

VYATTA, INC.

| **Vyatta System**

Encapsulations

REFERENCE GUIDE

Cisco HDLC
Frame Relay
Classical IPoA
Bridged Ethernet



Vyatta
Suite 200
1301 Shoreway Road
Belmont, CA 94002
vyatta.com
650 413 7200
1 888 VYATTA 1 (US and Canada)

COPYRIGHT

Copyright © 2005–2010 Vyatta, Inc. All rights reserved.

Vyatta reserves the right to make changes to software, hardware, and documentation without notice. For the most recent version of documentation, visit the Vyatta web site at vyatta.com.

PROPRIETARY NOTICES

Vyatta is a registered trademark of Vyatta, Inc.

VMware, VMware ESXi, and VMware Server are trademarks of VMware, Inc.

XenServer and XenCenter are trademarks of Citrix Systems, Inc.

All other trademarks are the property of their respective owners.

ISSUE DATE: April 2010

DOCUMENT REVISION: R6.0 v03

RELEASED WITH: R6.0

PART NO. A0-0227-10-0004

Table of Contents

Quick Reference to Commands	v
Quick List of Examples	vii
Preface	ix
Intended Audience	x
Organization of This Guide	x
Document Conventions	xi
Advisory Paragraphs	xi
Typographic Conventions	xi
Vyatta Publications	xii
Chapter 1 Cisco HDLC	1
Cisco HDLC Commands	2
clear interfaces serial <wanx> counters cisco-hdlc	4
interfaces serial <wanx> cisco-hdlc	6
interfaces serial <wanx> cisco-hdlc keepalives	8
interfaces serial <wanx> cisco-hdlc mtu <mtu>	10
interfaces serial <wanx> cisco-hdlc vif 1 address local-address <ipv4>	12
interfaces serial <wanx> cisco-hdlc vif 1 address prefix-length <prefix>	14
interfaces serial <wanx> cisco-hdlc vif 1 address remote-address <ipv4>	16
interfaces serial <wanx> cisco-hdlc vif 1 description <desc>	18
interfaces serial <wanx> encapsulation cisco-hdlc	20
show interfaces serial <wanx> cisco-hdlc	21
Chapter 2 Frame Relay	22
Frame Relay Commands	23
clear interfaces serial <wanx> counters frame-relay	25
interfaces serial <wanx> encapsulation frame-relay	26
interfaces serial <wanx> frame-relay	27
interfaces serial <wanx> frame-relay mtu <mtu>	29

interfaces serial <wanx> frame-relay signaling <value>	31
interfaces serial <wanx> frame-relay signaling-options	33
interfaces serial <wanx> frame-relay vif <dlci> address local-address <ipv4>	36
interfaces serial <wanx> frame-relay vif <dlci> address prefix-length <prefix>	38
interfaces serial <wanx> frame-relay vif <dlci> address remote-address <ipv4>	40
interfaces serial <wanx> frame-relay vif <dlci> description <desc>	42
interfaces serial <wanx> frame-relay vif <dlci> pvc rx-inverse-arp	44
interfaces serial <wanx> frame-relay vif <dlci> pvc tx-inverse-arp <value>	46
show interfaces serial <wanx> frame-relay	48
Chapter 3 Classical IPoA	49
Classical IPoA Configuration	50
Classical IPoA Overview	50
Classical IPoA Configuration Example	50
Classical IPoA Commands	52
clear interfaces connection <adslx>	54
connect interface <adslx>	55
disconnect interface <adslx>	56
interfaces adsl <adslx> pvc <pvc-id> classical-ipoa	57
interfaces adsl <adslx> pvc <pvc-id> classical-ipoa local-address <ipv4>	59
interfaces adsl <adslx> pvc <pvc-id> classical-ipoa mtu <mtu>	61
interfaces adsl <adslx> pvc <pvc-id> classical-ipoa prefix-length <prefix>	63
interfaces adsl <adslx> pvc <pvc-id> classical-ipoa remote-address <ipv4>	65
Chapter 4 Bridged Ethernet	67
Bridged Ethernet Configuration	67
Bridged Ethernet Overview	67
Bridged Ethernet Configuration Example	68
Bridged Ethernet Commands	70
interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet	71
interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet local-address <ipv4>	73
interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet mtu <mtu>	75
interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet prefix-length <prefix>	77
interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet remote-address <ipv4>	79
Glossary of Acronyms	81

Quick Reference to Commands

Use this section to help you quickly locate a command.

clear interfaces connection <adslx>	54
clear interfaces serial <wanx> counters cisco-hdlc	4
clear interfaces serial <wanx> counters frame-relay	25
connect interface <adslx>	55
disconnect interface <adslx>	56
interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet	71
interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet local-address <ipv4>	73
interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet mtu <mtu>	75
interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet prefix-length <prefix>	77
interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet remote-address <ipv4>	79
interfaces adsl <adslx> pvc <pvc-id> classical-ipoa	57
interfaces adsl <adslx> pvc <pvc-id> classical-ipoa local-address <ipv4>	59
interfaces adsl <adslx> pvc <pvc-id> classical-ipoa mtu <mtu>	61
interfaces adsl <adslx> pvc <pvc-id> classical-ipoa prefix-length <prefix>	63
interfaces adsl <adslx> pvc <pvc-id> classical-ipoa remote-address <ipv4>	65
interfaces serial <wanx> cisco-hdlc	6
interfaces serial <wanx> cisco-hdlc keepalives	8
interfaces serial <wanx> cisco-hdlc mtu <mtu>	10
interfaces serial <wanx> cisco-hdlc vif 1 address local-address <ipv4>	12
interfaces serial <wanx> cisco-hdlc vif 1 address prefix-length <prefix>	14
interfaces serial <wanx> cisco-hdlc vif 1 address remote-address <ipv4>	16
interfaces serial <wanx> cisco-hdlc vif 1 description <desc>	18
interfaces serial <wanx> encapsulation cisco-hdlc	20
interfaces serial <wanx> encapsulation frame-relay	26
interfaces serial <wanx> frame-relay	27
interfaces serial <wanx> frame-relay mtu <mtu>	29
interfaces serial <wanx> frame-relay signaling <value>	31
interfaces serial <wanx> frame-relay signaling-options	33
interfaces serial <wanx> frame-relay vif <dldci> address local-address <ipv4>	36

interfaces serial <wanx> frame-relay vif <dldci> address prefix-length <prefix>	38
interfaces serial <wanx> frame-relay vif <dldci> address remote-address <ipv4>	40
interfaces serial <wanx> frame-relay vif <dldci> description <desc>	42
interfaces serial <wanx> frame-relay vif <dldci> pvc rx-inverse-arp	44
interfaces serial <wanx> frame-relay vif <dldci> pvc tx-inverse-arp <value>	46
show interfaces serial <wanx> cisco-hdlc	21
show interfaces serial <wanx> frame-relay	48

Quick List of Examples

Use this list to help you locate examples you'd like to try or look at.

Example 1-1	Clearing a Cisco HDLC interface.	4
Example 3-1	Creating and configuring an ADSL interface for Classical IPoA encapsulation	51
Example 4-1	Creating and configuring an ADSL interface for Bridged Ethernet encapsulation	68

Preface

This guide explains how to configure and use encapsulated and tunneled interfaces on the Vyatta system. It describes the available commands and provides configuration examples.

This preface provides information about using this guide. The following topics are covered:

- Intended Audience
- Organization of This Guide
- Document Conventions
- Vyatta Publications

Intended Audience

This guide is intended for experienced system and network administrators. Depending on the functionality to be used, readers should have specific knowledge in the following areas:

- Networking and data communications
- TCP/IP protocols
- General router configuration
- Routing protocols
- Network administration
- Network security

Organization of This Guide

This guide has the following aid to help you find the information you are looking for:

- **Quick Reference to Commands**

Use this section to help you quickly locate a command.

- **Quick List of Examples**

Use this list to help you locate examples you'd like to try or look at.

This guide has the following chapters:

Chapter	Description	Page
Chapter 1: Cisco HDLC	This chapter describes commands for Cisco HDLC encapsulation on serial interfaces on the Vyatta system.	1
Chapter 2: Frame Relay	This chapter describes commands for configuring Frame Relay encapsulation on serial interfaces on the Vyatta system.	22
Chapter 3: Classical IPoA	This chapter describes the commands for setting up Classical IPoA encapsulation on ADSL interfaces on the Vyatta system.	49
Chapter 4: Bridged Ethernet	This chapter describes the commands for setting up Bridged Ethernet encapsulation (RFC 1483) on ADSL interfaces on the Vyatta system.	67
Glossary of Acronyms		81

Document Conventions

This guide contains advisory paragraphs and uses typographic conventions.

