VYATTA, INC. | Vyatta System

Encapsulation and Tunnels

REFERENCE GUIDE

Cisco HDLC
Frame Relay
PPP
PPPoE
PPPoA
Classical IPoA
Bridged Ethernet
Multilink Interfaces
Tunnel Interfaces

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

COPYRIGHT

PROPRIETARY NOTICES

ISSUE DATE: February 2009

DOCUMENT REVISION. VC5 v03

RELEASED WITH: VC5.0.2

PART NO. A0-0113-10-0002

Table of Contents

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.

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 and appendixes:

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.	24
Chapter 3: PPP	This chapter describes commands for configuring and using PPP encapsulation on the Vyatta system. PPP encapsulation is supported on serial interfaces.	53

Chapter 4: PPPoE	This chapter describes the commands for configuring and using PPPoE encapsulation on the Vyatta system. PPPoE encapsulation is supported on ADSL and Ethernet interfaces.	86
Chapter 5: PPPoA	This chapter describes the commands for configuring and using PPPoA encapsulation on the Vyatta system. PPPoA encapsulation is supported on ADSL interfaces.	144
Chapter 6: Classical IPoA	This chapter describes the commands for setting up Classical IPoA encapsulation on the Vyatta system. Classical IPoA encapsulation is supported on ADSL interfaces.	173
Chapter 8: Multilink Interfaces	This chapter describes commands for working with multilink interfaces.	205
Chapter 9: Tunnel Interfaces	This chapter lists the commands for configuring GRE and IP-in-IP routable tunnel interfaces.	238
Glossary of Acronyms		264

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 *Risk of injury. 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 *Risk of loss of service. 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:

`Courier`	Examples, command-line output, and representations of configuration nodes.
`boldface Courier`	In an example, your input: something you type at a command line.
boldface	In-line commands, keywords, and file names .
italics	Arguments and variables, where you supply a value.
<key>	A key on your keyboard. Combinations of keys are joined by plus signs (“+”). An example is <Ctrl>+<Alt>+<Del>.
[*arg1* | *arg2*]	Enumerated options for completing a syntax. An example is [enable | disable].
num1–numN	A inclusive range of numbers. An example is 1–65535, which means 1 through 65535.
arg1..*argN*	A range of enumerated values. An example is eth0..eth3, which means eth0, eth1, eth2, and eth3.
arg [*arg* ...] *arg*,[*arg*,...]	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 *Vyatta Documentation Map*. This guide is posted with every release of Vyatta software and provides a great starting point for finding what you need.

Chapter 1: Cisco HDLC

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

This chapter presents the following topics:

- Cisco HDLC Commands

Cisco HDLC Commands

This chapter contains the following commands.

Configuration Commands	
interfaces serial <wanx> cisco-hdlc	Defines the characteristics of Cisco High-Level Data Link Control encapsulation for a serial interface.
interfaces serial <wanx> cisco-hdlc keepalives	Set the characteristics of keep-alive messages for a Cisco HDLC serial interface.
interfaces serial <wanx> cisco-hdlc mru <mru>	Specify the Maximum Receive Unit (MRU) size for a Cisco HDLC serial interface.
interfaces serial <wanx> cisco-hdlc mtu <mtu>	Specify the Maximum Transmit Unit (MTU) size for a Cisco HDLC serial interface.
interfaces serial <wanx> cisco-hdlc vif 1 address local-address <ipv4>	Assign an IP address to a Cisco HDLC virtual interface.
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.
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.
interfaces serial <wanx> cisco-hdlc vif 1 description <desc>	Specifies a description for a Cisco HDLC virtual interface.
Operational Commands	
clear interfaces serial <wanx> counters cisco-hdlc	Clears counters for Cisco HDLC–encapsulated serial interfaces
show interfaces serial <wanx> 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 *Vyatta WAN Interfaces Reference Guide*.
Firewall	Commands for configuring firewall on serial interfaces are described in the *Vyatta Security Reference Guide*.

OSPF	Commands for configuring the Open Shortest Path First routing protocol on serial interfaces are described in the *Vyatta OSPF Reference Guide*.
RIP	Commands for configuring the Routing Information Protocol on serial interfaces are described in the *Vyatta RIP Reference Guide*.
QoS	Commands for configuring quality of service on serial interfaces are described in the *Vyatta Policy and QoS Reference Guide*.
System interfaces	Commands for showing the physical interfaces available on your system are described in the Vyatta Basic System Reference Guide.
VRRP	Commands for configuring Virtual Router Redundancy Protocol on serial interfaces are described in the *Vyatta Policy and QoS Reference Guide*.

clear interfaces serial <wanx> counters cisco-hdlc

Clears counters for Cisco HDLC–encapsulated serial interfaces

Syntax

clear interfaces serial *wanx* **counters cisco-hdl**]

Command Mode

Operational mode.

Parameters

wanx	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 Perfomance Monitoring counters were flushed.
vyatta@R1>
```

interfaces serial <wanx> cisco-hdlc

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

Syntax

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

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

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

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

wanx	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

wanx	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 mru <mru>

Specify the Maximum Receive Unit (MRU) size for a Cisco HDLC serial interface.

Syntax

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

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

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

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

wanx	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.
mru	The maximum packet size that the interface is willing to receive. The range is 8 to 8188. The default is 1500.

Default

The MRU is 1500.

Usage Guidelines

Use this command to specify the Maximum Receive Unit (MRU). This is the maximum packet size the interface is willing to receive.

Use the **set** form of this command to specify the MRU

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

Use the **show** form of this command to view MRU 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

wanx	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.
mtu	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

wanx	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.
ipv4	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

wanx	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.
prefix	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

wanx	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.
ipv4	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 1description** *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

wanx	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.
desc	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

wanx	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

wanx	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

Frame Relay Commands

This chapter contains the following commands.

Configuration Commands	
interfaces serial <wanx> encapsulation frame-relay	Sets the Frame Relay as the encapsulation type for a serial interface.
interfaces serial <wanx> frame-relay	Defines the characteristics of Frame Relay encapsulation on a serial interface.
interfaces serial <wanx> frame-relay mru <mru>	Specifies the Maximum Receive Unit (MRU) size for a Frame Relay serial interface.
interfaces serial <wanx> frame-relay mtu <mtu>	Specifies the Maximum Transmit Unit (MTU) size for a Frame Relay serial interface.
interfaces serial <wanx> frame-relay signaling <value>	Specifies the signaling variant (LMI type) for a Frame Relay serial interface.
interfaces serial <wanx> frame-relay signaling-options	Specifies the signaling options for a Frame Relay serial interface.
interfaces serial <wanx> frame-relay vif <dlci> address local-address <ipv4>	Sets the IP address for a Frame Relay virtual interface.
interfaces serial <wanx> frame-relay vif <dlci> address prefix-length <prefix>	Specifies the prefix defining the network served by a Frame Relay virtual interface.
interfaces serial <wanx> frame-relay vif <dlci> address remote-address <ipv4>	Sets the IP address for the remote endpoint of a Frame Relay connection.
interfaces serial <wanx> frame-relay vif <dlci> description <desc>	Specifies a description for a Frame Relay virtual interface.
interfaces serial <wanx> frame-relay vif <dlci> pvc rx-inverse-arp	Enables or disables receiving of inverse ARP messages on a Frame Relay virtual interface.
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.
Operational Commands	
clear interfaces serial <wanx> counters frame-relay	Clears counters for Frame Relay–encapsulated serial interfaces
show interfaces serial <wanx> frame-relay	Displays Frame Relay serial interface information.

Commands for using other system features with Frame Relay–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 *Vyatta WAN Interfaces Reference Guide*.
Firewall	Commands for configuring firewall on serial interfaces are described in the *Vyatta Security Reference Guide*.
OSPF	Commands for configuring the Open Shortest Path First routing protocol on serial interfaces are described in the *Vyatta OSPF Reference Guide*.
RIP	Commands for configuring the Routing Information Protocol on serial interfaces are described in the *Vyatta RIP Reference Guide*.
QoS	Commands for configuring quality of service on serial interfaces are described in the *Vyatta Policy and QoS Reference Guide*.
System interfaces	Commands for showing the physical interfaces available on your system are described in the Vyatta Basic System Reference Guide.
VRRP	Commands for configuring Virtual Router Redundancy Protocol on serial interfaces are described in the *Vyatta Policy and QoS Reference Guide*.

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

wanx	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

wanx	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

wanx	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*—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 mru <mru>

Specifies the Maximum Receive Unit (MRU) size for a Frame Relay serial interface.

Syntax

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

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

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

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   serial wan0..wan23 {
      frame-relay {
         mru 8-8188
      }
   }
}
```

Parameters

wanx	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.
mru	The maximum packet size that the interface is willing to receive. The range is 8 to 8188. The default is 1500.

Default

The MRU is 1500.

Usage Guidelines

Use this command to specify the Maximum Receive Unit on a Frame Relay serial interface. This is the maximum packet size the interface is willing to receive.

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

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

Use the **show** form of this command to view MRU 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

wanx	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.
mtu	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

wanx	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.
value	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

wanx	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*	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. The range is 1 to 255. The default is 6.
n392dte *value*	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. The range is 1 to 100. The default is 6.
n393dte *value*	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. The range is 1 to 10. The default is 4.
t391dte *value*	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. At the interval defined by the **n391dte** option, the DCE will send a full status report instead of just a keep-alive message. The range is 5 to 30. The default is 10.
t392 *value*	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. This value must be greater than the DTE keep-alive interval specified by the **t391dte** parameter. The range is 5 to 30. The default is 16.

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 <dlci> address local-address <ipv4>

Sets the IP address for a Frame Relay virtual interface.

Syntax

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

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

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

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

wanx	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.
dlci	The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. The range is 16 to 991.
ipv4	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

wanx	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.
dlci	The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. The range is 16 to 991.
prefix	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

wanx	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.
dlci	The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. The range is 16 to 991.
ipv4	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 <dlci> description <desc>

Specifies a description for a Frame Relay virtual interface.

