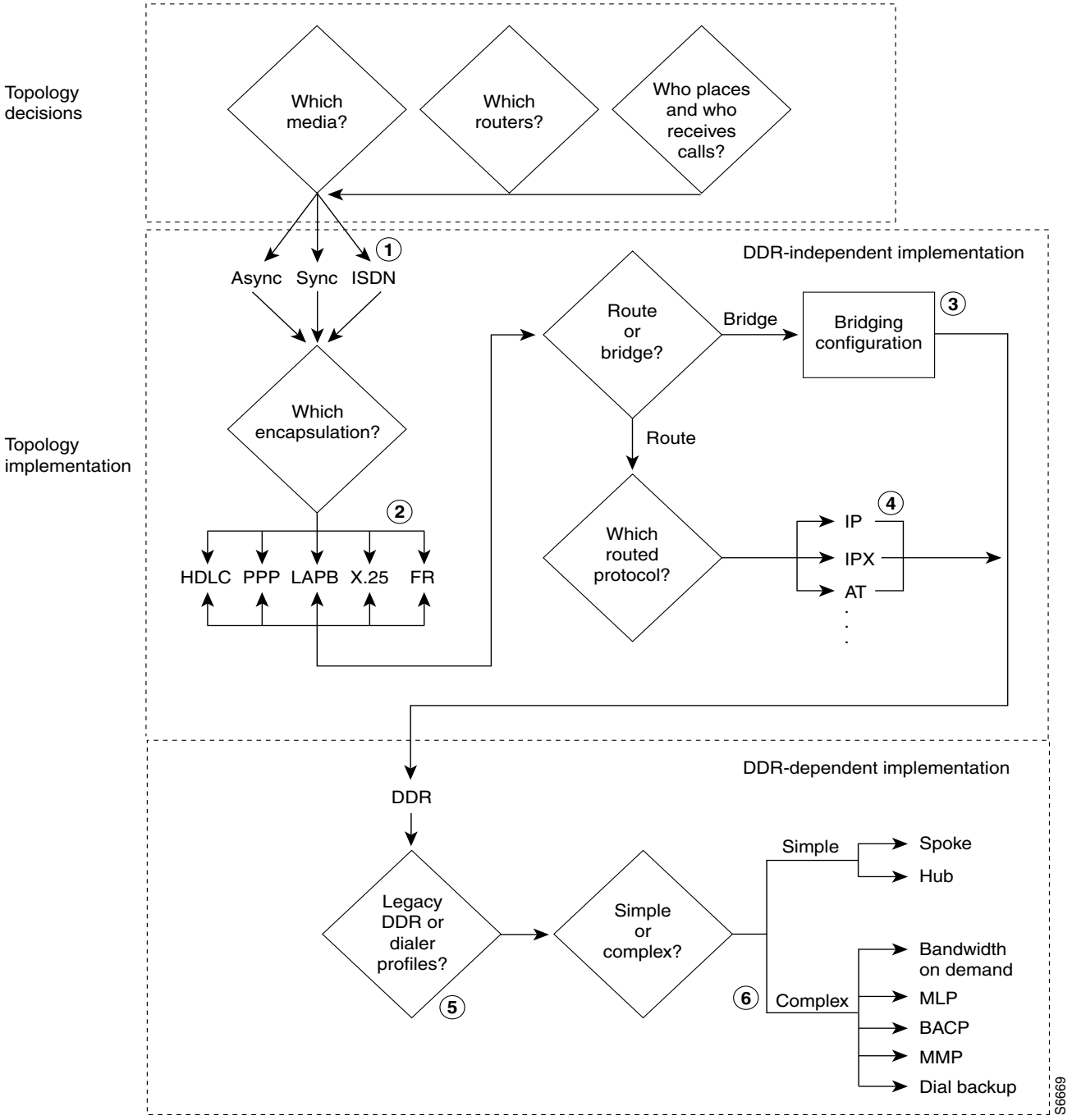
For a complete description of the global dialer commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

DDR Decision Flowchart

This section provides a flowchart of the decisions to be made before and while you configure DDR and also includes the flowchart.