Advisory Paragraphs

This guide uses the following advisory paragraphs:

Warnings alert you to situations that may pose a threat to personal safety, as in the following example:



WARNING *Switch off power at the main breaker before attempting to connect the remote cable to the service power at the utility box.*

Cautions alert you to situations that might cause harm to your system or damage to equipment, or that may affect service, as in the following example:



CAUTION *Restarting a running system will interrupt service.*

Notes provide information you might need to avoid problems or configuration errors:

NOTE *You must create and configure network interfaces before enabling them for routing protocols.*

Typographic Conventions

This document uses the following typographic conventions:

<i>Monospace</i>	Examples, command-line output, and representations of configuration nodes.
bold Monospace	Your input: something you type at a command line.
bold	Commands, keywords, and file names, when mentioned inline. Objects in the user interface, such as tabs, buttons, screens, and panes.
<i>italics</i>	An argument or variable where you supply a value.
<key>	A key on your keyboard, such as <Enter>. Combinations of keys are joined by plus signs (“+”), as in <Ctrl>+c.
[<i>arg1</i> <i>arg2</i>]	Enumerated options for completing a syntax. An example is [enable disable].
<i>num1–numN</i>	A inclusive range of numbers. An example is 1–65535, which means 1 through 65535, inclusive.
<i>arg1..argN</i>	A range of enumerated values. An example is eth0..eth3, which means eth0, eth1, eth2, or eth3.
<i>arg</i> [<i>arg...</i>] <i>arg</i> [, <i>arg...</i>]	A value that can optionally represent a list of elements (a space-separated list in the first case and a comma-separated list in the second case).

Vyatta Publications

More information about the Vyatta system is available in the Vyatta technical library, and on www.vyatta.com and www.vyatta.org.

Full product documentation is provided in the Vyatta technical library. To see what documentation is available for your release, see the *Guide to Vyatta Documentation*. This guide is posted with every release of Vyatta software and provides a great starting point for finding the information you need.

Chapter 1: Cisco HDLC

This chapter describes commands for Cisco HDLC encapsulation on serial interfaces on the Vyatta system.



This feature is available only in the Vyatta Subscription Edition.

This chapter presents the following topics:

- Cisco HDLC Commands

Cisco HDLC Commands

This chapter contains the following commands.

Configuration Commands

interfaces serial <wan> cisco-hdlc	Defines the characteristics of Cisco High-Level Data Link Control encapsulation for a serial interface.
interfaces serial <wan> cisco-hdlc keepalives	Set the characteristics of keep-alive messages for a Cisco HDLC serial interface.
interfaces serial <wan> cisco-hdlc mtu <mtu>	Specify the Maximum Transmit Unit (MTU) size for a Cisco HDLC serial interface.
interfaces serial <wan> cisco-hdlc vif 1 address local-address <ipv4>	Assign an IP address to a Cisco HDLC virtual interface.
interfaces serial <wan> cisco-hdlc vif 1 address prefix-length <prefix>	Specifies the prefix defining the network served by a virtual interface on a Cisco HDLC serial interface.
interfaces serial <wan> cisco-hdlc vif 1 address remote-address <ipv4>	Specifies the IP address of the remote endpoint on a Cisco HDLC serial connection.
interfaces serial <wan> cisco-hdlc vif 1 description <desc>	Specifies a description for a Cisco HDLC virtual interface.

Operational Commands

clear interfaces serial <wan> counters cisco-hdlc	Clears counters for Cisco HDLC–encapsulated serial interfaces
show interfaces serial <wan> cisco-hdlc	Displays Cisco HDLC serial interface information.

Commands for using other system features with Cisco HDLC–encapsulated interfaces can be found in the following locations.

Related Commands Documented Elsewhere

Serial interfaces	Commands for clearing and configuring serial interfaces and displaying serial interface information are described in the <i>Vyatta WAN Interfaces Reference Guide</i> .
Firewall	Commands for configuring firewall on serial interfaces are described in the <i>Vyatta Firewall Reference Guide</i> .
OSPF	Commands for configuring the Open Shortest Path First routing protocol on serial interfaces are described in the <i>Vyatta OSPF Reference Guide</i> .
RIP	Commands for configuring the Routing Information Protocol on serial interfaces are described in the <i>Vyatta RIP Reference Guide</i> .

QoS	Commands for configuring quality of service on serial interfaces are described in the <i>Vyatta QoS Reference Guide</i> .
System interfaces	Commands for showing the physical interfaces available on your system are described in the <i>Vyatta Basic System Reference Guide</i> .
VRRP	Commands for configuring Virtual Router Redundancy Protocol on serial interfaces are described in the <i>Vyatta High Availability Reference Guide</i> .

clear interfaces serial <wanx> counters cisco-hdlc

Clears counters for Cisco HDLC–encapsulated serial interfaces

Syntax

```
clear interfaces serial wanx counters cisco-hdlc]
```

Command Mode

Operational mode.

Parameters

<i>wanx</i>	The identifier of a configured serial interface.
-------------	--

Usage Guidelines

Use this command to clear statistics for a Cisco HDLC–encapsulated serial interface.

Example 1-1 shows the result of using this command.

Example 1-1 Clearing a Cisco HDLC interface.

```
vyatta@R1> clear interfaces serial wan0 counters cisco-hdlc
DSU/CSU Perfomance Monitoring counters were flushed.
Performance monitoring counters flushed
-----
wan0.1: SLARP STATISTICS
-----
SLARP frame transmission/reception statistics
  SLARP request packets transmitted: 0
    SLARP request packets received: 0
  SLARP Reply packets transmitted: 0
    SLARP Reply packets received: 0
  SLARP keepalive packets transmitted: 0
    SLARP keepalive packets received: 0
Incoming SLARP Packets with format errors
  Invalid SLARP Code: 0
  Replies with bad IP addr: 0
  Replies with bad netmask: 0
SLARP timeout/retry statistics
  SLARP Request timeouts: 0
```

```
keepalive reception timeouts: 0
Cisco Discovery Protocol frames
                               Transmitted: 0
                               Received: 0
DSU/CSU Performance Monitoring counters were flushed.
vyatta@R1>
```

Defines the characteristics of Cisco High-Level Data Link Control encapsulation for a serial interface.

set interfaces serial *wanx* **cisco-hdlc**
delete interfaces serial *wanx* **cisco-hdlc**
show interfaces serial *wanx* **cisco-hdlc**

Configuration mode.

Configuration Statement

```
interfaces {
  serial wan0..wan23 {
    cisco-hdlc {
    }
  }
}
```

Parameters

<i>wanx</i>	Mandatory. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
-------------	---

Default

None.

Usage Guidelines

Use this command to define the Cisco High-Level Data Link Control characteristics of the line.

Note that on Cisco HDLC interfaces, IP addresses are assigned to virtual interfaces, not directly to the interface. Currently, only one vif is supported, but multiple addresses may be defined for the vif.

The full identifier of an HDLC interface is *int cisco-hdlc vif vif*. For example, the full identifier of the HDLC vif on wan1 is **wan1 cisco-hdlc vif 1**. Note that subsequent to initial definition, the notation for referring to this is *int.vif*—that is, **wan1.1**.

Use the **set** form of this command to create a cisco-hdlc serial interface, provided the interface physically exists on your system.

To see the interfaces available to the system kernel, use the **system** option of use the **show interfaces system** command, which is described in the *Vyatta Basic System Reference Guide*.

Note that you cannot use **set** to change the identifier of configuration nodes. To change the identifier of a configuration node, you must **delete** the old configuration node and create a new one with the correct identifier.

Use the **delete** form of this command to remove all configuration for a Cisco HDLC serial interface.

Use the **show** form of this command to view Cisco HDLC serial interface configuration.

interfaces serial <wanx> cisco-hdlc keepalives

Set the characteristics of keep-alive messages for a Cisco HDLC serial interface.