Syntax

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

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

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

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

wanx	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.
dlci	The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. The range is 16 to 991.
desc	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 <dlci> 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** *dlci* **pvc rx-inverse-arp** *state*

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

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

wanx	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.
vlan-id	The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. The range is 16 to 991.
state	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

wanx	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.
vlan-id	The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. The range is 16 to 991.
value	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

wanx	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: PPP

This chapter describes commands for configuring and using PPP encapsulation on the Vyatta system. PPP encapsulation is supported on serial interfaces.

This chapter presents the following topics:

- PPP Commands

PPP Commands

This chapter contains the following commands.

Configuration Commands	
interfaces serial <wanx> encapsulation ppp	Sets PPP as the encapsulation type for a serial interface.
interfaces serial <wanx> ppp	Defines the characteristics of PPP encapsulation on a serial interface.
interfaces serial <wanx> ppp authentication	Specifies the authentication parameters for a PPP interface.
interfaces serial <wanx> ppp lcp-echo-failure <value>	Specifies the LCP echo failure threshold for a PPP serial interface.
interfaces serial <wanx> ppp lcp-echo-interval <interval>	Specifies the LCP echo interval for a PPP serial interface.
interfaces serial <wanx> ppp logging <state>	Specifies whether to enable or disable logging of debugging messages for the PPP process.
interfaces serial <wanx> ppp mru <mru>	Specify the Maximum Receive Unit (MRU) size for a PPP serial interface.
interfaces serial <wanx> ppp mtu <mtu>	Specify the Maximum Transmit Unit (MTU) size for a PPP serial interface.
interfaces serial <wanx> ppp multilink <bundle>	Assigns a PPP serial link to a multilink PPP bundle.
interfaces serial <wanx> ppp vif 1 address local-address <ipv4>	Specify the IP address for this virtual interface.
interfaces serial <wanx> ppp vif 1 address prefix-length <prefix>	Specifies the prefix defining the network served by a virtual interface on a PPPP serial interface.
interfaces serial <wanx> ppp vif 1 address remote-address <ipv4>	Specifies the IP address of the remote endpoint on a PPP serial connection.
interfaces serial <wanx> ppp vif 1 description <desc>	Specifies a description for a virtual interface on a PPP serial interface.
Operational Commands	
clear interfaces serial <wanx> counters ppp	Clears counters for PPP-encapsulated serial interfaces
show interfaces serial <wanx> ppp	Displays PPP serial interface information.

Commands for using other system features with PPP–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 *Vyatta WAN Interfaces Reference Guide*.
Firewall	Commands for configuring firewall on serial interfaces are described in the *Vyatta Security Reference Guide*.
OSPF	Commands for configuring the Open Shortest Path First routing protocol on serial interfaces are described in the *Vyatta OSPF Reference Guide*.
RIP	Commands for configuring the Routing Information Protocol on serial interfaces are described in the *Vyatta RIP Reference Guide*.
QoS	Commands for configuring quality of service on serial interfaces are described in the *Vyatta Policy and QoS Reference Guide*.
System interfaces	Commands for showing the physical interfaces available on your system are described in the Vyatta Basic System Reference Guide.
VRRP	Commands for configuring Virtual Router Redundancy Protocol on serial interfaces are described in the *Vyatta Policy and QoS Reference Guide*.

clear interfaces serial <wanx> counters ppp

Clears counters for PPP-encapsulated serial interfaces

Syntax

clear interfaces serial *wanx* **counters ppp**

Command Mode

Operational mode.

Parameters

wanx	The identifier of a configured serial interface.

Usage Guidelines

Use this command to clear statistics for a Point-to-Point Protocol (PPP) serial interface.

interfaces serial <wanx> encapsulation ppp

Sets PPP as the encapsulation type for a serial interface.

Syntax

set interfaces serial *wanx* **encapsulation ppp**

delete interfaces serial *wanx* **encapsulation**

show interfaces serial *wanx* **encapsulation**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   serial wan0..wan23 {
      encapsulation ppp
   }
}
```

Parameters

wanx	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 Point-to-Point Protocol (PPP) 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> ppp

Defines the characteristics of PPP encapsulation on a serial interface.

Syntax

set interfaces serial *wanx* **ppp**

delete interfaces serial *wanx* **ppp**

show interfaces serial *wanx* **ppp**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   serial wan0..wan23 {
      ppp {
      }
   }
}
```

Parameters

wanx	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 Point-to-Point Protocol (PPP) settings on an interface.

The full identifier of a PPP interface is *int* **ppp vif** *vif*. For example, the full identifier of the PPP vif on wan1 is **wan1 ppp vif 1**. Note that subsequent to initial definition, the notation for referring to this is *int.vif*—that is, **wan1.1**.

PPP connections can be “bundled” to form a multilink PPP connection. To do this, use the **multilink** option to specify the identifier of the multilink bundle to which the connection will belong.

When PPP connections are bundled into a multilink, the settings on the multilink override the settings on the individual PPP link. The exceptions is authentication (authentication settings specified for individual PPP links override authentication settings for the multilink) and MTU/MRU/MRRU.

A transmitted packet may not be larger than the remote device is willing to receive. The actual MTU is the smaller of the configured MTU of the local device and the configured MRU of the remote device; this value is determined by MRU negotiation when the link is established.

The interaction between MTU/MRU in PPP links and MTU/MRRU in a multilink bundle is as follows:

- If MTU is unconfigured in both the member PPP link and the multilink bundle, the default for member links is used.
- If MTU is set in member links but not in the multilink bundle, the configured value for member links is used. These must match for every PPP link in the bundle.
- If MTU is set in the multilink bundle, it overrides any value (default or configured) for member links.
- MRRU (for the multilink bundle) and MRU (for member links) are configured independently and used separately during MRU negotiation. If neither is set, the MRU default value is used for MRU and the MRRU default value is used for MRRU.

LCP echo is a heartbeat-like mechanism for determining the operational status of a peer. This feature can be used to terminate a connection after the physical connection has been broken (for example, if the modem has hung up) in situations where no hardware modem control lines are available.

Use the **set** form of this command to define Point-to-Point Protocol (PPP) settings on an interface.

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

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

interfaces serial <wanx> ppp authentication

Specifies the authentication parameters for a PPP interface.

Syntax

set interfaces serial *wanx* **ppp authentication** [**password** *password* | **peer-password** *password* | **peer-system-name** *name* | **peer-user-id** *user-id* | **refuse-type** *type* | **system-name** *name* | **type** *type* | **user-id** *user-id*]

delete interfaces serial *wanx* **ppp authentication**

show interfaces serial *wanx* **ppp authentication**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   serial wan0..wan23 {
      ppp {
         authentication {
            password text
            peer-password text
            peer-system-name text
            peer-user-id text
            refuse-type [none|chap|pap|papchap|mschap|mschap-v2|
               eap]
            system-name text
            type [none|chap|pap|papchap|mschap|mschap-v2|eap|
               any]
            user-id text
         }
      }
   }
}
```

Parameters

wanx	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.

password *password*	Optional. Sets the password this system will use when authenticating itself to a peer.
peer-password *password*	Optional. Sets the password this system will accept from a peer.
peer-system-name *name*	Optional. The system name this system will accept from a peer.
peer-user-id *user-id*	Optional. The user ID this system will accept from a peer.
refuse-type *type*	Defines authentication types that will be refused during authentication negotiations. Used when the Vyatta system is acting as the client side of the communication. **none**: Does not refuse any type of authentication; that is, the system will authenticate to the peer any type of authentication requested, including not using authentication. **chap**: Refuses CHAP authentication if offered by the remote peer. **pap**: Refuses PAP authentication if offered by the remote peer. **papchap**: Refuses PAP or CHAP authentication if offered by the remote peer. **mschap**: Refuses MS-CHAP authentication if offered by the remote peer. **mschap-v2**: Refuses MS-CHAP v2 authentication if offered by the remote peer. **eap**: Refuses EAP authentication if offered by the remote peer. The default is none.
system-name *name*	Optional. The system name this system will use when authenticating itself to a peer.

type *type*	Optional. Sets the authentication required from the remote peer. Used when the Vyatta system is acting as the server side of the communication. Supported values are as follows: **none**: The remote peer is not required to authenticate itself. **chap**: The remote peer must authenticate using the Challenge Handshake Authentication Protocol (CHAP), as defined in RFC 1994. **pap**: The remote peer must authenticate using the Password Authentication Protocol (PAP). The client authenticates itself by sending a user ID and a password to the server, which the server compares to the password in its internal database. **papchap**: The remote peer must authenticate using either PAP or CHAP as the authentication method. **mschap**: The remote peer must authenticate using the Microsoft Challenge Handshake Authentication Protocol (MS-CHAP), which is the Microsoft version of CHAP and is an extension to RFC 1994. **mschap-v2**: The remote peer must authenticate using version 2 of MS-CHAP. **eap**: The remote peer must authenticate using Extensible Authentication Protocol (EAP), which is an authentication framework frequently used in mobile networks and point-to-point connections. **any**: The peer is required to authenticate itself (that is, **none** is refused), but any supported method of authentication offered by the remote peer is accepted. The default is none.
user-id *user-id*	Optional. The user ID this system will use when authenticating itself to a peer.

Default

None.

Usage Guidelines

Use this command to set the authentication parameters for a Point-to-Point protocol (PPP) serial interface. These authentication requirements must be satisfied before network packets are sent or received.

Use the **set** form of this command to set the authentication parameters.

Use the **delete** form of this command to remove authentication configuration or restore default information.

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

interfaces serial <wanx> ppp lcp-echo-failure <value>

Specifies the LCP echo failure threshold for a PPP serial interface.

Syntax

set interfaces serial *wanx* **ppp lcp-echo-failure** *value*

delete interfaces serial *wanx* **ppp lcp-echo-failure**

show interfaces serial *wanx* **ppp lcp-echo-failure**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    serial wan0..wan23 {
        ppp {
            lcp-echo-failure u32
        }
    }
}
```

Parameters

wanx	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.
value	Optional. Sets the LCP echo failure threshold. The failure threshold is the maximum number of LCP echo-requests that can be sent without receiving a valid LCP echo-reply. If this threshold is met, the peer is considered to be dead and the connection is terminated. The default is 3. If this parameter is set, the **lcp-echo-interval** parameter must also be set.