Figure 1 presents the entire decision flowchart. The decision phases are shown in separate boxes. Numbers in parentheses refer to notes, which follow the figure.

Figure 1 Decisions and Implementation Flow to DDR



Flowchart Notes

The DDR chapters do not provide complete configuration information for most of the items in the following list. However, detailed information is available in other chapters and publications. The numbers in this list correspond to the circled numbers in the flowchart.

1. Configuration of the dial port and interface. The port, line, and interface are expected to be configured and operational before you configure DDR. See the relevant chapters in the "Preparing for Dial Access" part of this manual.
2. Encapsulation; including encapsulation for other WANs. See the "Configuring Media-Independent PPP and Multilink PPP" chapter of this publication for PPP encapsulation and refer to the *Cisco IOS Wide-Area Networking Configuration Guide* for sections on Frame Relay and X.25.
3. Bridging configurations. Refer to the *Cisco IOS Bridging and IBM Networking Configuration Guide*.
4. Routed protocols to be supported. See the protocol-specific chapters and publications.
5. Dialer profiles and legacy DDR are described in different chapters of the "Dial-on-Demand Routing" part of this publication.
6. Complex DDR configurations. Refer to the chapter "Configuring Media-Independent PPP and Multilink PPP" in this publication.

The DDR chapters provide complete configuration information about the simple hub-and-spoke DDR configurations, about the dialer profiles implementation of DDR, and about preparations required for configuring asynchronous interfaces for DDR.

DDR Topology Decisions

Topology decisions determine which routers will use DDR, which media and interfaces each one will use for DDR, and how each interface will function when using DDR. For example, if you choose a hub-and-spoke topology, one router will communicate with multiple routers. You must decide whether that router will use one interface or multiple interfaces for DDR, and whether it will receive calls only (forcing the spokes to initiate and bear the cost of calls). If it will use multiple interfaces, you must decide whether they will be of different types or the same type.

DDR-Independent Implementation Decisions

DDR-independent implementation decisions include the following:

- Using a specific interface or combination of interfaces for DDR.

  For complete configuration steps for the various media and interfaces, see the chapters in the "Dial-In Port Setup" part of this publication.

- Using nondefault encapsulations.

  The default encapsulation is High-Level Data Link Control (HDLC). However, PPP is widely used for situations in which authentication is desired, especially situations in which an interface will receive calls from multiple sites. Detailed PPP encapsulation requirements are described in the "Configuring Media-Independent PPP and Multilink PPP" and "Configuring Asynchronous PPP and SLIP" chapters of this publication.

If you decide to send DDR traffic over Frame Relay, X.25, or Link Access Procedure, Balanced (LAPB) networks, the interface must be configured with the appropriate encapsulation. For configuration details, refer to the related chapters in the Cisco *IOS Wide-Area Networking Configuration Guide*.

- Routing or bridging the DDR traffic.

  Legacy DDR supports bridging to only one destination, but the dialer profiles support bridging to multiple destinations.

  If you decide to bridge traffic over a dial-on-demand connection, configure the interface for transparent bridging. For detailed information, refer to the "Configuring Transparent Bridging" chapter of the *Cisco IOS Bridging and IBM Networking Configuration Guide*.

- Supporting one or more specific routed protocols, if you decide to route traffic.

  Depending on the protocol, you do need to control access by entering access lists and to decide how to support network addressing on an interface to be configured for DDR. You might also need to spoof keepalive or other packets. For configuration details, refer to the related network protocol chapters in the appropriate network protocols configuration guide, such as the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

DDR-Dependent Implementation Decisions

You must decide whether to implement legacy DDR or the newer dialer profiles; both are documented in the "Dial-on-Demand Routing" part of this publication. You must also decide whether a simple DDR configuration meets your business needs or whether to add other features.

Dialer Profiles

The dialer profiles implementation of DDR is based on a separation between logical and physical interface configuration. Dialer profiles also allow the logical and physical configurations to be bound together dynamically on a per-call basis.