Syntax

```
set interfaces serial wanx cisco-hdlc keepalives [require-rx {enable | disable} | timer interval]
```

```
delete interfaces serial wanx cisco-hdlc keepalives require-rx
```

```
show interfaces serial wanx cisco-hdlc keepalives require-rx
```

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  serial wan0..wan23 {
    cisco-hdlc {
      keepalives {
        require-rx [enable|disable]
        timer 10-60000
      }
    }
  }
}
```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
require-rx	Specifies whether or not to require receiving keep-alive message from the Cisco HDLC peer. Supported values are as follows: enable : Requires keep-alive messages. If keep-alive messages are not received, the peer interface is declared down. disable : Does not require keep-alive messages.
timer interval	Specifies the interval at which keep-alive messages are to be sent. The range is 10 to 60000. The default is 10.

Default

None.

Usage Guidelines

Use this command to set the characteristics of keep-alive messages on a Cisco HDLC interface.

Use the **set** form of this command to specify keep-alive information.

Use the **delete** form of this command to restore the default keep-alive configuration.

Use the **show** form of this command to view keep-alive configuration.

interfaces serial <wanx> cisco-hdlc mtu <mtu>

Specify the Maximum Transmit Unit (MTU) size for a Cisco HDLC serial interface.

Syntax

set interfaces serial *wanx* **cisco-hdlc mtu** *mtu*

delete interfaces serial *wanx* **cisco-hdlc mtu**

show interfaces serial *wanx* **cisco-hdlc mtu**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  serial wan0..wan23 {
    cisco-hdlc {
      mtu 8-8188
    }
  }
}
```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
<i>mtu</i>	The maximum packet size that the interface will send. The range is 8 to 8188. The default is 1500.

Default

The MTU is 1500.

Usage Guidelines

Use this command to specify the Maximum Transmit Unit. This is the maximum packet size the interface will send.

Use the **set** form of this command to specify the MTU.

Use the **delete** form of this command to restore the default MTU.

Use the **show** form of this command to view MTU configuration.

interfaces serial <wanx> cisco-hdlc vif 1 address local-address <ipv4>

Assign an IP address to a Cisco HDLC virtual interface.

Syntax

set interfaces serial *wanx* **cisco-hdlc vif 1 address local-address** *ipv4*

delete interfaces serial *wanx* **cisco-hdlc vif 1 address local-address**

show interfaces serial *wanx* **cisco-hdlc vif 1 address local-address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  serial wan0..wan23 {
    cisco-hdlc {
      vif 1 {
        address {
          local-address ipv4
        }
      }
    }
  }
}
```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
1	The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1.
<i>ipv4</i>	Mandatory. The IPv4 address for this vif. Each serial vif can support exactly one IP address.

Default

None.

Usage Guidelines

Use this command to specify an IP address for a Cisco HDLC virtual interface.

Use the **set** form of this command to set the IP address.

Use the **delete** form of this command to remove IP address configuration.

Use the **show** form of this command to view IP address configuration.

interfaces serial <wanx> cisco-hdlc vif 1 address prefix-length <prefix>

Specifies the prefix defining the network served by a virtual interface on a Cisco HDLC serial interface.

Syntax

set interfaces serial *wanx* **cisco-hdlc vif 1 address prefix-length** *prefix*

delete interfaces serial *wanx* **cisco-hdlc vif 1 address prefix-length**

show interfaces serial *wanx* **cisco-hdlc vif 1 address prefix-length**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  serial wan0..wan23 {
    cisco-hdlc {
      vif 1 {
        address {
          prefix-length 0-32
        }
      }
    }
  }
}
```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
1	The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1.
<i>prefix</i>	Mandatory. The prefix defining the network served by this interface. The range is 0 to 32.

Default

None.

Usage Guidelines

Use this command to specify the prefix defining the network served by this virtual interface.

Use the **set** form of this command to specify the network prefix.

Use the **delete** form of this command to remove network prefix configuration.

Use the **show** form of this command to view network prefix configuration.

interfaces serial <wanx> cisco-hdlc vif 1 address remote-address <ipv4>

Specifies the IP address of the remote endpoint on a Cisco HDLC serial connection.

Syntax

set interfaces serial *wanx* **cisco-hdlc vif 1 address remote-address** *ipv4*

delete interfaces serial *wanx* **cisco-hdlc vif 1 address remote-address**

show interfaces serial *wanx* **cisco-hdlc vif 1 address remote-address**

Command Mode

Configuration mode.

Configuration Statement

```

interfaces {
  serial wan0..wan23 {
    cisco-hdlc {
      vif 1 {
        address {
          remote-address ipv4
        }
      }
    }
  }
}

```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
1	The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1.
<i>ipv4</i>	Mandatory. An IP address representing the remote endpoint.

Default

None.

Usage Guidelines

Use this command to specify the IP address representing the remote endpoint.

Use the **set** form of this command to set the remote endpoint's IP address.

Use the **delete** form of this command to remove remote address configuration.

Use the **show** form of this command to view remote address configuration.

interfaces serial <wanx> cisco-hdlc vif 1 description <desc>

Specifies a description for a Cisco HDLC virtual interface.

Syntax

set interfaces serial *wanx* **cisco-hdlc vif 1**description *desc*

delete interfaces serial *wanx* **cisco-hdlc vif 1** description

show interfaces serial *wanx* **cisco-hdlc vif 1** description

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  serial wan0..wan23 {
    cisco-hdlc {
      vif 1 {
        description text
      }
    }
  }
}
```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
1	The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1.
<i>desc</i>	Optional. A brief description for the virtual interface. If the description contains spaces, it must be enclosed in double quotes.

Default

None.

Usage Guidelines

Use this command to specify a description for the virtual interface.

Use the **set** form of this command to set the description for the virtual interface.

Use the **delete** form of this command to remove description configuration.

Use the **show** form of this command to view description configuration.

interfaces serial <wanx> encapsulation cisco-hdlc

Sets the Cisco HDLC as the encapsulation type for a serial interface.

Syntax

```
set interfaces serial wanx encapsulation cisco-hdlc  
delete interfaces serial wanx encapsulation  
show interfaces serial wanx encapsulation
```

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {  
    serial wan0..wan23 {  
        encapsulation cisco-hdlc  
    }  
}
```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
-------------	---

Default

None.

Usage Guidelines

Use this command to specify Cisco HDLC as the encapsulation type for a serial interface.

Use the **set** form of this command to set the encapsulation type.

Use the **delete** form of this command to remove encapsulation type configuration.

Use the **show** form of this command to view encapsulation type configuration.

show interfaces serial <wanx> cisco-hdlc

Displays Cisco HDLC serial interface information.

Syntax

```
show interfaces serial wanx cisco-hdlc
```

Command Mode

Operational mode.

Parameters

<i>wanx</i>	The name of a serial interface. If an interface is specified, you must also specify one of the cisco-hdlc , frame-relay , physical , ppp , or trace options.
-------------	---

Default

None.

Usage Guidelines

Use this command to view the operational status of a Cisco HDLC–encapsulated serial interface.

Chapter 2: Frame Relay

This chapter describes commands for configuring Frame Relay encapsulation on serial interfaces on the Vyatta system.



This chapter presents the following topics:

- Frame Relay Commands

Commands for using other system features with Frame Relay–encapsulated interfaces can be found in the following locations.

Related Commands Documented Elsewhere

clear interfaces serial <wanx> counters frame-relay

Clears counters for Frame Relay–encapsulated serial interfaces

Syntax

```
clear interfaces serial wanx counters frame-relay
```

Command Mode

Operational mode.

Parameters

<i>wanx</i>	The identifier of a configured serial interface.
-------------	--

Usage Guidelines

Use this command to clear statistics for a specified Frame Relay–encapsulated serial interface.

interfaces serial <wanx> encapsulation frame-relay

Sets the Frame Relay as the encapsulation type for a serial interface.

Syntax

set interfaces serial *wanx* **encapsulation frame-relay**

delete interfaces serial *wanx* **encapsulation**

show interfaces serial *wanx* **encapsulation**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  serial wan0..wan23 {
    encapsulation frame-relay
  }
}
```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
-------------	---

Default

None.

Usage Guidelines

Use this command to specify Frame Relay as the encapsulation type for a serial interface.

Use the **set** form of this command to set the encapsulation type.

Use the **delete** form of this command to remove encapsulation type configuration.

Use the **show** form of this command to view encapsulation type configuration.

interfaces serial <wanx> frame-relay

Defines the characteristics of Frame Relay encapsulation on a serial interface.

Syntax

```
set interfaces serial wanx frame-relay
delete interfaces serial wanx frame-relay
show interfaces serial wanx frame-relay
```

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  serial wan0..wan23 {
    frame-relay {
    }
  }
}
```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
-------------	---

Default

None.

Usage Guidelines

Use this command to define Frame Relay settings on an interface. This consists primarily of defining the signaling variant, the PVC characteristics, and the keep-alive (health checking) characteristics of the line.