Default

A maximum of 3 LCP echo-requests can be sent without receiving a valid LCP echo-reply.

Usage Guidelines

Use this command to specify the LCP echo failure threshold for a Point-to-Point Protocol (PPP) serial interface.

Use the **set** form of this command to set the LCP echo failure threshold.

Use the **delete** form of this command to restore the default LCP echo failure threshold configuration.

Use the **show** form of this command to view LCP echo failure threshold configuration.

interfaces serial <wanx> ppp lcp-echo-interval <interval>

Specifies the LCP echo interval for a PPP serial interface.

Syntax

set interfaces serial *wanx* **ppp lcp-echo-interval** *value*

delete interfaces serial *wanx* **ppp lcp-echo-interval**

show interfaces serial *wanx* **ppp lcp-echo-interval**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   serial wan0..wan23 {
      ppp {
         lcp-echo-interval u32
      }
   }
}
```

Parameters

wanx	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.
interval	Optional. Sets the LCP echo interval, in seconds. This is the number of seconds between LCP echo-requests. LCP echoes are used to determine whether the connection is still operational. The default is 3. Specifying a low value for this parameter allows fast detection of failed links. The value set for this parameter must match the value set on the peer.

Default

LCP echo-requests are sent at 3-second intervals.

Usage Guidelines

Use this command to specify the LCP echo interval for a Point-to-Point Protocol (PPP serial interface.

Use the **set** form of this command to set the LCP echo interval.

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

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

interfaces serial <wanx> ppp logging <state>

Specifies whether to enable or disable logging of debugging messages for the PPP process.

Syntax

set interfaces serial *wanx* **ppp logging** *state*

delete interfaces serial *wanx* **ppp logging**

show interfaces serial *wanx* **ppp logging**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    serial wan0..wan23 {
        ppp {
            authentication {
                logging [on | off]
            }
        }
    }
}
```

Parameters

wanx	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.
state	Enables logging of debugging messages for the PPP process. Supported values are as follows: **on**: Enables debugging for PPP connections. Trace-level messages are sent from the PPP process to the system log. **off**: Disables debugging for PPP connections. Note that logging creates additional system load and may degrade performance.

Default

Logging of debugging messages is disabled.

Usage Guidelines

Use this command to enable or disable logging of debugging messages for the Point-to-Point protocol (PPP) process.

Use the **set** form of this command to specify whether to enable or disable debugging on a PPP serial interface.

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

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

interfaces serial <wanx> ppp mru <mru>

Specify the Maximum Receive Unit (MRU) size for a PPP serial interface.

Syntax

set interfaces serial *wanx* **ppp mru** *mru*

delete interfaces serial *wanx* **ppp mru**

show interfaces serial *wanx* **ppp mru**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    serial wan0..wan23 {
        ppp {
            mru 8-8188
        }
    }
}
```

Parameters

wanx	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.
mru	The maximum packet size that the interface is willing to receive. The range is 8 to 8188. The default is 1500.

Default

The default is 1500.

Usage Guidelines

Use this command to specify the Maximum Receive Unit (MRU) for a Point-to-Point Protocol (PPP) serial interface. This is the maximum packet size the interface is willing to receive.

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

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

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

interfaces serial <wanx> ppp mtu <mtu>

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

Syntax

set interfaces serial *wanx* **ppp mtu** *mtu*

delete interfaces serial *wanx* **ppp mtu**

show interfaces serial *wanx* **ppp mtu**

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

wanx	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.
mtu	The maximum packet size that the interface will send. The range is 8 to 8188. The default is 1500.

Default

The default is 1500.

Usage Guidelines

Use this command to specify the Maximum Transmit Unit (MTU) for a Point-to-Point Protocol (PPP) 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 value.

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

interfaces serial <wanx> ppp multilink <bundle>

Assigns a PPP serial link to a multilink PPP bundle.

Syntax

set interfaces serial *wanx* **ppp multilink** *bundle*

delete interfaces serial *wanx* **ppp multilink**

show interfaces serial *wanx* **ppp multilink**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    serial wan0..wan23 {
        ppp {
            multilink ml0..ml23
        }
    }
}
```

Parameters

wanx	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.
bundle	The multilink bundle to which to assign this PPP link. The multilink interface must already be defined.

Default

None.

Usage Guidelines

Use this command to assign a Point-to-Point Protocol (PPP) link to a multilink PPP (MLPPP) bundle. For information about defining MLPPP interfaces, see “Chapter 8: Multilink Interfaces.”

All options defined on the multilink interface override those specified for an individual link, except for authentication.

Use the **set** form of this command to assign this PPP link to the specified multilink bundle.

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

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

interfaces serial <wanx> ppp vif 1 address local-address <ipv4>

Specify the IP address for this virtual interface.

Syntax

set interfaces serial *wanx* **ppp vif 1 address local-address** *ipv4*

delete interfaces serial *wanx* **ppp vif 1 address local-address**

show interfaces serial *wanx* **ppp vif 1 address local-address**

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

wanx	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 PPP interfaces, and the identifier must be **1**.
ipv4	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 virtual interface on a Point-to-Point Protocol (PPP) serial 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> ppp vif 1 address prefix-length <prefix>

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

Syntax

set interfaces serial *wanx* **ppp vif 1 address prefix-length** *prefix*

delete interfaces serial *wanx* **ppp vif 1 address prefix-length**

show interfaces serial *wanx* **ppp vif 1 address prefix-length**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   serial wan0..wan23 {
      ppp {
         vif 1 {
            address {
               prefix-length u32
            }
         }
      }
   }
}
```

Parameters

wanx	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 PPP interfaces, and the identifier must be **1**.
prefix	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 a virtual interface on a Point-to-Point Protocol (PPP) serial 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> ppp vif 1 address remote-address <ipv4>

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

Syntax

set interfaces serial *wanx* **ppp vif 1 address remote-address** *ipv4*

delete interfaces serial *wanx* **ppp vif 1 address remote-address**

show interfaces serial *wanx* **ppp vif 1 address remote-address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    serial wan0..wan23 {
        ppp {
            vif 1 {
                address {
                    remote-address ipv4
                }
            }
        }
    }
}
```

Parameters

wanx	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 PPP interfaces, and the identifier must be **1**.
ipv4	Mandatory. The IP address of the remote endpoint.

Default

None.

Usage Guidelines

Use this command to specify the IP address of the remote endpoint in a Point-to-Point Protocol link.

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> ppp vif 1 description <desc>

Specifies a description for a virtual interface on a PPP serial interface.

Syntax

set interfaces serial *wanx* **ppp vif 1 description** *desc*

delete interfaces serial *wanx* **ppp vif 1 description**

show interfaces serial *wanx* **ppp vif 1 description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   serial wan0..wan23 {
      ppp {
         vif 1 {
            description text
         }
      }
   }
}
```

Parameters

wanx	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 PPP interfaces, and the identifier must be **1**.
desc	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 virtual interface on a Point-to-Point Protocol (PPP) serial interface.

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

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

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

show interfaces serial <wanx> ppp

Displays PPP serial interface information.

Syntax

show interfaces serial *wanx* **ppp**

Command Mode

Operational mode.

Parameters

wanx	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

Information is shown for all available serial interfaces.

Usage Guidelines

Use this command to view the operational status of a serial interface.

Examples

Example 3-1 shows the output for **show interfaces serial** *wanx* **ppp**.

Example 3-1 “show interfaces serial *wanx* ppp”

```
vyatta@ppp> show interfaces serial wan0 ppp
   ----------------------------------
      wan0: ROUTER UP TIME
   ----------------------------------
    Router UP Time:  14 minute(s),  6 seconds