Dialer profiles are advantageous in the following situations:

- When you want to share an interface (ISDN, asynchronous, or synchronous serial) to place or receive calls.
- When you want to change any configuration on a per-user basis.
- When you want to maximize ISDN channel usage using the Dynamic Multiple Encapsulations feature to configure various encapsulation types and per-user configurations on the same ISDN B channel at different times according to the type of call.
- When you want to bridge to many destinations, and for avoiding split horizon problem.

Most routed protocols are supported; however, International Organization for Standardization Connectionless Network Service (ISO CLNS) is not supported.

If you decide to configure dialer profiles, you must disable validation of source addresses for the routed protocols you support.

For detailed dialer profiles information, see the "Configuring Peer-to-Peer DDR with Dialer Profiles" chapter in this publication. For more information about Dynamic Multiple Encapsulations, see the "How to Configure Dialer Profiles" section in that chapter.

Legacy DDR

Legacy DDR is powerful and comprehensive, but its limitations affect scaling and extensibility. Legacy DDR is based on a static binding between the per-destination call specification and the physical interface configuration.

However, legacy DDR also has many strengths. It supports Frame Relay, ISO CLNS, LAPB, snapshot routing, and all routed protocols that are supported on Cisco routers. By default, legacy DDR supports fast switching.

For information about simple legacy DDR spoke configurations, see the "Configuring Legacy DDR Spokes" chapter. For information about simple legacy DDR hub configurations, see the "Configuring Legacy DDR Hubs" chapter. Both chapters are in this publication.

Simple or Complex DDR Configuration

You must also decide whether to implement a simple DDR configuration—whether it is a simple point-to-point (spoke-to-spoke) layout or a simple hub-and-spoke layout—or to add on features that make the implementation more complex. Add-on features include dial backup, bandwidth on demand, application of the Bandwidth Allocation Control Protocol (BACP), Multilink PPP, and many others.

Global and Interface Preparations for DDR

Some preparations are global and some depend on the type of interface you will configure for DDR. After you have made the required global decision whether to bridge or to route a specified protocol over a dial-on-demand link, you can make the following preparations:

- If you choose to bridge the protocol, decide whether to allow bridge packet access by Ethernet type codes or to permit all bridge packets across the link. Allowing access by Ethernet type codes requires you to define a bridging access list in global configuration mode.

  Allowing all bridge packets to trigger calls across a dial-on-demand link to a single destination is a DDR-dependent task addressed in the "Configure Dialer Access Lists to Trigger Outgoing Calls" section of both the "Configuring Legacy DDR Spokes" and "Configuring Legacy DDR Hubs" chapters in this publication.

  Bridging to multiple destinations requires dialer profiles.

- If you choose to route the protocol:
  - Define one or more access lists for the selected routed protocol to determine which packets should be permitted or denied access to the dial-on-demand link.

    Allowing those packets to trigger calls across a dial-on-demand link is a DDR-dependent task addressed in the "Configure Dialer Access Lists to Trigger Outgoing Calls" section of both the "Configuring Legacy DDR Spokes" and "Configuring Legacy DDR Hubs" chapters in this publication.
  - Define an appropriate dialer list for the protocol.
  - Disable validation of source addresses, if you decide to configure dialer profiles.

Preparations Depending on the Selected Interface Type

The steps shown in this chapter assume that you have also completed the required preparatory steps for the type of interface you will configure for DDR:

- The interface is installed, the cable is connected as needed, and operational.
- Chat scripts are ready, as needed, for any asynchronous interfaces and modem scripts have been assigned to the relevant asynchronous lines.
- Asynchronous lines and modems are configured and operational, as needed.
- Any ISDN line that will be used for DDR is properly provisioned and running.
- You have decided which interfaces and how many interfaces are to be configured for DDR, and what functions each interface will perform.