The full identifier of an Frame Relay interface is *int* **frame-relay vif** *vif*. For example, the full identifier of the Frame Relay vif 16 on wan0 is **wan0 frame-relay vif 16**. Note that subsequent to initial definition, the notation for referring to this is *int.vif*²—that is, **wan0.16**.

Use the **set** form of this command to set Frame Relay characteristics.

Use the **delete** form of this command to remove all configuration for a Frame Relay serial interface.

Use the **show** form of this command to view a Frame Relay serial interface configuration.

interfaces serial <wanx> frame-relay mtu <mtu>

Specifies the Maximum Transmit Unit (MTU) size for a Frame Relay serial interface.

Syntax

set interfaces serial *wanx* **frame-relay mtu** *mtu*

delete interfaces serial *wanx* **frame-relay mtu**

show interfaces serial *wanx* **frame-relay mtu**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  serial wan0..wan23 {
    frame-relay {
      mtu u32
    }
  }
}
```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
<i>mtu</i>	The maximum packet size that the interface will send. The range is 8 to 8188. The default is 1500.

Default

The MTU is 1500.

Usage Guidelines

Use this command to specify the Maximum Transfer Unit (MTU) for a Frame Relay serial interface. This is the maximum packet size the interface will send.

Use the **set** form of this command to set the MTU.

Use the **delete** form of this command to restore the default MTU configuration.

Use the **show** form of this command to view MTU configuration.

interfaces serial <wanx> frame-relay signaling <value>

Specifies the signaling variant (LMI type) for a Frame Relay serial interface.

Syntax

set interfaces serial *wanx* **frame-relay signaling** *value*

delete interfaces serial *wanx* **frame-relay signaling**

show interfaces serial *wanx* **frame-relay signaling**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  serial wan0..wan23 {
    frame-relay {
      signaling [auto | ansi | q933 | lmi]
    }
  }
}
```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system. Sets the full status message polling interval.
<i>value</i>	Specifies the Frame Relay signaling variant (LMI type). Supported values are as follows: auto : Autonegotiates the LMI type. ansi : Uses ANSI-617d Annex D LMI type. q933 : Uses the Q.933 (ITU-T (CCIT) Q.933 annex A) LMI type. lmi : Uses Cisco proprietary LMI type. The default is auto.

Default

LMI type is automatically negotiated.

Usage Guidelines

Use this command to specify the signaling variant (LMI type) for a Frame Relay serial interface.

Use the **set** form of this command to set the signaling variant.

Use the **delete** form of this command to restore the default signaling variant.

Use the **show** form of this command to view signaling variant configuration.

interfaces serial <wanx> frame-relay signaling-options

Specifies the signaling options for a Frame Relay serial interface.

Syntax

```
set interfaces serial wanx frame-relay signaling-options [n391dte value | n392dte value  
| n393dte value | t391dte value | t392 value]
```

```
delete interfaces serial wanx frame-relay signaling-options
```

```
show interfaces serial wanx frame-relay signaling-options
```

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {  
  serial wan0..wan23 {  
    frame-relay {  
      signaling-options {  
        n391dte 1-255  
        n392dte 1-100  
        n393dte 1-10  
        t391dte 5-30  
        t392 5-30  
      }  
    }  
  }  
}
```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
-------------	---

n391dte value	<p>Optional. Sets the frequency at which the DTE expects a full status message in response to keep-alive messages. The DTE sends a keep-alive request to the DCE at the interval specified by the t391dte parameter. This parameter, n391dte, specifies the frequency at which the DTE expects the response. For example, if n391dte is set to 6 it means that the DTE expects a full status message to be sent in response to every 6th status enquiry.</p> <p>The range is 1 to 255. The default is 6.</p>
n392dte value	<p>Optional. Sets the DTE error threshold, which is the number of errors which, if they occur within the event count specified by the n393dte parameter, the link will be declared down.</p> <p>The range is 1 to 100. The default is 6.</p>
n393dte value	<p>Optional. Sets the DTE monitored event count. This parameter is used in conjunction with the number of errors specified in the n392dte parameter to determine whether a link should be declared down. If n392dte errors occur within n393dte events, the link is considered down.</p> <p>The range is 1 to 10. The default is 4.</p>
t391dte value	<p>Optional. Sets the DTE keep-alive timer. This is the interval, in seconds, at which the interface sends out a keep-alive request to the DCE interface, which should respond with a keep-alive message.</p> <p>At the interval defined by the n391dte option, the DCE will send a full status report instead of just a keep-alive message.</p> <p>The range is 5 to 30. The default is 10.</p>
t392 value	<p>Optional. Sets the DCE timer variable. This is the maximum time, in seconds, that the DCE will wait for a keep-alive request from the DTE interface. If this condition occurs a time-out is counted.</p> <p>This value must be greater than the DTE keep-alive interval specified by the t391dte parameter.</p> <p>The range is 5 to 30. The default is 16.</p>

Default

The default value for each parameter is used.

Usage Guidelines

Use this command to specify the Frame Relay signaling options. These options control how often the keepalive and full status enquiries are sent and expected.

Use the **set** form of this command to specify the signaling option for the interface.

Use the **delete** form of this command to remove the configuration.

Use the **show** form of this command to view the configuration.

interfaces serial <wanx> frame-relay vif <dci> address local-address <ipv4>

Sets the IP address for a Frame Relay virtual interface.

Syntax

set interfaces serial *wanx* **frame-relay vif** *dci* **address local-address** *ipv4*

delete interfaces serial *wanx* **frame-relay vif** *dci* **address local-address**

show interfaces serial *wanx* **frame-relay vif** *dci* **address local-address**

Command Mode

Configuration mode.

Configuration Statement

```

interfaces {
  serial wan0..wan23 {
    frame-relay {
      vif 16-991 {
        address {
          local-address ipv4
        }
      }
    }
  }
}

```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
<i>dci</i>	The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. The range is 16 to 991.
<i>ipv4</i>	Mandatory. The IPv4 address for this vif. Each serial vif can support exactly one IP address.

Default

None.

Usage Guidelines

Use this command to specify the IP address for a Frame Relay virtual interface.

Use the **set** form of this command to set the IP address.

Use the **delete** form of this command to remove IP address configuration.

Use the **show** form of this command to view IP address configuration.

interfaces serial <wanx> frame-relay vif <dlci> address prefix-length <prefix>

Specifies the prefix defining the network served by a Frame Relay virtual interface.

Syntax

set interfaces serial *wanx* **frame-relay vif** *dlci* **address prefix-length** *prefix*

delete interfaces serial *wanx* **frame-relay vif** *dlci* **address prefix-length**

show interfaces serial *wanx* **frame-relay vif** *dlci* **address prefix-length**

Command Mode

Configuration mode.

Configuration Statement

```

interfaces {
  serial wan0..wan23 {
    frame-relay {
      vif 16-991 {
        address {
          prefix-length 0-32
        }
      }
    }
  }
}

```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
<i>dlci</i>	The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. The range is 16 to 991.
<i>prefix</i>	Mandatory. The prefix defining the network served by this interface. The range is 0 to 32.

Default

None.

Usage Guidelines

Use this command to specify the prefix defining the network served by this virtual interface.

Use the **set** form of this command to set the network prefix.

Use the **delete** form of this command to remove network prefix configuration.

Use the **show** form of this command to view network prefix configuration.

interfaces serial <wanx> frame-relay vif <dlci> address remote-address <ipv4>

Sets the IP address for the remote endpoint of a Frame Relay connection.

Syntax

set interfaces serial *wanx* **frame-relay vif** *dlci* **address remote-address** *ipv4*

delete interfaces serial *wanx* **frame-relay vif** *dlci* **address remote-address**

show interfaces serial *wanx* **frame-relay vif** *dlci* **address remote-address**

Command Mode

Configuration mode.

Configuration Statement

```

interfaces {
  serial wan0..wan23 {
    frame-relay {
      vif 16-991 {
        address {
          remote-address ipv4
        }
      }
    }
  }
}

```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
<i>dlci</i>	The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. The range is 16 to 991.
<i>ipv4</i>	Mandatory. The IP address of the remote endpoint.

Default

None.

Usage Guidelines

Use this command to specify the IP address of the remote endpoint of a Frame Relay connection.

Use the **set** form of this command to set the remote address.

Use the **delete** form of this command to remove remote address configuration.

Use the **show** form of this command to view remote address configuration.

interfaces serial <wanx> frame-relay vif <dpci> description <desc>

Specifies a description for a Frame Relay virtual interface.

Syntax

set interfaces serial *wanx* **frame-relay vif** *dpci* **description** *desc*

delete interfaces serial *wanx* **frame-relay vif** *dpci* **description**

show interfaces serial *wanx* **frame-relay vif** *dpci* **description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  serial wan0..wan23 {
    frame-relay {
      vif 16-991 {
        description text
      }
    }
  }
}
```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
<i>dpci</i>	The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. The range is 16 to 991.
<i>desc</i>	Optional. A brief description for the virtual interface. If the description contains spaces, it must be enclosed in double quotes.