PPP data:
IN.BYTES  :          0
IN.PACK   :          0
IN.VJCOMP :          0
```

```
   IN.VJUNC  :          0
   IN.VJERR  :          0
   OUT.BYTES :          0
   OUT.PACK  :          0
   OUT.VJCOMP:          0
   OUT.VJUNC :          0
   OUT.NON-VJ:          0
```

Chapter 4: PPPoE

This chapter describes the commands for configuring and using PPPoE encapsulation on the Vyatta system. PPPoE encapsulation is supported on ADSL and Ethernet interfaces.

This chapter presents the following topics:

- PPPoE Configuration
- PPPoE Commands

PPPoE Configuration

This section presents the following topics:

- PPPoE Overview
- PPPoE Configuration Example

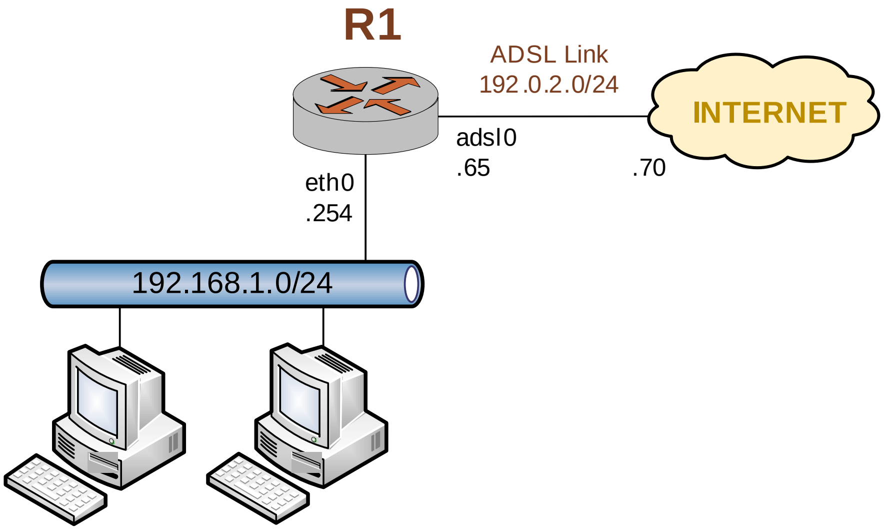
PPPoE Overview

The Point-to-Point Protocol over Ethernet (PPPoE) encapsulation for a PVC on an ADSL interface is defined in RFC 2516. This type of interface is modeled as point-to-point and is used to connect to an PPPoE endpoint.

PPPoE Configuration Example

Figure 4-1 shows a typical ADSL configuration as an access protocol between a customer premesis and an Internet Service Provider. In this example, the ADSL interface is configured using Point-to-Point Protocol over Ethernet (PPPoE). PPPoE links typically include authentication, so a user ID and password are configured in this example.

Figure 4-1 Typical ADSL network configuration



With PPPoE encapsulation the local and remote IP addresses can be automatically negotiated instead of explicitly specified. This is the default: auto-negotiation is performed automatically if the addresses are not specified.

PPPoE encapsulation also allows for "on-demand" connection, in which the interface establishes the PPPoE connection when traffic is sent. On-demand connection is enabled using the **connect-on-demand** option.

Example 4-1 sets up a PPPoE encapsulation on interface adsl0. In this example:

- A Sangoma S518 ADSL NIC is connected to the interface.
- The interface has one PVC. The PVC identifier is automatically detected.
- The PPPoE unit number is 0.
- The local IP address is 192.0.2.65 (prefix-length 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 (prefix-length 24) as this interface.
- The user id is set to "customerA".
- The password is set to "Aremotsuc".

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 PPPoE 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 PPPoE using unit number 0.	vyatta@R1# **set interfaces adsl adsl0 pvc auto pppoe 0** [edit]
Assign the local IP address to the interface.	vyatta@R1# **set interfaces adsl adsl0 pvc auto pppoe 0 local-address 192.0.2.65** [edit]
Set the network mask (prefix length) for the interface.	vyatta@R1# **set interfaces adsl adsl0 pvc auto pppoe 0 prefix-length 24** [edit]
Set the IP address of the far end of the connection.	vyatta@R1# **set interfaces adsl adsl0 pvc auto pppoe 0 remote-address 192.0.2.70** [edit]
Set the user id for the link.	vyatta@R1# **set interfaces adsl adsl0 pvc auto pppoe 0 user-id customerA** [edit]
Set the password for the link.	vyatta@R1# **set interfaces adsl adsl0 pvc auto pppoe 0 password Aremotsuc** [edit]
Commit the configuration.	vyatta@R1# **commit** [edit]

View the configuration.	vyatta@R1# **show interfaces adsl adsl0**
	pvc auto {
	pppoe 0 {
	local-address 192.0.2.65
	prefix-length 24
	remote-address 192.0.2.70
	user-id customerA
	password Aremotsuc
	}
	}
	vyatta@R1#

PPPoE Commands

This chapter contains the following commands.

Configuration Commands	
PPPoE on ADSL	
interfaces adsl <adslx> pvc <pvc-id> pppoe <num>	Enables or disables a PPPoE unit on a PVC with PPPoE encapsulation on an ADSL interface.
interfaces adsl <adslx> pvc <pvc-id> pppoe <num> access-concentrator <name>	Allows you to restrict ADSL PPPoE sessions to one specific access concentrator.
interfaces adsl <adslx> pvc <pvc-id> pppoe <num> connect-on-demand	Enables or disables on-demand PPPoE connection on an ADSL PPPoE unit.
interfaces adsl <adslx> pvc <pvc-id> pppoe <num> default-route <param>	Enables or disables automatically adding a default route when an ADSL PPPoE link is brought up.
interfaces adsl <adslx> pvc <pvc-id> pppoe <num> idle-timeout <timeout>	Specifies the length of time in seconds to wait before disconnecting an idle on-demand ADSL PPPoE session.
interfaces adsl <adslx> pvc <pvc-id> pppoe <num> local-address <ipv4>	Sets the IP address of the local endpoint of an ADSL PPPoE link.
interfaces adsl <adslx> pvc <pvc-id> pppoe <num> mtu <mtu>	Specifies the MTU for an ADSL PPPoE interface.
interfaces adsl <adslx> pvc <pvc-id> pppoe <num> name-server <param>	Specifies whether an ADSL PPPoE interface should obtain name server entries from the remote peer interface.
interfaces adsl <adslx> pvc <pvc-id> pppoe <num> password <password>	Specifies the password to use to authenticate with a remote ADSL PPPoE endpoint.
interfaces adsl <adslx> pvc <pvc-id> pppoe <num> remote-address <ipv4>	Sets the IP address of the remote end of an ADSL PPPoE link.
interfaces adsl <adslx> pvc <pvc-id> pppoe <num> service-name <name>	Allows an ADSL PPPoE interface to restrict connections to access concentrators by service name.
interfaces adsl <adslx> pvc <pvc-id> pppoe <num> user-id <user-id>	Specifies the user ID to use to authenticate with a remote ADSL PPPoE endpoint.
PPPoE on Ethernet	
interfaces ethernet <ethx> pppoe <num>	Enables or disables a PPPoE unit on an Ethernet interface.
interfaces ethernet <ethx> pppoe <num> access-concentrator <name>	Allows you to restrict Ethernet PPPoE sessions to one specific access concentrator.

interfaces ethernet <ethx> pppoe <num> connect-on-demand	Enables or disables on-demand PPPoE connection on an Ethernet PPPoE unit.
interfaces ethernet <ethx> pppoe <num> default-route <param>	Enables or disables automatically adding a default route when an Ethernet PPPoE link is brought up.
interfaces ethernet <ethx> pppoe <num> idle-timeout <timeout>	Specifies the length of time in seconds to wait before disconnecting an idle on-demand Ethernet PPPoE session.
interfaces ethernet <ethx> pppoe <num> local-address <ipv4>	Sets the IP address of the local endpoint of an Ethernet PPPoE link.
interfaces ethernet <ethx> pppoe <num> mtu <mtu>	Specifies the MTU for an Ethernet PPPoE interface.
interfaces ethernet <ethx> pppoe <num> name-server <param>	Specifies whether an Ethernet PPPoE interface should obtain name server entries from the remote peer interface.
interfaces ethernet <ethx> pppoe <num> password <password>	Specifies the password to use to authenticate with a remote Ethernet PPPoE endpoint.
interfaces ethernet <ethx> pppoe <num> remote-address <ipv4>	Sets the IP address of the remote end of an Ethernet PPPoE link.
interfaces ethernet <ethx> pppoe <num> service-name <name>	Allows an Ethernet PPPoE interface to restrict connections to access concentrators by service name.
interfaces ethernet <ethx> pppoe <num> user-id <user-id>	Specifies the user ID to use to authenticate with a remote Ethernet PPPoE endpoint.
Operational Commands	
clear interfaces connection <pppoex>	Brings a PPPoE-encapsulated DSL interface down then up.
connect interface <pppoex>	Brings a PPPoE-encapsulated DSL interface up.
disconnect interface <pppoex>	Brings a PPPoE-encapsulated DSL interface down.
show interfaces pppoe	Displays information about PPPoE interfaces.

Commands for using other system features with PPPoE–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 *Vyatta WAN Interfaces Reference Guide*.
Firewall	Commands for configuring firewall on PPPoE interfaces are described in the *Vyatta Security Reference Guide*.
OSPF	Commands for configuring the Open Shortest Path First routing protocol on PPPoE interfaces are described in the *Vyatta OSPF Reference Guide*.
RIP	Commands for configuring the Routing Information Protocol on PPPoE interfaces are described in the *Vyatta RIP Reference Guide*.
QoS	Commands for configuring quality of service on PPPoE interfaces are described in the *Vyatta Policy and QoS Reference Guide*.
System interfaces	Commands for showing the physical interfaces available on your system are described in the *Vyatta Basic System Reference Guide*.

clear interfaces connection <pppoex>

Brings a PPPoE-encapsulated DSL interface down then up.

Syntax

clear interfaces connection *pppoex*

Command Mode

Operational mode.

Parameters

pppoex	Mandatory. The interface to be operationally brought down, then up. The interface is the name of a PPPoE- encapsulated DSL interface; that is the interface name is **pppoe***x*.

Default

None.

Usage Guidelines

Use this command to operationally bring a Point-to-Point Protocol over Ethernet (PPPoE) interface down and then up.

connect interface <pppoex>

Brings a PPPoE-encapsulated DSL interface up.

Syntax

connect interface *pppoex*

Command Mode

Operational mode.

Parameters

pppoex	Mandatory. The name of the interface. This is the name of a PPPoE-encapsulated DSL interface; that is the interface name is **pppoe***x*.

Default

None.

Usage Guidelines

Use this command to operationally bring a Point-to-Point Protocol over Ethernet (PPPoE) interface up.

disconnect interface <pppoex>

Brings a PPPoE-encapsulated DSL interface down.

Syntax

disconnect interface *pppoex*

Command Mode

Operational mode.

Parameters

pppoex	Mandatory. The name of the interface. This is the name of a PPPoE-encapsulated DSL interface; that is the interface name is **pppoe***x*.

Default

None.

Usage Guidelines

Use this command to operationally bring a Point-to-Point Protocol over Ethernet (PPPoE), DSL interface down.

interfaces adsl <adslx> pvc <pvc-id> pppoe <num>

Enables or disables a PPPoE unit on a PVC with PPPoE encapsulation on an ADSL interface.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num*

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoe 0-15 {
         }
      }
   }
}
```

Parameters

adslx	Mandatory. The name of the interface. This can be the name of a PPPoA-, PPPoE-, or Classical IPOA- encapsulated DSL interface; that is the interface name can be **pppoa***x*, **pppoe***x*, or **adsl***x*.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.

num	Mandatory. The PPPoE unit number. This number must be unique for a given ADSL interface but need not be globally unique (for example, a PPPoE unit number 3 can be defined on on both adsl0 and adsl2). The PPPoE interface will be named **pppoe***unit* (e.g. pppoe7). The range of values is 0 to 15.The range of values is 0 to 15.