Preparations for Routing or Bridging over DDR

The following tasks are DDR-independent and can be completed before you configure DDR. Minimal tasks required for each item are presented in this chapter. For detailed information about bridging, routing, and wide-area networking configurations, refer to the appropriate chapters in other manuals of the Cisco IOS documentation set.

Complete the following minimal tasks for the global decisions you have made:

- Preparing for Transparent Bridging over DDR (As required)
- Preparing for Routing over DDR (As required)

Preparing for Transparent Bridging over DDR

To prepare for transparent bridging over DDR, complete the tasks in the following sections:

- Defining the Protocols to Bridge (As required)
- Specifying the Bridging Protocol (As required)
- Controlling Bridging Access (As required)

Defining the Protocols to Bridge

IP packets are routed by default unless they are explicitly bridged; all others are bridged by default unless they are explicitly routed. To bridge IP packets, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no ip routing`	Disables IP routing.

If you choose *not* to bridge another protocol supported on your network, use the relevant command to enable routing of that protocol. For more information about tasks and commands, refer to the relevant protocol chapter in the appropriate network protocols configuration guide, such as the *Cisco IOS AppleTalk and Novell IPX Configuration Guide* or *Cisco IOS IP Configuration Guide*.

Specifying the Bridging Protocol

You must specify the type of spanning-tree bridging protocol to use and also identify a bridge group. To specify the spanning-tree protocol and a bridge group number, use the following command in global configuration mode:

Command	Purpose
`Router(config)# bridge bridge-group protocol {ieee | dec}`	Defines the type of spanning tree protocol and identifies a bridge group.

The bridge-group number is used when you configure the interface and assign it to a bridge group. Packets are bridged only among members of the same bridge group.

Controlling Bridging Access

You can control access by defining any transparent bridge packet as *interesting*, or you can use the finer granularity of controlling access by Ethernet type codes.

To control access by Ethernet type codes, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# access-list access-list-number {permit | deny} type-code [mask]`	Identifies interesting packets by Ethernet type codes (access list numbers must be in the range 200–299).
Step 2	`Router(config)# dialer-list dialer-group protocol bridge list access-list-number`	Defines a dialer list for the specified access list.

Packets with a specified Ethernet type code can trigger outgoing calls. Spanning tree bridge protocol data units (BPDUs) are always treated as *uninteresting* and cannot trigger calls.

For a table of some common Ethernet types codes, refer to the "Ethernet Types Codes" appendix in the *Cisco IOS Bridging and IBM Networking Command Reference*.

To identify all transparent bridge packets as interesting, use the following command in global configuration mode:

Command	Purpose
`Router(config)# dialer-list dialer-group protocol bridge permit`	Defines a dialer list that treats all transparent bridge packets as interesting.

Preparing for Routing over DDR

DDR supports the following routed protocols: AppleTalk, Banyan VINES, DECnet, IP, Internet Protocol Exchange (IPX), ISO CLNS, and Xerox Network Systems (XNS).

To prepare for routing a protocol over DDR, perform the tasks in the relevant section:

- Configuring the Protocol for Routing and Access Control (As required)
- Associating the Protocol Access List with a Dialer Group (As required)

Configuring the Protocol for Routing and Access Control

This section specifies the minimal steps required to configure a protocol for routing over DDR. For more options and more detailed descriptions, refer to the relevant protocol chapter.

Configuring IP Routing

IP routing is enabled by default on Cisco routers; thus no preparation is required simply to enable it. You might, however, need to decide your addressing strategy and complete other global preparations for routing IP in your networks. To use dynamic routing where multiple remote sites communicate with each other through a central site, you might need to disable the IP split horizon feature. Refer to the "Configuring IP Addressing" chapter in the *Cisco IOS IP Configuration Guide* for more information.

At a minimum, you must complete the following tasks:

- Disable validation of source addresses.
- Configure one or more IP access lists before you refer to the access lists in DDR **dialer-list** commands to specify which packets can trigger outgoing calls.