Default

None.

Usage Guidelines

Use this command to specify a description for a Frame Relay virtual interface.

Use the **set** form of this command to specify the description.

Use the **delete** form of this command to remove description configuration.

Use the **show** form of this command to view description configuration.

interfaces serial <wanx> frame-relay vif <dpci> pvc rx-inverse-arp

Enables or disables receiving of inverse ARP messages on a Frame Relay virtual interface.

Syntax

set interfaces serial *wanx* **frame-relay vif** *dpci* **pvc rx-inverse-arp** *state*

delete interfaces serial *wanx* **frame-relay vif** *dpci* **pvc rx-inverse-arp**

show interfaces serial *wanx* **frame-relay vif** *dpci* **pvc rx-inverse-arp**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  serial wan0..wan23 {
    frame-relay {
      vif 16-991 {
        pvc {
          rx-inverse-arp [enable|disable]
        }
      }
    }
  }
}
```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
<i>vlan-id</i>	The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. The range is 16 to 991.
<i>state</i>	Enables or disables inverse ARP on this DLCI. Supported values are as follows: enable : Enables inverse ARP on this DLCI. disable : Disables inverse ARP on this DLCI.

Default

Receiving of inverse ARP messages is disabled.

Usage Guidelines

Use this command to enable or disable inverse Address Resolution Protocol (inverse ARP) on this virtual interface. Inverse ARP allows you to determine a hosts' hardware address from its network address.

Use the **set** form of this command to enable or disable receiving inverse ARP on the virtual interface.

Use the **delete** form of this command to restore the default inverse ARP reception behavior.

Use the **show** form of this command to view inverse ARP reception configuration.

interfaces serial <wanx> frame-relay vif <dlci> pvc tx-inverse-arp <value>

Specifies the number of inverse ARP messages to be sent by a Frame Relay virtual interface.

Syntax

set interfaces serial *wanx* **frame-relay vif** *dlci* **pvc tx-inverse-arp** *value*

delete interfaces serial *wanx* **frame-relay vif** *dlci* **pvc tx-inverse-arp**

show interfaces serial *wanx* **frame-relay vif** *dlci* **pvc tx-inverse-arp**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  serial wan0..wan23 {
    frame-relay {
      vif 16-991 {
        pvc {
          tx-inverse-arp 0-84600
        }
      }
    }
  }
}
```

Parameters

<i>wanx</i>	Mandatory. Multi-node. The identifier for the serial interface you are defining. This may be wan0 to wan23 , depending on what serial interfaces that are actually available on the system.
<i>vlan-id</i>	The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. The range is 16 to 991.
<i>value</i>	Sets a limit on the number of inverse ARP messages that will be sent by the system for use in dynamic address-to-DLCI mapping. The range is 0 to 86400, where 0 means there is no limit.

Default

There is no limit to the number of inverse ARP messages that can be sent from the interface.

Usage Guidelines

Use this command to set the maximum number of inverse Address Resolution Protocol (inverse ARP) messages that can be sent from a Frame Relay virtual interface. Inverse ARP allows you to determine a hosts' hardware address from its network address.

Use the **set** form of this command to set the inverse ARP transmission limit.

Use the **delete** form of this command to restore the default inverse ARP transmission limit behavior.

Use the **show** form of this command to view inverse ARP transmission limit configuration.

show interfaces serial <wanx> frame-relay

Displays Frame Relay serial interface information.

Syntax

```
show interfaces serial wanx frame-relay [pvc | pvc-list [active]]
```

Command Mode

Operational mode.

Parameters

<i>wanx</i>	The name of a serial interface. If an interface is specified, you must also specify one of the cisco-hdlc , frame-relay , physical , ppp , or trace options.
pvc	Displays details for Frame Relay PVCs.
pvc-list	Lists Frame Relay permanent virtual circuits (PVCs). When used with no option, displays all configured PVCs.
active	Lists only active Frame Relay PVCs.

Default

Information is shown for all PVCs on the Frame Relay interface.

Usage Guidelines

Use this command to view the operational status of a Frame Relay–encapsulated serial interface.

Chapter 3: Classical IPoA

This chapter describes the commands for setting up Classical IPoA encapsulation on ADSL interfaces on the Vyatta system.



This feature is available only in the Vyatta Subscription Edition.

This chapter presents the following topics:

- Classical IPoA Configuration
- Classical IPoA Commands

Classical IPoA Configuration

This section presents the following topics:

- Classical IPoA Overview
- Classical IPoA Configuration Example

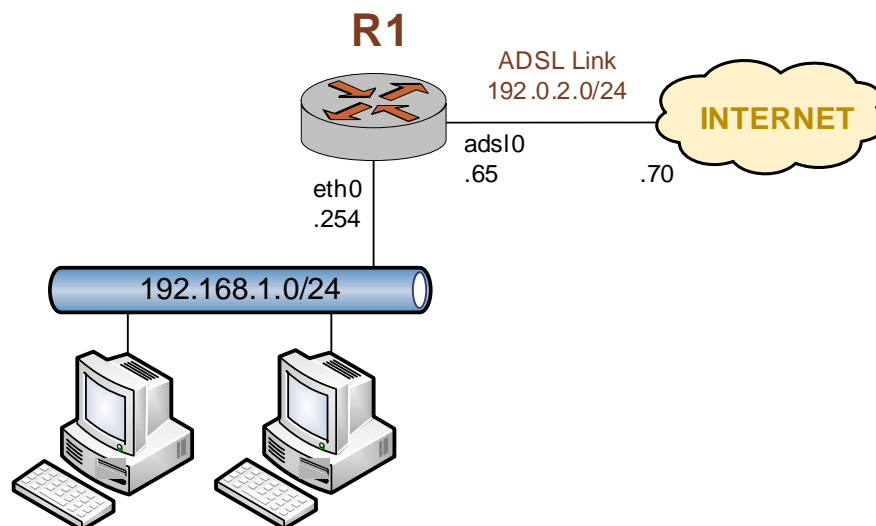
Classical IPoA Overview

Classical IPoA encapsulation for a permanent virtual circuit (PVC) on an ADSL interface is defined in RFC 1577. This type of interface is modeled as point-to-point and is used to connect to an IPoA endpoint.

Classical IPoA Configuration Example

Figure 3-1 shows a typical ADSL configuration with IPoA used as an access protocol between a customer premise and an Internet Service Provider (ISP). In this example, the ADSL interface is configured using Classical IPoA.

Figure 3-1 Typical ADSL network configuration



Example 3-1 sets up a Classical IPoA encapsulation on interface ads10. In this example:

- A Sangoma S518 ADSL network interface card (NIC) is connected to the interface.

Tip: Where public IP addresses would normally be used, the example uses RFC 3330 "TEST-NET" IP addresses (192.0.2.0/24)

- The interface has one PVC. The PVC identifier is automatically detected.
- The local IP address is 192.0.2.65, on a network with a prefix length of 24. This is in the public IP range, since this interface will connect over the wide area network.
- The IP address of the far end is 192.0.2.70. This is on the same network as the local interface.

To create and configure this ADSL interface, perform the following steps in configuration mode:

Example 3-1 Creating and configuring an ADSL interface for Classical IPoA encapsulation

Step	Command
Specify that the system should auto-detect an identifier for the pvc.	vyatta@R1# set interfaces adsl adsl0 pvc auto [edit]
Set the line encapsulation to Classical IPoA.	vyatta@R1# set interfaces adsl adsl0 pvc auto classical-ipoa [edit]
Assign the local IP address to the interface.	vyatta@R1# set interfaces adsl adsl0 pvc auto classical-ipoa local-address 192.0.2.65 [edit]
Set the network mask (prefix length) for the interface.	vyatta@R1# set interfaces adsl adsl0 pvc auto classical-ipoa prefix-length 24 [edit]
Set the IP address of the far end of the connection.	vyatta@R1# set interfaces adsl adsl0 pvc auto classical-ipoa remote-address 192.0.2.70 [edit]
Commit the configuration.	vyatta@R1# commit [edit]
View the configuration.	vyatta@R1# show interfaces adsl adsl0 pvc auto { classical-ipoa { local-address 192.0.2.65 prefix-length 24 remote-address 192.0.2.70 } }

Classical IPoA Commands

This chapter contains the following commands.