Default

None.

Usage Guidelines

Use this command to configure a Point-to-Point Protocol over Ethernet (PPPoE) unit on a PVC with PPPoE encapsulation on an ADSL interface.

A PPPoE interface comes into being on the system only when the PPPoE session is established. So, a PPPoE interface could be defined but not be “present” on a running system.

Use the **set** form of this command to create the PPPoE unit on an interface.

Use the **delete** form of this command to remove a PPPoE unit from an interface.

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

interfaces adsl <adslx> pvc <pvc-id> pppoe <num> access-concentrator <name>

Allows you to restrict ADSL PPPoE sessions to one specific access concentrator.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **access-concentrator** *name*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **access-concentrator**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **access-concentrator**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoe 0-15 {
            access-concentrator text
         }
      }
   }
}
```

Parameters

adslx	Mandatory. The name of the interface. This can be the name of a PPPoA-, PPPoE-, or Classical IPOA- encapsulated DSL interface; that is the interface name can be **pppoa***x*, **pppoe***x*, or **adsl***x*.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoE unit number. The range of values is 0 to 15.
name	The name of the access concentrator for this PPPoE unit to use exclusively for PPPoE sessions.

Default

None.

Usage Guidelines

Use this command to restrict the Point-to-Point Protocol over Ethernet (PPPoE) sessions of a given ADSL PPPoE unit to one access concentrator.

Normally, when a host issues a PPPoE initiation packet to start the PPPoE discovery process, a number of access concentrators respond with offer packets and the host selects one of the responding access concentrators to request the PPPoE session. This command allows you to forego the discovery process and send PPPoE session requests directly to the specified access concentrator.

Use the **set** form of this command to specify an access concentrator to use for ADSL PPPoE sessions.

Use the **delete** form of this command to remove access concentrator configuration. If no access concentrator is specified, the PPPoE discover process will proceed as outlined in RFC 2516.

Use the **show** form of this command to show access concentrator configuration.

interfaces adsl <adslx> pvc <pvc-id> pppoe <num> connect-on-demand

Enables or disables on-demand PPPoE connection on an ADSL PPPoE unit.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **connect-on-demand**

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **connect-on-demand**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoe 0-15 {
            connect-on-demand
         }
      }
   }
}
```

Parameters

adslx	Mandatory. The name of the interface. This can be the name of a PPPoA-, PPPoE-, or Classical IPOA- encapsulated DSL interface; that is the interface name can be **pppoa***x*, **pppoe***x*, or **adsl***x*.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoE unit number. The range of values is 0 to 15.

Default

On-demand PPPoE connection is disabled.

Usage Guidelines

Use this command to direct the system to establish ADSL Point-to-Point Protocol over Ethernet (PPPoE) connections automatically just when traffic is sent.

When on-demand PPPoE connection is disabled, PPPoE links are created at boot time and remain up. If the link fails for any reason, the system brings the link back up immediately.

When on-demand PPPoE connection is enabled, the PPPoE link is brought up only when IP traffic needs to be sent on the link. If the link fails for any reason, it is brought back up again the next time traffic needs to be sent.

If you configure an on-demand PPPoE connection, you must also configure the idle timeout period, after which an idle PPPoE link will be disconnected. If a non-zero idle timeout period is not configured, the on-demand link will never be disconnected after the first time it is brought up. To configure the idle timeout period, use the **interfaces adsl <adslx> pvc <pvc-id> pppoe <num> idle-timeout <timeout>** command (see page 103).

If you configure an on-demand PPPoE connection, you must also configure remote-address. To configure the remote address, use the **interfaces adsl <adslx> pvc <pvc-id> pppoe <num> remote-address <ipv4>** command (see page 113).

Use the **set** form of this command to enable on-demand PPPoE connections.

Use the **delete** form of this command to disable on-demand PPPoE connections.

Use the **show** form of this command to show PPPoE connection configuration.

interfaces adsl <adslx> pvc <pvc-id> pppoe <num> default-route <param>

Enables or disables automatically adding a default route when an ADSL PPPoE link is brought up.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **default-route** *param*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **default-route**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoe 0-15 {
            default-route [auto|none]
         }
      }
   }
}
```

Parameters

adslx	Mandatory. The name of the interface. This can be the name of a PPPoA-, PPPoE-, or Classical IPOA- encapsulated DSL interface; that is the interface name can be **pppoa***x*, **pppoe***x*, or **adsl***x*.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoE unit number. The range of values is 0 to 15.

param	Mandatory. Specifies whether a default route is automatically added when the PPP link comes up. **auto**: The PPP process automatically adds a default route to the remote end of the link. **none**: No default route is added.

Default

A default route to the remote endpoint is automatically added when the link comes up (i.e. **auto**).

Usage Guidelines

Use this command to specify whether to automatically add a default route pointing to the endpoint of an ADSL Point-to-Point Protocol over Ethernet (PPPoE) link when the link comes up.

The default route is only added if no other default route already exists in the system.

Use the **set** form of this command to enable or disable adding the default route.

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

Use the **show** form of this command to show configuration for the PPPoE unit.

interfaces adsl <adslx> pvc <pvc-id> pppoe <num> idle-timeout <timeout>

Specifies the length of time in seconds to wait before disconnecting an idle on-demand ADSL PPPoE session.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **idle-timeout** *timeout*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **idle-timeout**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **idle-timeout**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoe 0-15 {
            idle-timeout u32
         }
      }
   }
}
```

Parameters

adslx	Mandatory. The name of the interface. This can be the name of a PPPoA-, PPPoE-, or Classical IPOA- encapsulated DSL interface; that is the interface name can be **pppoa***x*, **pppoe***x*, or **adsl***x*.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoE unit number. The range of values is 0 to 15.

timeout	Mandatory. The amount of time, in seconds, after which an idle connection will be closed. The range is 0 to 4294967295, where 0 means the connection is never closed.

Default

Idle connections are never disconnected.

Usage Guidelines

Use this command to set the idle timeout interval to be used with on-demand ADSL Point-to-Point Protocol over Ethernet (PPPoE) connections.

When on-demand PPPoE link connection is enabled, the link is brought up only when traffic is to be sent and is disabled when the link is idle for the interval specified by this command. On-demand PPPoE connection is enabled using the **interfaces adsl <adslx> pvc <pvc-id> pppoe <num> connect-on-demand** command (see page 99).

If this parameter is not set or is set to 0, an on-demand link will not be taken down when it is idle and after the initial establishment of the connection will behave like an ordinary PPPoE link.

Use the **set** form of this command to specify the idle timeout value.

Use the **delete** form of this command to restore default behavior for idle timeout.

Use the **show** form of this command to display idle timeout configuration.

interfaces adsl <adslx> pvc <pvc-id> pppoe <num> local-address <ipv4>

Sets the IP address of the local endpoint of an ADSL PPPoE link.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **local-address** *ipv4*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **local-address**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **local-address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoe 0-15 {
            local-address ipv4
         }
      }
   }
}
```

Parameters

adslx	Mandatory. The name of the interface. This can be the name of a PPPoA-, PPPoE-, or Classical IPOA- encapsulated DSL interface; that is the interface name can be **pppoa***x*, **pppoe***x*, or **adsl***x*.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoE unit number. The range of values is 0 to 15.
ipv4	Mandatory. The IP address of the local end of the PPPoE link. Only one local address can be specified.

Default

None.

Usage Guidelines

Use this command to set the IP address of the local endpoint of an ADSL Point-to-Point Protocol over Ethernet (PPPoE) connection. If not set it will be negotiated.

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

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

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

interfaces adsl <adslx> pvc <pvc-id> pppoe <num> mtu <mtu>

Specifies the MTU for an ADSL PPPoE interface.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **mtu** *mtu*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **mtu**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **mtu**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoe 0-15 {
            mtu 68-1492
         }
      }
   }
}
```

Parameters

adslx	Mandatory. The name of the interface. This can be the name of a PPPoA-, PPPoE-, or Classical IPOA- encapsulated DSL interface; that is the interface name can be **pppoa***x*, **pppoe***x*, or **adsl***x*.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoE unit number. The range of values is 0 to 15.
mtu	Sets the MTU for the PPPoE interface. Packets larger that this value are fragmented. The range is 68 to 1492.

Default

The default MTU is 1492 bytes.

Usage Guidelines

Use this command to set the Maximum Transfer Unit (MTU) of an ADSL Point-to-Point Protocol over Ethernet (PPPoE) unit. Packets larger than the MTU are fragmented.

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

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

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

interfaces adsl <adslx> pvc <pvc-id> pppoe <num> name-server <param>

Specifies whether an ADSL PPPoE interface should obtain name server entries from the remote peer interface.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **name-server** *param*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **name-server**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoe 0-15 {
            name-server [auto|none]
         }
      }
   }
}
```

Parameters

adslx	Mandatory. The name of the interface. This can be the name of a PPPoA-, PPPoE-, or Classical IPOA- encapsulated DSL interface; that is the interface name can be **pppoa***x*, **pppoe***x*, or **adsl***x*.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoE unit number. The range of values is 0 to 15.

param	Mandatory. Specifies whether the local PPPoE endpoint should obtain name server entries from the remote endpoint. Supported values are as follows: **auto**: The endpoint obtains name server entries from its peer. **none**: The endpoint uses the name server(s) configured for the local system.

Default

The interface obtains name server entries from its peer (i.e. **auto**).

Usage Guidelines

Use this command to define how a name server is defined when an ADSL Point-to-Point Protocol over Ethernet (PPPoE) link is brought up.

Use the **set** form of this command to set the way that name server entries are obtained by the PPPoE endpoint.

Use the **delete** form of this command to restore the default behavior for obtaining name server entries.

Use the **show** form of this command to show the PPPoE name server configuration.

interfaces adsl <adslx> pvc <pvc-id> pppoe <num> password <password>

Specifies the password to use to authenticate with a remote ADSL PPPoE endpoint.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **password** *password*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **password**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **password**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoe 0-15 {
            password text
         }
      }
   }
}
```

Parameters

adslx	Mandatory. The name of the interface. This can be the name of a PPPoA-, PPPoE-, or Classical IPOA- encapsulated DSL interface; that is the interface name can be **pppoa***x*, **pppoe***x*, or **adsl***x*.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoE unit number. The range of values is 0 to 15.
password	Mandatory. The password used to authenticate the local endpoint with the remote PPPoE server.