To disable validation of source addresses, use the following commands in global configuration mode:

Command	Purpose
`Router(config)# router rip`	Specifies the routing protocol; RIP, for example.
`Router(config)# no validate-update-source`	Disables validation of source addresses.
`Router(config)# network number`	Specifies the IP address.

For more information about IP routing protocols, refer to the *Cisco IOS IP Configuration Guide*.

To configure IP access lists, use one of the following commands in global configuration mode:

Command	Purpose
`Router(config)# access-list access-list-number {deny | permit} source [source-mask]`	Specifies an IP standard access list.
or	
`Router(config)# access-list access-list-number {deny | permit} protocol source source-mask destination destination-mask [operator operand]`	Specifies an IP extended access list.

You can also use simplified IP access lists that use the **any** keyword instead of the numeric forms of source and destination addresses and masks. Other forms of IP access lists are also available. For more information, refer to the "IP Services Commands" chapter in the *Cisco IOS IP Configuration Guide*.

For an example of configuring DDR for IP, see the chapters "Configuring a Legacy DDR Spoke" or "Configuring a Legacy DDR Hub" in this publication.

You can configure IP routing on DDR asynchronous, synchronous serial, and ISDN interfaces, as well as dialer rotary groups.

Configuring Novell IPX Routing

To configure routing of IPX over DDR, you must complete both global and interface-specific tasks:

- Enable IPX routing globally.
- Enable IPX watchdog spoofing, or enable Sequenced Packet Exchange (SPX) keepalive spoofing on the interface.

To enable IPX routing, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx routing [node]`	Enables IPX routing.

To enable IPX watchdog spoofing on the interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx watchdog-spoof`	Enables IPX watchdog spoofing.

To enable SPX keepalive spoofing, use the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx spx-spoof`	Enables SPX keepalive spoofing.
`Router(config-if)# ipx spx-idle-time delay-in-seconds`	Sets the idle time after which SPX spoofing begins.

You can configure IPX routing on DDR asynchronous, synchronous serial, and ISDN interfaces, as well as dialer rotary groups.

For detailed DDR for IPX configuration examples, refer to the section "IPX over DDR Example" in the "Configuring Novell IPX" chapter of the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

Configuring AppleTalk Routing

You must enable AppleTalk routing and then specify AppleTalk access lists. After you specify AppleTalk access lists, define dialer lists. Use the **dialer-list protocol** command to define permit or deny conditions for the entire protocol; for a finer granularity, use the **dialer-list protocol** command with the **list** keyword.

You can configure AppleTalk routing on DDR asynchronous, synchronous serial, and ISDN interfaces, as well as dialer rotary groups.

See the chapters "Configuring a Legacy DDR Spoke" or "Configuring a Legacy DDR Hub" for more information and examples.

Configuring Banyan VINES Routing

To configure DDR for Banyan VINES, use one of the following commands in global configuration mode:

Command	Purpose
`Router(config)# vines access-list access-list-number {permit | deny} source source-mask1`	Specifies a VINES standard access list.
or	
`Router(config)# vines access-list access-list-number {permit | deny} source source-mask [destination] [destination-mask]`	Specifies a VINES extended access list.

After you specify VINES standard or extended access lists, define DDR dialer lists. Use the **dialer-list protocol** command to define permit or deny conditions for the entire protocol; for a finer granularity, use the **dialer-list protocol** command with the **list** keyword. See the chapters "Configuring a Legacy DDR Spoke" or "Configuring a Legacy DDR Hub" for more information and examples.

You can configure Banyan VINES on DDR asynchronous, synchronous serial, and ISDN interfaces, as well as dialer rotary groups.

Note The Banyan VINES **neighbor** command is not supported for LAPB and X.25 encapsulations.

Configuring DECnet Routing

To configure DDR for DECnet, use one of the following commands in global configuration mode:

Command	Purpose
`Router(config)# access-list access-list-number {permit | deny} source source-mask1`	Specifies a DECnet standard access list.
or	
`Router(config)# access-list access-list-number {permit | deny} source source-mask [destination] [destination-mask]`	Specifies a DECnet extended access list.