Configuration Commands

<code>interfaces adsl <adslx> pvc <pvc-id> classical-ipoa</code>	Specifies RFC 1577 Classical IPoA encapsulation for a PVC on an ADSL interface.
<code>interfaces adsl <adslx> pvc <pvc-id> classical-ipoa local-address <ipv4></code>	Assign an IP address to a PVC with RFC 1577 Classical IPoA encapsulation on an ADSL interface.
<code>interfaces adsl <adslx> pvc <pvc-id> classical-ipoa mtu <mtu></code>	Specify the Maximum Transmit Unit (MTU) size for a PVC with RFC 1577 Classical IPoA encapsulation on an ADSL interface.
<code>interfaces adsl <adslx> pvc <pvc-id> classical-ipoa prefix-length <prefix></code>	Specifies the prefix defining the network served by a PVC with RFC 1577 Classical IPoA encapsulation on an ADSL interface.
<code>interfaces adsl <adslx> pvc <pvc-id> classical-ipoa remote-address <ipv4></code>	Sets the IP address of the remote end of a PVC with RFC 1577 Classical IPoA encapsulation on an ADSL interface.

Operational Commands

<code>clear interfaces connection <adslx></code>	Brings a Classical IPoA-encapsulated DSL interface down then up.
<code>connect interface <adslx></code>	Brings a Classical IPoA-encapsulated DSL interface up.
<code>disconnect interface <adslx></code>	Brings a Classical IPoA-encapsulated DSL interface down.

Commands for using other system features with Classical IPoA-encapsulated interfaces can be found in the following locations.

Related Commands Documented Elsewhere

Serial interfaces	Commands for clearing and configuring serial interfaces and displaying serial interface information are described in the <i>Vyatta WAN Interfaces Reference Guide</i> .
Firewall	Commands for configuring firewall on Classical-IPoA encapsulated interfaces are described in the <i>Vyatta Firewall Reference Guide</i> .
OSPF	Commands for configuring the Open Shortest Path First routing protocol on Classical-IPoA encapsulated interfaces are described in the <i>Vyatta OSPF Reference Guide</i> .
RIP	Commands for configuring the Routing Information Protocol on Classical-IPoAencapsulated interfaces are described in the <i>Vyatta RIP Reference Guide</i> .

QoS

Commands for configuring quality of service on Classical-IPoA encapsulated interfaces are described in the *Vyatta QoS Reference Guide*.

clear interfaces connection <adslx>

Brings a Classical IPoA-encapsulated DSL interface down then up.

Syntax

clear interfaces connection *adslx*

Command Mode

Operational mode.

Parameters

<i>adslx</i>	Mandatory. The interface to be operationally brought down, then up. The interface is the name of a Classical IPoA-encapsulated DSL interface; that is the interface name can be adslx .
--------------	--

Default

None.

Usage Guidelines

Use this command to operationally bring a Classical IP over Asynchronous Transfer Mode (IPoA) interface down and then up.

connect interface <adslx>

Brings a Classical IPoA–encapsulated DSL interface up.

Syntax

```
connect interface adslx
```

Command Mode

Operational mode.

Parameters

<i>adslx</i>	Mandatory. The name of the interface. This is the name of a Classical IPoA-encapsulated DSL interface; that is the interface name can be adslx .
--------------	---

Default

None.

Usage Guidelines

Use this command to operationally bring a Classical IP over Asynchronous Transfer Mode (IPoA) interface up.

disconnect interface <adslx>

Brings a Classical IPoA-encapsulated DSL interface down.

Syntax

```
disconnect interface adslx
```

Command Mode

Operational mode.

Parameters

<i>adslx</i>	Mandatory. The name of the interface. This is the name of a Classical IPoA-encapsulated DSL interface; that is the interface name can be adslx .
--------------	---

Default

None.

Usage Guidelines

Use this command to operationally bring a Classical IP over Asynchronous Transfer Mode (IPoA) DSL interface down.

interfaces adsl <adslx> pvc <pvc-id> classical-ipoa

Specifies RFC 1577 Classical IPoA encapsulation for a PVC on an ADSL interface.

Syntax

```
set interfaces adsl adslx pvc pvc-id classical-ipoa
delete interfaces adsl adslx pvc pvc-id classical-ipoa
show interfaces adsl adslx pvc pvc-id classical-ipoa
```

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  adsl adslx {
    pvc [0-255/0-65535|auto] {
      classical-ipoa {}
    }
  }
}
```

Parameters

<i>adslx</i>	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be adsl0 to adslx , depending on what physical ADSL ports are actually available on the system.
<i>pvc-id</i>	Mandatory. The identifier for the PVC. It can either be the <i>vpi/vci</i> pair or the keyword auto , where <i>vpi</i> is a Virtual Path Index from 0 to 255, <i>vci</i> is a Virtual Circuit Index from 0 to 65535, and auto directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.

Default

None.

Usage Guidelines

Use this command to specify Classical IP over Asynchronous Transfer Mode (IPoA) encapsulation for a PVC on an ADSL interface as defined in RFC 1577. This type of interface is modeled as point-to-point.

Use the **set** form of this command to specify Classical IPoA encapsulation.

Use the **delete** form of this command to remove all configuration for Classical IPoA encapsulation.

Use the **show** form of this command to view Classical IPoA configuration.

interfaces adsl <adslx> pvc <pvc-id> classical-ipoa local-address <ipv4>

Assign an IP address to a PVC with RFC 1577 Classical IPoA encapsulation on an ADSL interface.

Syntax

```
set interfaces adsl adslx pvc pvc-id classical-ipoa local-address ipv4
delete interfaces adsl adslx pvc pvc-id classical-ipoa local-address
show interfaces adsl adslx pvc pvc-id classical-ipoa local-address
```

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  adsl adslx {
    pvc [0-255/0-65535|auto] {
      classical-ipoa {
        local-address ipv4
      }
    }
  }
}
```

Parameters

<i>adslx</i>	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be adsl0 to adslx , depending on what physical ADSL ports are actually available on the system.
<i>pvc-id</i>	Mandatory. The identifier for the PVC. It can either be the <i>vpi/vci</i> pair or the keyword auto , where <i>vpi</i> is a Virtual Path Index from 0 to 255, <i>vci</i> is a Virtual Circuit Index from from 0 to 65535, and auto directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
<i>ipv4</i>	Mandatory. The IPv4 address for this interface.

Default

None.

Usage Guidelines

Use this command to specify an IP address for a PVC with Classical IP over Asynchronous Transfer Mode (IPoA) encapsulation on an ADSL interface.

Use the **set** form of this command to set the IP address for an IPoA-encapsulated ADSL interface.

Use the **delete** form of this command to remove IP address configuration for an IPoA-encapsulated ADSL interface.

Use the **show** form of this command to view IP address configuration for an IPoA-encapsulated ADSL interface.

interfaces adsl <adslx> pvc <pvc-id> classical-ipoa mtu <mtu>

Specify the Maximum Transmit Unit (MTU) size for a PVC with RFC 1577 Classical IPoA encapsulation on an ADSL interface.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **classical-ipoa mtu** *mtu*

delete interfaces adsl *adslx* **pvc** *pvc-id* **classical-ipoa mtu**

show interfaces adsl *adslx* **pvc** *pvc-id* **classical-ipoa mtu**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  adsl adslx {
    pvc [0-255/0-65535|auto] {
      classical-ipoa {
        mtu 8-8188
      }
    }
  }
}
```

Parameters

<i>adslx</i>	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be adsl0 to adslx , depending on what physical ADSL ports are actually available on the system.
<i>pvc-id</i>	Mandatory. The identifier for the PVC. It can either be the <i>vpi/vci</i> pair or the keyword auto , where <i>vpi</i> is a Virtual Path Index from 0 to 255, <i>vci</i> is a Virtual Circuit Index from from 0 to 65535, and auto directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
<i>mtu</i>	The maximum packet size that the interface will send. The range is 8 to 8188.

Default

The default MTU is 1500.

Usage Guidelines

Use this command to specify the Maximum Transmit Unit for an ADSL interface encapsulated with RFC 1577 Classical IP over Asynchronous Transfer Mode (IPoA). This is the maximum packet size the interface will send.

Use the **set** form of this command to specify the MTU.

Use the **delete** form of this command to restore the default MTU.

Use the **show** form of this command to view MTU configuration.

interfaces adsl <adslx> pvc <pvc-id> classical-ipoa prefix-length <prefix>

Specifies the prefix defining the network served by a PVC with RFC 1577 Classical IPoA encapsulation on an ADSL interface.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **classical-ipoa prefix-length** *prefix*

delete interfaces adsl *adslx* **pvc** *pvc-id* **classical-ipoa prefix-length**

show interfaces adsl *adslx* **pvc** *pvc-id* **classical-ipoa prefix-length**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  adsl adslx {
    pvc [0-255/0-65535|auto] {
      classical-ipoa {
        prefix-length 0-32
      }
    }
  }
}
```