Default

None.

Usage Guidelines

Use this command to set the authentication password for an ADSL Point-to-Point Protocol over Ethernet (PPPoE) endpoint.

Authentication is optional from the system’s point of view; however, most service providers require it.

The password is used in conjunction with the user ID to authenticate the local system to the remote endpoint. The user ID is set by using the **interfaces adsl <adslx> pvc <pvc-id> pppoe <num> user-id <user-id>** command (see page 117). The authentication protocol is determined by the remote endpoint. Use the **set** form of this command to set the password.

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

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

interfaces adsl <adslx> pvc <pvc-id> pppoe <num> remote-address <ipv4>

Sets the IP address of the remote end of an ADSL PPPoE link.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **remote-address** *ipv4*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **remote-address**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **remote-address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoe 0-15 {
            remote-address ipv4
         }
      }
   }
}
```

Parameters

adslx	Mandatory. The name of the interface. This can be the name of a PPPoA-, PPPoE-, or Classical IPOA- encapsulated DSL interface; that is the interface name can be **pppoa***x*, **pppoe***x*, or **adsl***x*.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoE unit number. The range of values is 0 to 15.
ipv4	Mandatory. The IP address of the remote end of the PPPoE link. Only one remote address can be specified.

Default

None.

Usage Guidelines

Use this command to set the IP address of the remote endpoint of an ADSL Point-to-Point Protocol over Ethernet (PPPoE) connection. This address will be negotiated if not set.

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.

interfaces adsl <adslx> pvc <pvc-id> pppoe <num> service-name <name>

Allows an ADSL PPPoE interface to restrict connections to access concentrators by service name.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **service-name** *name*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **service-name**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **service-name**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoe 0-15 {
            service-name text
         }
      }
   }
}
```

Parameters

adslx	Mandatory. The name of the interface. This can be the name of a PPPoA-, PPPoE-, or Classical IPOA- encapsulated DSL interface; that is the interface name can be **pppoa***x*, **pppoe***x*, or **adsl***x*.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoE unit number. The range of values is 0 to 15.

name	Mandatory. A service name. The local endpoint will send session requests only to access concentrators advertising this service name

Default

None.

Usage Guidelines

Use this command to specify a service name by which the local ADSL Point-to-Point Protocol over Ethernet (PPPoE) interface can select access concentrators to connect with. It will connect to any access concentrator if not set.

Use the **set** form of this command to specify a service name.

Use the **delete** form of this command to remove a service name.

Use the **show** form of this command to show service name configuration.

interfaces adsl <adslx> pvc <pvc-id> pppoe <num> user-id <user-id>

Specifies the user ID to use to authenticate with a remote ADSL PPPoE endpoint.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **user-id** *user-id*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **user-id**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **user-id**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoe 0-15 {
            user-id text
         }
      }
   }
}
```

Parameters

adslx	Mandatory. The name of the interface. This can be the name of a PPPoA-, PPPoE-, or Classical IPOA- encapsulated DSL interface; that is the interface name can be **pppoa***x*, **pppoe***x*, or **adsl***x*.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoE unit number. The range of values is 0 to 15.
user-id	Optional. The user ID to be used by the local endpoint to authenticate itself to the remote endpoint.

Default

None.

Usage Guidelines

Use this command to set the user ID for authenticating with a remote ADSL Point-to-Point Protocol over Ethernet (PPPoE) endpoint.

Authentication is optional from the system’s point of view; however, most service providers require it.

The user ID is used in conjunction with the password to authenticate the local system to the remote endpoint. The password is set by using the **interfaces adsl <adslx> pvc <pvc-id> pppoe <num> password <password>** command (see page 111). The authentication protocol is determined by the remote endpoint. Use the **set** form of this command to set the user ID.

Use the **delete** form of this command to remove the user ID.

Use the **show** form of this command to display user ID configuration.

interfaces ethernet <ethx> pppoe <num>

Enables or disables a PPPoE unit on an Ethernet interface.

Syntax

set interfaces ethernet *ethx* **pppoe** *num*

delete interfaces ethernet *ethx* **pppoe** *num*

show interfaces ethernet *ethx* **pppoe** *num*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet [eth0..eth23] {
      pppoe 0-15 {
      }
   }
}
```

Parameters

ethx	Mandatory. The name of a defined Ethernet interface. The range is **eth0** to **eth23**.
num	Mandatory. The PPPoE unit number. This number must be unique for a given Ethernet interface but need not be globally unique (for example, a PPPoE unit number 3 can be defined on on both eth0 and eth2). The PPPoE interface will be named **pppoe***unit* (e.g. **pppoe7**). The range of values is 0 to 15.

Default

None.

Usage Guidelines

Use this command to configure a Point-to-Point Protocol over Ethernet (PPPoE) unit on an Ethernet interface.

A PPPoE interface comes into being on the system only when the PPPoE session is established. So, a PPPoE interface could be defined but not be “present” on a running system.

Use the **set** form of this command to create the PPPoE unit on an interface.

Use the **delete** form of this command to remove a PPPoE unit from an interface.

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

interfaces ethernet <ethx> pppoe <num> access-concentrator <name>

Allows you to restrict Ethernet PPPoE sessions to one specific access concentrator.

Syntax

set interfaces ethernet *ethx* **pppoe** *num* **access-concentrator** *name*

delete interfaces ethernet *ethx* **pppoe** *num* **access-concentrator**

show interfaces ethernet *ethx* **pppoe** *num* **access-concentrator**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet [eth0..eth23] {
      pppoe 0-15 {
         access-concentrator text
      }
   }
}
```

Parameters

ethx	Mandatory. The name of a defined Ethernet interface. The range is **eth0** to **eth23**.
num	Mandatory. The name of a defined PPPoE unit. The range of values is 0 to 15.
name	The name of the access concentrator for this PPPoE unit to use exclusively for PPPoE sessions.

Default

None.

Usage Guidelines

Use this command to restrict the Point-to-Point Protocol over Ethernet (PPPoE) sessions of a given Ethernet PPPoE unit to one access concentrator.

Normally, when a host issues a PPPoE initiation packet to start the PPPoE discovery process, a number of access concentrators respond with offer packets and the host selects one of the responding access concentrators to request the PPPoE session. This command allows you to forego the discovery process and send PPPoE session requests directly to the specified access concentrator.

Use the **set** form of this command to specify an access concentrator to use for PPPoE sessions.

Use the **delete** form of this command to remove access concentrator configuration. If no access concentrator is specified, the PPPoE discover process will proceed as outlined in RFC 2516.

Use the **show** form of this command to show access concentrator configuration.

interfaces ethernet <ethx> pppoe <num> connect-on-demand

Enables or disables on-demand PPPoE connection on an Ethernet PPPoE unit.

Syntax

set interfaces ethernet *ethx* **pppoe** *num* **connect-on-demand**

delete interfaces ethernet *ethx* **pppoe** *num* **connect-on-demand**

show interfaces ethernet *ethx* **pppoe** *num*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet [eth0..eth23] {
      pppoe 0-15 {
         connect-on-demand
      }
   }
}
```

Parameters

ethx	Mandatory. The name of a defined Ethernet interface. The range is **eth0** to **eth23**.
num	Mandatory. The name of a defined PPPoE unit. The range of values is 0 to 15.

Default

On-demand PPPoE connection is disabled.

Usage Guidelines

Use this command to direct the system to establish Point-to-Point Protocol over Ethernet (PPPoE) connections automatically just when traffic is sent.

When on-demand PPPoE connection is disabled, PPPoE links are created at boot time and remain up. If the link fails for any reason, the system brings the link back up immediately.

When on-demand PPPoE connection is enabled, the PPPoE link is brought up only when IP traffic needs to be sent on the link. If the link fails for any reason, it is brought back up again the next time traffic needs to be sent.

If you configure an on-demand PPPoE connection, you must also configure the idle timeout period, after which an idle PPPoE link will be disconnected. If a non-zero idle timeout period is not configured, the on-demand link will never be disconnected after the first time it is brought up. To configure the idle timeout period, use the **interfaces ethernet <ethx> pppoe <num> idle-timeout <timeout>** command (see page 127).

If you configure an on-demand PPPoE connection, you must also configure remote-address. To configure the remote address, use the **interfaces ethernet <ethx> pppoe <num> remote-address <ipv4>** command (see page 137).

Use the **set** form of this command to enable on-demand PPPoE connections.

Use the **delete** form of this command to disable on-demand PPPoE connections.

Use the **show** form of this command to show PPPoE connection configuration.

interfaces ethernet <ethx> pppoe <num> default-route <param>

Enables or disables automatically adding a default route when an Ethernet PPPoE link is brought up.