After you specify DECnet standard or extended access lists, define DDR dialer lists. Use the **dialer-list protocol** command to define permit or deny conditions for the entire protocol; for a finer granularity, use the **dialer-list protocol** command with the **list** keyword. See the chapters "Configuring a Legacy DDR Spoke" or "Configuring a Legacy DDR Hub" in this publication for more information and examples.

You classify DECnet control packets, including hello packets and routing updates, using one or more of the following commands: **dialer-list protocol decnet_router-L1 permit**, **dialer-list protocol decnet_router-L2 permit**, and **dialer-list protocol decnet_node permit**.

You can configure DECnet on DDR asynchronous, synchronous serial, and ISDN interfaces, as well as dialer rotary groups.

Configuring ISO CLNS Routing

To configure ISO CLNS for DDR, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# clns filter-set name [permit | deny] template`	Specifies one or more CLNS filters, repeating this command as needed to build the filter list associated with the filter name.
Step 2	`Router(config)# interface type number`	Specifies the interface to apply the filter to and begins interface configuration mode.
Step 3	`Router(config-if)# clns access-group name out`	Filters CLNS traffic going out of the interface, on the basis of the filter specified and named in Step 1.

After you complete these CLNS-specific steps, define a dialer list for CLNS. Use the **dialer-list protocol** command to define permit or deny conditions for the entire protocol; for a finer granularity, use the **dialer-list protocol** command with the **list** keyword. Use the *access-group* argument with this command, because ISO CLNS uses access groups but does not use access lists. See the chapters "Configuring a Legacy DDR Spoke" or "Configuring a Legacy DDR Hub" in this publication for more information and examples.

You classify CLNS control packets, including hello packets and routing updates, using the **dialer-list protocol clns_is permit** and/or **dialer-list protocol clns_es permit** command.

You can configure ISO CLNS on DDR asynchronous, synchronous serial, and ISDN interfaces, as well as dialer rotary groups.

Configuring XNS Routing

You must enable XNS routing and then define an access list. To define an XNS access list, use one of the following commands in global configuration mode:

Command	Purpose
`Router(config)# access-list access-list-number {deny | permit} source-network[.source-address [source-address-mask]] [destination-network[.destination-address [destination-address-mask]]]`	Specifies a standard XNS access list.
or	
`Router(config)# access-list access-list-number {deny | permit} protocol [source-network[.source-host [source-network-mask.]source-host-mask] source-socket [destination-network [.destination-host [destination-network-mask.destination-host-mask] destination-socket[/pep]]]`	Specifies an extended XNS access list.

After you specify an XNS access list, define a DDR dialer list. Use the **dialer-list protocol** command to define permit or deny conditions for the entire protocol; for a finer granularity, use the **dialer-list protocol** command with the **list** keyword. See the chapters "Configuring a Legacy DDR Spoke" or "Configuring a Legacy DDR Hub" for more information and examples.

You can configure XNS on DDR asynchronous, synchronous serial, and ISDN interfaces, as well as dialer rotary groups.

Associating the Protocol Access List with a Dialer Group

DDR supports the following routed protocols: AppleTalk, Banyan VINES, DECnet, IP, Novell IPX, ISO CLNS, and XNS.

You can permit or deny access by protocol, or you can specify an access list for more refined control. To associate a protocol or access list with a dialer group, use the following command in global configuration mode:

Command	Purpose
`Router(config)# dialer-list` *dialer-group* `protocol` *protocol-name* {`permit` | `deny` | `list` *access-list-number* | *access-group*}	Associates a protocol access list number or access group name with the dialer group.

Note For a given protocol and a given dialer group, only one access list can be specified in the **dialer-list** command.