Parameters

<i>adslx</i>	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be adsl0 to adslx , depending on what physical ADSL ports are actually available on the system.
<i>pvc-id</i>	Mandatory. The identifier for the PVC. It can either be the <i>vpi/vci</i> pair or the keyword auto , where <i>vpi</i> is a Virtual Path Index from 0 to 255, <i>vci</i> is a Virtual Circuit Index from from 0 to 65535, and auto directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
<i>prefix</i>	Mandatory. The prefix defining the network served by this interface. The range is 0 to 32.

Default

None.

Usage Guidelines

Use this command to specify the prefix defining the network served by an ADSL interface encapsulated with RFC 1577 Classical IP over Asynchronous Transfer Mode (IPoA).

Use the **set** form of this command to specify the network prefix.

Use the **delete** form of this command to remove network prefix configuration.

Use the **show** form of this command to view network prefix configuration.

interfaces adsl <adslx> pvc <pvc-id> classical-ipoa remote-address <ipv4>

Sets the IP address of the remote end of a PVC with RFC 1577 Classical IPoA encapsulation on an ADSL interface.

Syntax

```
set interfaces adsl adslx pvc pvc-id classical-ipoa remote-address ipv4
delete interfaces adsl adslx pvc pvc-id classical-ipoa remote-address
show interfaces adsl adslx pvc pvc-id classical-ipoa remote-address
```

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  adsl adslx {
    pvc [0-255/0-65535|auto] {
      classical-ipoa {
        remote-address ipv4
      }
    }
  }
}
```

Parameters

<i>adslx</i>	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be adsl0 to adslx , depending on what physical ADSL ports are actually available on the system.
<i>pvc-id</i>	Mandatory. The identifier for the PVC. It can either be the <i>vpi/vci</i> pair or the keyword auto , where <i>vpi</i> is a Virtual Path Index from 0 to 255, <i>vci</i> is a Virtual Circuit Index from from 0 to 65535, and auto directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
<i>ipv4</i>	Mandatory. The IP address of the remote end of the Classical IPoA link. Only one remote address can be specified.

Default

If not set, the remote address is negotiated.

Usage Guidelines

Use this command to set the IP address of the remote endpoint of an RFC 1577 Classical IP over Asynchronous Transfer Mode (IPoA) connection.

Use the **set** form of this command to specify the remote address.

Use the **delete** form of this command to remove the remote address.

Use the **show** form of this command to display remote address configuration.

Chapter 4: Bridged Ethernet

This chapter describes the commands for setting up Bridged Ethernet encapsulation (RFC 1483) on ADSL interfaces on the Vyatta system.

This chapter presents the following topics:

- Bridged Ethernet Configuration
- Bridged Ethernet Commands

Bridged Ethernet Configuration

This section presents the following topics:

- Bridged Ethernet Overview
- Bridged Ethernet Configuration Example

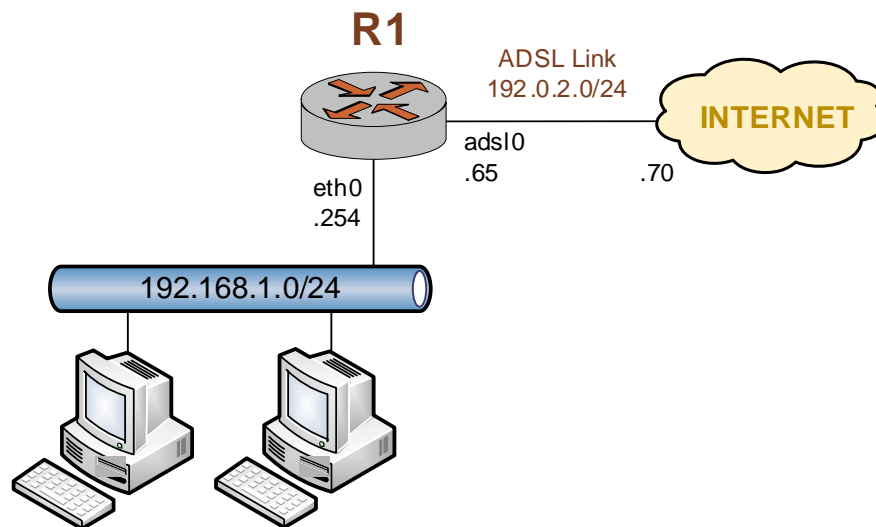
Bridged Ethernet Overview

Bridged Ethernet encapsulation for a permanent virtual circuit (PVC) on an ADSL interface is defined in RFC 1483. This type of interface is modeled as a virtual Ethernet.

Bridged Ethernet Configuration Example

Figure 4-1 shows a typical ADSL configuration with Bridged Ethernet used as an access protocol between a customer premise and an Internet Service Provider (ISP). In this example, the ADSL interface is configured using Bridged Ethernet.

Figure 4-1 Typical ADSL network configuration



Example 4-1 sets up a Bridged Ethernet encapsulation on interface ads10. In this example:

- A Sangoma S518 ADSL network interface card (NIC) is connected to the interface.
- The interface has one PVC (0/35).
- The local IP address is 192.0.2.65, on a network with a prefix length of 24. This is in the public IP range, since this interface will connect over the wide area network.

Tip: Where public IP addresses would normally be used, the example uses RFC 3330 "TEST-NET" IP addresses (192.0.2.0/24)

To create and configure this ADSL interface, perform the following steps in configuration mode:

Example 4-1 Creating and configuring an ADSL interface for Bridged Ethernet encapsulation

Step	Command
Specify that the system should use VPI/VCI of 0/35 (typical for most ISPs)	vyatta@R1# set interfaces ads1 ads10 pvc 0/35 [edit]
Set the line encapsulation to Bridged Ethernet.	vyatta@R1# set interfaces ads1 ads10 pvc 0/35 bridged-ethernet [edit]

Example 4-1 Creating and configuring an ADSL interface for Bridged Ethernet encapsulation

Assign the local IP address to the interface.

```
vyatta@R1# set interfaces adsl ads10 pvc 0/35
bridged-ethernet local-address 192.0.2.65
[edit]
```

Set the network mask (prefix length) for the interface.

```
vyatta@R1# set interfaces adsl ads10 pvc 0/35
bridged-ethernet prefix-length 24
[edit]
```

Commit the configuration.

```
vyatta@R1# commit
[edit]
```

View the configuration.

```
vyatta@R1# show interfaces adsl ads10
pvc 0/35 {
    bridged-ethernet {
        local-address 192.0.2.65
        prefix-length 24
    }
}
[edit]
vyatta@R1#
```

Bridged Ethernet Commands

This chapter contains the following commands.

Configuration Commands

interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet	Specifies RFC 1483 Bridged Ethernet encapsulation for a PVC on an ADSL interface.
interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet local-address <ipv4>	Assign an IP address to a PVC with RFC 1483 Bridged Ethernet encapsulation on an ADSL interface.
interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet mtu <mtu>	Specify the Maximum Transmit Unit (MTU) size for a PVC with RFC 1483 Bridged Ethernet encapsulation on an ADSL interface.
interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet prefix-length <prefix>	Specifies the prefix defining the network served by a PVC with RFC 1483 Bridged Ethernet encapsulation on an ADSL interface.
interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet remote-address <ipv4>	Sets the IP address of the remote end of a PVC with RFC 1483 Bridged Ethernet encapsulation on an ADSL interface.

Commands for using other system features with Bridged Ethernet–encapsulated interfaces can be found in the following locations.

Related Commands Documented Elsewhere

Firewall	Commands for configuring firewall on Bridged Ethernet encapsulated interfaces are described in the <i>Vyatta Firewall Reference Guide</i> .
OSPF	Commands for configuring the Open Shortest Path First routing protocol on Bridged Ethernet encapsulated interfaces are described in the <i>Vyatta OSPF Reference Guide</i> .
RIP	Commands for configuring the Routing Information Protocol on Bridged Ethernet encapsulated interfaces are described in the <i>Vyatta RIP Reference Guide</i> .
QoS	Commands for configuring quality of service on Bridged Ethernet encapsulated interfaces are described in the <i>Vyatta QoS Reference Guide</i> .

interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet

Specifies RFC 1483 Bridged Ethernet encapsulation for a PVC on an ADSL interface.

Syntax