Syntax

set interfaces ethernet *ethx* **pppoe** *num* **default-route** *param*

delete interfaces ethernet *ethx* **pppoe** *num* **default-route**

show interfaces ethernet *ethx* **pppoe** *num*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet [eth0..eth23] {
      pppoe 0-15 {
         default-route [auto|none]
      }
   }
}
```

Parameters

ethx	Mandatory. The name of a defined Ethernet interface. The range is **eth0** to **eth23**.
num	Mandatory. The name of a defined PPPoE unit. The range of values is 0 to 15.
param	Mandatory. Specifies whether a default route is automatically added when the PPP link comes up. **auto**: The PPP process automatically adds a default route to the remote end of the link. **none**: No default route is added.

Default

A default route to the remote endpoint is automatically added when the link comes up (i.e. **auto**).

Usage Guidelines

Use this command to specify whether to automatically add a default route pointing to the endpoint of the when a Point-to-Point Protocol over Ethernet (PPPoE) link comes up.

The default route is only added if no other default route already exists in the system.

Use the **set** form of this command to enable or disable adding the default route.

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

Use the **show** form of this command to show configuration for the PPPoE unit.

interfaces ethernet <ethx> pppoe <num> idle-timeout <timeout>

Specifies the length of time in seconds to wait before disconnecting an idle on-demand Ethernet PPPoE session.

Syntax

set interfaces ethernet *ethx* **pppoe** *num* **idle-timeout** *timeout*

delete interfaces ethernet *ethx* **pppoe** *num* **idle-timeout**

show interfaces ethernet *ethx* **pppoe** *num* **idle-timeout**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet [eth0..eth23] {
      pppoe 0-15 {
         idle-timeout u32
      }
   }
}
```

Parameters

ethx	Mandatory. The name of a defined Ethernet interface. The range is **eth0** to **eth23**.
num	Mandatory. The name of a defined PPPoE unit. The range of values is 0 to 15.
timeout	Mandatory. The amount of time, in seconds, after which an idle connection will be closed. The range is 0 to 4294967295, where 0 means the connection is never closed.

Default

Idle connections are never disconnected.

Usage Guidelines

Use this command to set the idle timeout interval to be used with on-demand Point-to-Point Protocol over Ethernet (PPPoE) connections.

When on-demand PPPoE link connection is enabled, the link is brought up only when traffic is to be sent and is disabled when the link is idle for the interval specified by this command. On-demand PPPoE connection is enabled using the **interfaces ethernet <ethx> pppoe <num> connect-on-demand** command (see page 123).

If this parameter is not set or is set to 0, an on-demand link will not be taken down when it is idle and after the initial establishment of the connection will behave like an ordinary PPPoE link.

Use the **set** form of this command to specify the idle timeout value.

Use the **delete** form of this command to restore default behavior for idle timeout.

Use the **show** form of this command to display idle timeout configuration.

interfaces ethernet <ethx> pppoe <num> local-address <ipv4>

Sets the IP address of the local endpoint of an Ethernet PPPoE link.

Syntax

set interfaces ethernet *ethx* **pppoe** *num* **local-address** *ipv4*

delete interfaces ethernet *ethx* **pppoe** *num* **local-address**

show interfaces ethernet *ethx* **pppoe** *num* **local-address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet [eth0..eth23] {
      pppoe 0-15 {
         local-address ipv4
      }
   }
}
```

Parameters

ethx	Mandatory. The name of a defined Ethernet interface. The range is **eth0** to **eth23**.
num	Mandatory. The name of a defined PPPoE unit. The range of values is 0 to 15.
ipv4	Mandatory. The IP address of the local end of the PPPoE link. Only one local address can be specified.

Default

None.

Usage Guidelines

Use this command to set the IP address of the local endpoint of a Point-to-Point Protocol over Ethernet (PPPoE) connection. If not set it will be negotiated.

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

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

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

interfaces ethernet <ethx> pppoe <num> mtu <mtu>

Specifies the MTU for an Ethernet PPPoE interface.

Syntax

set interfaces ethernet *ethx* **pppoe** *num* **mtu** *mtu*

delete interfaces ethernet *ethx* **pppoe** *num* **mtu**

show interfaces ethernet *ethx* **pppoe** *num* **mtu**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet [eth0..eth23] {
      pppoe 0-15 {
         mtu 68-1492
      }
   }
}
```

Parameters

ethx	Mandatory. The name of a defined Ethernet interface. The range is **eth0** to **eth23**.
num	Mandatory. The name of a defined PPPoE unit. The range of values is 0 to 15.
mtu	Sets the MTU for the PPPoE interface. Packets larger that this value are fragmented. The range is 68 to 1492.

Default

If not set, the MTU for the PPPoE interface will be set to the MTU for the Ethernet interface minus 8 bytes.

Usage Guidelines

Use this command to set the Maximum Transfer Unit (MTU) of a Point-to-Point Protocol over Ethernet (PPPoE) unit. Packets larger than the MTU are fragmented.

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

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

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

interfaces ethernet <ethx> pppoe <num> name-server <param>

Specifies whether an Ethernet PPPoE interface should obtain name server entries from the remote peer interface.

Syntax

set interfaces ethernet *ethx* **pppoe** *num* **name-server** *param*

delete interfaces ethernet *ethx* **pppoe** *num* **name-server**

show interfaces ethernet *ethx* **pppoe** *num*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet [eth0..eth23] {
      pppoe 0-15 {
         name-server [auto|none]
      }
   }
}
```

Parameters

ethx	Mandatory. The name of a defined Ethernet interface. The range is **eth0** to **eth23**.
num	Mandatory. The name of a defined PPPoE unit. The range of values is 0 to 15.
param	Mandatory. Specifies whether the local PPPoE endpoint should obtain name server entries from the remote endpoint. Supported values are as follows: **auto**: The endpoint obtains name server entries from its peer. **none**: The endpoint uses the name server(s) configured for the local system.

Default

The interface obtains name server entries from its peer.

Usage Guidelines

Use this command to define how a name server is defined when an Point-to-Point Protocol over Ethernet (PPPoE) link is brought up.

Use the **set** form of this command to set the way that name server entries are obtained by the PPPoE endpoint.

Use the **delete** form of this command to restore the default behavior for obtaining name server entries.

Use the **show** form of this command to show the PPPoE name server configuration.

interfaces ethernet <ethx> pppoe <num> password <password>

Specifies the password to use to authenticate with a remote Ethernet PPPoE endpoint.

Syntax

set interfaces ethernet *ethx* **pppoe** *num* **password** *password*

delete interfaces ethernet *ethx* **pppoe** *num* **password**

show interfaces ethernet *ethx* **pppoe** *num* **password**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet [eth0..eth23] {
      pppoe 0-15 {
         password text
      }
   }
}
```

Parameters

ethx	Mandatory. The name of a defined Ethernet interface. The range is **eth0** to **eth23**.
num	Mandatory. The name of a defined PPPoE unit. The range of values is 0 to 15.
password	Mandatory. The password used to authenticate the local endpoint with the remote PPPoE server.

Default

None.

Usage Guidelines

Use this command to set the authentication password for an Point-to-Point Protocol over Ethernet (PPPoE) endpoint.

Authentication is optional from the system’s point of view; however, most service providers require it.

The password is used in conjunction with the user ID to authenticate the local system to the remote endpoint. The user ID is set by using the **interfaces ethernet <ethx> pppoe <num> user-id <user-id>** command (see page 141). The authentication protocol is determined by the remote endpoint. Use the **set** form of this command to set the password.

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

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

interfaces ethernet <ethx> pppoe <num> remote-address <ipv4>

Sets the IP address of the remote end of an Ethernet PPPoE link.

Syntax

set interfaces ethernet *ethx* **pppoe** *num* **remote-address** *ipv4*

delete interfaces ethernet *ethx* **pppoe** *num* **remote-address**

show interfaces ethernet *ethx* **pppoe** *num* **remote-address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet [eth0..eth23] {
      pppoe 0-15 {
         remote-address ipv4
      }
   }
}
```

Parameters

ethx	Mandatory. The name of a defined Ethernet interface. The range is **eth0** to **eth23**.
num	Mandatory. The name of a defined PPPoE unit. The range of values is 0 to 15.
ipv4	Mandatory. The IP address of the remote end of the PPPoE link. Only one remote address can be specified.

Default

None.

Usage Guidelines

Use this command to set the IP address of the remote endpoint of an Point-to-Point Protocol over Ethernet (PPPoE) connection. This address will be negotiated if not set.

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.

interfaces ethernet <ethx> pppoe <num> service-name <name>

Allows an Ethernet PPPoE interface to restrict connections to access concentrators by service name.

Syntax

set interfaces ethernet *ethx* **pppoe** *num* **service-name** *name*

delete interfaces ethernet *ethx* **pppoe** *num* **service-name**

show interfaces ethernet *ethx* **pppoe** *num* **service-name**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet [eth0..eth23] {
      pppoe 0-15 {
         service-name text
      }
   }
}
```

Parameters

ethx	Mandatory. The name of a defined Ethernet interface. The range is **eth0** to **eth23**.
num	Mandatory. The name of a defined PPPoE unit. The range of values is 0 to 15.
name	Mandatory. A service name. The local endpoint will send session requests only to access concentrators advertising this service name

Default

None.

Usage Guidelines

Use this command to specify a service name by which the local Point-to-Point Protocol over Ethernet (PPPoE) interface can select access concentrators to connect with. It will connect to any access concentrator if not set.

Use the **set** form of this command to specify a service name.

Use the **delete** form of this command to remove a service name.

Use the **show** form of this command to show service name configuration.

interfaces ethernet <ethx> pppoe <num> user-id <user-id>

Specifies the user ID to use to authenticate with a remote Ethernet PPPoE endpoint.