For the **dialer-list protocol list** command form, acceptable access list numbers are as follows:

- Banyan VINES, DECnet, IP, and XNS standard and extended access list numbers
- Novell IPX standard, extended, and SAP access list numbers
- AppleTalk access lists numbers
- Bridge type codes

Configuration Examples for Legacy DDR

The following sections provide DDR configuration examples:

- Point-to-Point DDR Without Authentication Examples
- Point-to-Point DDR with Authentication Examples

Point-to-Point DDR Without Authentication Examples

The following example sets up two-way reciprocal DDR without authentication; the client and server have dial-in access to each other. This configuration is demonstrated in the following two subsections.

Remote Configuration

The following sample configuration is performed on the remote side of the connection:

```
interface ethernet 0
 ip address 172.30.44.1 255.255.255.0
!
interface async 7
 ip address 172.30.45.2 255.255.255.0
 async mode dedicated
 peer default ip address 172.30.45.1
 encapsulation ppp
 dialer in-band
```

```
 dialer string 1234
 dialer-group 1
!
ip route 172.30.43.0 255.255.255.0 async 7
 ip default-network 172.30.0.0
 chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
 dialer-list 1 protocol ip permit
!
line 7
 no exec
 modem InOut
 speed 38400
 flowcontrol hardware
 script dialer generic
```

Local Configuration

The following sample configuration is performed on the local side of the connection:

```
interface ethernet 0
 ip address 172.30.43.1 255.255.255.0
!
interface async 7
 async mode dedicated
 peer default ip address 172.30.45.2
 encapsulation ppp
 dialer in-band
 dialer string 1235
 dialer rotary-group 1
!
interface async 8
 async mode dedicated
 peer default ip address 172.30.45.2
 dialer rotary-group 1
!
ip route 172.30.44.0 255.255.255.0 async 7
 ip address 172.30.45.2 255.255.255.0
 encapsulation ppp
 ppp authentication chap
 dialer in-band
 dialer map ip 172.30.45.2 name remote 4321
 dialer load-threshold 80
!
ip route 172.30.44.0 255.255.255.0 128.150.45.2
 chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
 dialer-list 1 protocol ip permit
!
route igrp 109
network 172.30.0.0
redistribute static
passive-interface async 7
!
line 7
 modem InOut
 speed 38400
 flowcontrol hardware
 script dialer generic
```

Point-to-Point DDR with Authentication Examples

The following sample sets up two-way DDR with authentication; the client and server have dial-in access to each other. This configuration is demonstrated in the following two subsections.

Remote Configuration

The following example is performed on the remote side of the connection. It provides authentication by identifying a password that must be provided on each end of the connection.

```
username local password secret1
username remote password secret2
interface ethernet 0
 ip address 172.30.44.1 255.255.255.0
!
interface async 7
 ip address 172.30.45.2 255.255.255.0
 async mode dedicated
 peer default ip address 172.30.45.1
 encapsulation ppp
 dialer in-band
 dialer string 1234
 dialer-group 1
!
ip route 172.30.43.0 255.255.255.0 async 7
 ip default-network 172.30.0.0
 chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
 dialer-list 1 protocol ip permit
!
line 7
 no exec
 modem InOut
 speed 38400
 flowcontrol hardware
 script dialer generic
```

Local Configuration

The following example configuration is performed on the local side of the connection. As with the remote side configuration, it provides authentication by identifying a password for each end of the connection.

```
username remote password secret1
username local password secret2
!
interface ethernet 0
 ip address 172.30.43.1 255.255.255.0
!
interface async 7
 async mode dedicated
 peer default ip address 172.30.45.2
 dialer rotary-group 1
!
interface async 8
 async mode dedicated
 peer default ip address 172.30.45.2
 dialer rotary-group 1
!
interface dialer 1
 ip address 172.30.45.2 255.255.255.0
 encapsulation ppp
```

```
 ppp authentication chap
 dialer in-band
 dialer map ip 172.30.45.2 name remote 4321
 dialer load-threshold 80
!
ip route 172.30.44.0 255.255.255.0 172.30.45.2
 chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
!
 route igrp 109
 network 172.30.0.0
 redistribute static
 passive-interface async 7
!
line 7
 modem InOut
 speed 38400
 flowcontrol hardware
 script dialer generic
```