```
set interfaces adsl adslx pvc pvc-id bridged-ethernet
delete interfaces adsl adslx pvc pvc-id bridged-ethernet
show interfaces adsl adslx pvc pvc-id bridged-ethernet
```

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  adsl adslx {
    pvc [0-255/0-65535|auto] {
      bridged-ethernet {}
    }
  }
}
```

Parameters

<i>adslx</i>	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be adsl0 to adslx , depending on what physical ADSL ports are actually available on the system.
<i>pvc-id</i>	Mandatory. The identifier for the PVC. It can either be the <i>vpi/vci</i> pair (e.g., 0/35) or the keyword auto , where <i>vpi</i> is a Virtual Path Index from 0 to 255, <i>vci</i> is a Virtual Circuit Index from 0 to 65535, and auto directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.

Default

None.

Usage Guidelines

Use this command to specify Bridged Ethernet encapsulation for a PVC on an ADSL interface as defined in RFC 1483. This type of interface is modeled as a virtual Ethernet.

Use the **set** form of this command to specify Bridged Ethernet encapsulation.

Use the **delete** form of this command to remove all configuration for Bridged Ethernet encapsulation.

Use the **show** form of this command to view Bridged Ethernet configuration.

interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet local-address <ipv4>

Assign an IP address to a PVC with RFC 1483 Bridged Ethernet encapsulation on an ADSL interface.

Syntax

```
set interfaces adsl adslx pvc pvc-id bridged-ethernet local-address ipv4
delete interfaces adsl adslx pvc pvc-id bridged-ethernet local-address
show interfaces adsl adslx pvc pvc-id bridged-ethernet local-address
```

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  adsl adslx {
    pvc [0-255/0-65535|auto] {
      bridged-ethernet {
        local-address ipv4
      }
    }
  }
}
```

Parameters

<i>adslx</i>	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be adsl0 to adslx , depending on what physical ADSL ports are actually available on the system.
<i>pvc-id</i>	Mandatory. The identifier for the PVC. It can either be the <i>vpi/vci</i> pair (e.g., 0/35) or the keyword auto , where <i>vpi</i> is a Virtual Path Index from 0 to 255, <i>vci</i> is a Virtual Circuit Index from from 0 to 65535, and auto directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
<i>ipv4</i>	Mandatory. The IPv4 address for this interface.

Default

None.

Usage Guidelines

Use this command to specify an IP address for a PVC with Bridged Ethernet encapsulation on an ADSL interface.

Use the **set** form of this command to set the IP address for a Bridged Ethernet-encapsulated ADSL interface.

Use the **delete** form of this command to remove IP address configuration for a Bridged Ethernet-encapsulated ADSL interface.

Use the **show** form of this command to view IP address configuration for a Bridged Ethernet-encapsulated ADSL interface.

interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet mtu <mtu>

Specify the Maximum Transmit Unit (MTU) size for a PVC with RFC 1483 Bridged Ethernet encapsulation on an ADSL interface.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **bridged-ethernet mtu** *mtu*

delete interfaces adsl *adslx* **pvc** *pvc-id* **bridged-ethernet mtu**

show interfaces adsl *adslx* **pvc** *pvc-id* **bridged-ethernet mtu**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  adsl adslx {
    pvc [0-255/0-65535|auto] {
      bridged-ethernet {
        mtu 68-1500
      }
    }
  }
}
```

Parameters

<i>adslx</i>	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be adsl0 to adslx , depending on what physical ADSL ports are actually available on the system.
<i>pvc-id</i>	Mandatory. The identifier for the PVC. It can either be the <i>vpi/vci</i> pair (e.g., 0/35) or the keyword auto , where <i>vpi</i> is a Virtual Path Index from 0 to 255, <i>vci</i> is a Virtual Circuit Index from 0 to 65535, and auto directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
<i>mtu</i>	The maximum packet size that the interface will send. The range is 68 to 1500.

Default

The default MTU is 1500.

Usage Guidelines

Use this command to specify the Maximum Transmit Unit for an ADSL interface encapsulated with RFC 1483 Bridged Ethernet. This is the maximum packet size the interface will send.

Use the **set** form of this command to specify the MTU.

Use the **delete** form of this command to restore the default MTU.

Use the **show** form of this command to view MTU configuration.

interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet prefix-length <prefix>

Specifies the prefix defining the network served by a PVC with RFC 1483 Bridged Ethernet encapsulation on an ADSL interface.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **bridged-ethernet prefix-length** *prefix*

delete interfaces adsl *adslx* **pvc** *pvc-id* **bridged-ethernet prefix-length**

show interfaces adsl *adslx* **pvc** *pvc-id* **bridged-ethernet prefix-length**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  adsl adslx {
    pvc [0-255/0-65535|auto] {
      bridged-ethernet {
        prefix-length 0-32
      }
    }
  }
}
```

Parameters

<i>adslx</i>	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be adsl0 to adslx , depending on what physical ADSL ports are actually available on the system.
<i>pvc-id</i>	Mandatory. The identifier for the PVC. It can either be the <i>vpi/vci</i> pair (e.g., 0/35) or the keyword auto , where <i>vpi</i> is a Virtual Path Index from 0 to 255, <i>vci</i> is a Virtual Circuit Index from from 0 to 65535, and auto directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
<i>prefix</i>	Mandatory. The prefix defining the network served by this interface. The range is 0 to 32.

Default

None.

Usage Guidelines

Use this command to specify the prefix defining the network served by an ADSL interface encapsulated with RFC 1483 Bridged Ethernet.

Use the **set** form of this command to specify the network prefix.

Use the **delete** form of this command to remove network prefix configuration.

Use the **show** form of this command to view network prefix configuration.

interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet remote-address <ipv4>

Sets the IP address of the remote end of a PVC with RFC 1483 Bridged Ethernet encapsulation on an ADSL interface.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **bridged-ethernet remote-address** *ipv4*

delete interfaces adsl *adslx* **pvc** *pvc-id* **bridged-ethernet remote-address**

show interfaces adsl *adslx* **pvc** *pvc-id* **bridged-ethernet remote-address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
  adsl adslx {
    pvc [0-255/0-65535|auto] {
      bridged-ethernet {
        remote-address ipv4
      }
    }
  }
}
```

Parameters

<i>adslx</i>	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be adsl0 to adslx , depending on what physical ADSL ports are actually available on the system.
<i>pvc-id</i>	Mandatory. The identifier for the PVC. It can either be the <i>vpi/vci</i> pair (e.g., 0/35) or the keyword auto , where <i>vpi</i> is a Virtual Path Index from 0 to 255, <i>vci</i> is a Virtual Circuit Index from from 0 to 65535, and auto directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
<i>ipv4</i>	Mandatory. The IP address of the remote end of the Bridged Ethernet link. Only one remote address can be specified.

Default

If not set, the remote address is negotiated.

Usage Guidelines

Use this command to set the IP address of the remote endpoint of an RFC 1483 Bridged Ethernet connection.

Use the **set** form of this command to specify the remote address.

Use the **delete** form of this command to remove the remote address.

Use the **show** form of this command to display remote address configuration.

Glossary of Acronyms

ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DLCI	data-link connection identifier
DMI	desktop management interface
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System

DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Output
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP security
IPv4	IP Version 4
IPv6	IP Version 6
ISP	Internet Service Provider
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol

MAC	medium access control
MIB	Management Information Base
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
ND	Neighbor Discovery
NIC	network interface card
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RA	router advertisement
RIB	Routing Information Base
RIP	Routing Information Protocol

RIPng	RIP next generation
RS	router solicitation
Rx	receive
SLAAC	Stateless address auto-configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SSH	Secure Shell
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TCP	Transmission Control Protocol
ToS	Type of Service
Tx	transmit
UDP	User Datagram Protocol
vif	virtual interface
VLAN	virtual LAN
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