Syntax

set interfaces ethernet *ethx* **pppoe** *num* **user-id** *user-id*

delete interfaces ethernet *ethx* **pppoe** *num* **user-id**

show interfaces ethernet *ethx* **pppoe** *num* **user-id**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet [eth0..eth23] {
      pppoe 0-15 {
         user-id text
      }
   }
}
```

Parameters

ethx	Mandatory. The name of a defined Ethernet interface. The range is **eth0** to **eth23**.
num	Mandatory. The name of a defined PPPoE unit. The range of values is 0 to 15.
user-id	Optional. The user ID to be used by the local endpoint to authenticate itself to the remote endpoint.

Default

None.

Usage Guidelines

Use this command to set the user ID for authenticating with a remote Point-to-Point Protocol over Ethernet (PPPoE) endpoint.

Authentication is optional from the system's point of view; however, most service providers require it.

The user ID is used in conjunction with the password to authenticate the local system to the remote endpoint. The password is set by using the **interfaces ethernet <ethx> pppoe <num> password <password>** command (see page 135). The authentication protocol is determined by the remote endpoint. Use the **set** form of this command to set the user ID.

Use the **delete** form of this command to remove the user ID.

Use the **show** form of this command to display user ID configuration.

show interfaces pppoe

Displays information about PPPoE interfaces.

Syntax

show interfaces pppoe [*num*]

Command Mode

Operational mode.

Parameters

num	Displays information for the specified PPPoE unit.

Default

Displays information for all PPPoE interfaces.

Usage Guidelines

Use this command to display Point-to-Point Protocol over Ethernet (PPPoE) interface information.

Chapter 5: PPPoA

This chapter describes the commands for configuring and using PPPoA encapsulation on the Vyatta system. PPPoA encapsulation is supported on ADSL interfaces.

This chapter presents the following topics:

- PPPoA Configuration
- PPPoA Commands

PPPoA Configuration

This section presents the following topics:

- PPPoA Overview
- PPPoA Configuration Example

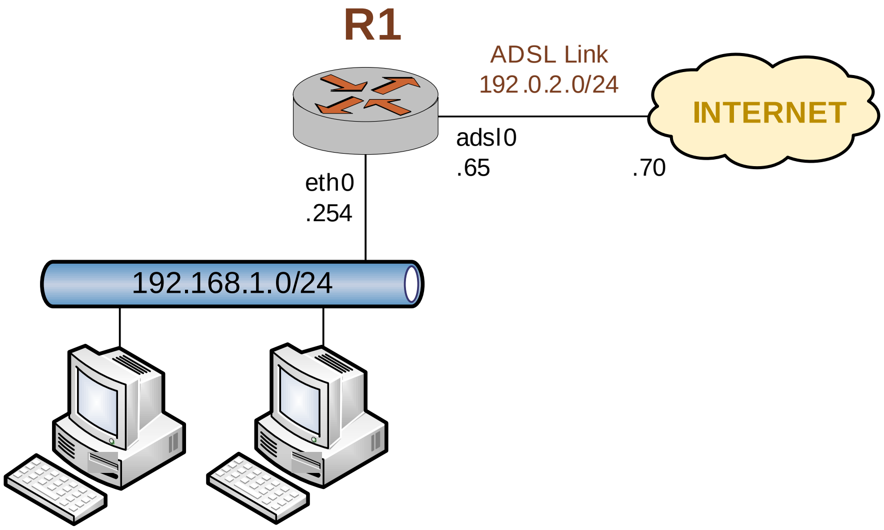
PPPoA Overview

The Point-to-Point Protocol over Asynchronous Transfer Mode (PPPoA) encapsulation for a permanent virtual circuit (PVC) on an ADSL interface is defined in RFC 2364. This type of interface is modeled as point-to-point and is used to connect to a PPPoA endpoint.

PPPoA Configuration Example

Figure 5-1 shows a typical ADSL configuration, where ADSL is used as an access protocol between a customer premise and an Internet Service Provider (ISP). In this example, the ADSL interface is encapsulated with PPPoA. PPPoA links typically include authentication, so a user ID and password are configured in this example.

Figure 5-1 Typical ADSL network configuration



With PPPoA encapsulation, the local and remote IP addresses can be automatically negotiated instead of explicitly specified. If addresses are not specified, the default behavior is to autonegotiate the addresses.

Example 5-1 sets up a PPPoA encapsulation on interface adsl0. In this example:

- A Sangoma S518 ADSL NIC is connected to the interface.
- The interface has one PVC. The PVC identifier is automatically detected.
- The PPPoA unit number is 0.
- 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 address resides on the same network as local interface.
- The user ID is set to “customerA”.
- The password is set to “Aremotsuc”.

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 5-1 Creating and configuring an ADSL interface for PPPoA 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 PPPoA using unit number 0.	vyatta@R1# **set interfaces adsl adsl0 pvc auto pppoa 0** [edit]
Assign the local IP address to the interface.	vyatta@R1# **set interfaces adsl adsl0 pvc auto pppoa 0 local-address 192.0.2.65** [edit]
Set the network mask (prefix length) for the interface.	vyatta@R1# **set interfaces adsl adsl0 pvc auto pppoa 0 prefix-length 24** [edit]
Set the IP address of the far end of the connection.	vyatta@R1# **set interfaces adsl adsl0 pvc auto pppoa 0 remote-address 192.0.2.70** [edit]
Set the user id for the link.	vyatta@R1# **set interfaces adsl adsl0 pvc auto pppoa 0 user-id customerA** [edit]
Set the password for the link.	vyatta@R1# **set interfaces adsl adsl0 pvc auto pppoa 0 password Aremotsuc** [edit]
Commit the configuration.	vyatta@R1# **commit** [edit]

View the configuration.	vyatta@R1# **show interfaces adsl adsl0**
	pvc auto {
	pppoa 0 {
	local-address 192.0.2.65
	prefix-length 24
	remote-address 192.0.2.70
	user-id customerA
	password Aremotsuc
	}
	}
	vyatta@R1#

PPPoA Commands

This chapter contains the following commands.

Configuration Commands	
interfaces adsl <adslx> pvc <pvc-id> pppoa <num>	Specifies PPPoA encapsulation for a PVC on an ADSL interface.
interfaces adsl <adslx> pvc <pvc-id> pppoa <num> connect-on-demand	Enables or disables on-demand PPPoA connection on an ADSL interface.
interfaces adsl <adslx> pvc <pvc-id> pppoa <num> default-route <param>	Enables or disables automatically adding a default route when a PPPoA link is brought up.
interfaces adsl <adslx> pvc <pvc-id> pppoa <num> idle-timeout <timeout>	Specifies the length of time in seconds to wait before disconnecting an idle on-demand ADSL PPPoA session.
interfaces adsl <adslx> pvc <pvc-id> pppoa <num> local-address <ipv4>	Assign an IP address to a PVC with PPPoA encapsulation on an ADSL interface.
interfaces adsl <adslx> pvc <pvc-id> pppoa <num> mtu <mtu>	Specify the Maximum Transmit Unit (MTU) size for a PVC with PPPoA encapsulation on an ADSL interface.
interfaces adsl <adslx> pvc <pvc-id> pppoa <num> name-server <param>	Specifies whether an ADSL PPPoA interface should obtain name server entries from the remote peer interface.
interfaces adsl <adslx> pvc <pvc-id> pppoa <num> password <password>	Specifies the password to use to authenticate with the remote PPPoA endpoint.
interfaces adsl <adslx> pvc <pvc-id> pppoa <num> remote-address <ipv4>	Sets the IP address of the remote end of a PPPoA-encapsulated link on an ADSL interface.
interfaces adsl <adslx> pvc <pvc-id> pppoa <num> user-id <user-id>	Specifies the user ID to use to authenticate with the remote PPPoA endpoint.
Operational Commands	
clear interfaces connection <pppoax>	Brings a PPPoA-encapsulated DSL interface down then up.
connect interface <pppoax>	Brings a PPPoA-encapsulated DSL interface up.
disconnect interface <pppoax>	Brings a PPPoA-encapsulated DSL interface down.
show interfaces pppoa	Displays IP layer information about PPPoA interfaces.

Commands for using other system features with PPPoA–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 *Vyatta WAN Interfaces Reference Guide*.
Firewall	Commands for configuring firewall on PPPoA encapsulated interfaces are described in the *Vyatta Security Reference Guide*.
OSPF	Commands for configuring the Open Shortest Path First routing protocol on PPPoA encapsulated interfaces are described in the *Vyatta OSPF Reference Guide*.
RIP	Commands for configuring the Routing Information Protocol on PPPoA encapsulated interfaces are described in the *Vyatta RIP Reference Guide*.
QoS	Commands for configuring quality of service on PPPoA encapsulated interfaces are described in the *Vyatta Policy and QoS Reference Guide*.

clear interfaces connection <pppoax>

Brings a PPPoA-encapsulated DSL interface down then up.

Syntax

clear interfaces connection *pppoax*

Command Mode

Operational mode.

Parameters

pppoax	Mandatory. The interface to be operationally brought down, then up. The interface is the name of a PPPoA-encapsulated DSL interface; that is the interface name is **pppoa***x*.

Default

None.

Usage Guidelines

Use this command to operationally bring a Point-to-Point Protocol over Asynchronous Transfer Mode (PPPoA) interface down and then up.

connect interface <pppoax>

Brings a PPPoA-encapsulated DSL interface up.

Syntax

connect interface *pppoax*

Command Mode

Operational mode.

Parameters

pppoax	Mandatory. The name of the interface. This is the name of a PPPoA-encapsulated DSL interface; that is the interface name is **pppoa***x*.

Default

None.

Usage Guidelines

Use this command to operationally bring a Point-to-Point Protocol over Asynchronous Transfer Mode (PPPoA) interface up.

disconnect interface <pppoax>

Brings a PPPoA-encapsulated DSL interface down.

Syntax

disconnect interface *pppoax*

Command Mode

Operational mode.

Parameters

pppoax	Mandatory. The name of the interface. This is the name of a PPPoA-encapsulated DSL interface; that is the interface name is **pppoa***x*.

Default

None.

Usage Guidelines

Use this command to operationally bring a Point-to-Point Protocol over Asynchronous Transfer Mode (PPPoA) DSL interface down.

interfaces adsl <adslx> pvc <pvc-id> pppoa <num>

Specifies PPPoA encapsulation for a PVC on an ADSL interface.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num*

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoa 0-15 {
         }
      }
   }
}
```

Parameters

adslx	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be **adsl0** to **adsl***x*, depending on what physical ADSL ports are actually available on the system.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named **pppoa0** to **pppoa15**.

Default

None.

Usage Guidelines

Use this command to specify PPPoA (Point-to-Point Protocol over Asynchronous Transfer Mode) encapsulation.

Use the **set** form of this command to apply PPPoA encapsulation.

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

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

interfaces adsl <adslx> pvc <pvc-id> pppoa <num> connect-on-demand

Enables or disables on-demand PPPoA connection on an ADSL interface.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **connect-on-demand**

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **connect-on-demand**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoa 0-15 {
            connect-on-demand
         }
      }
   }
}
```

Parameters

adslx	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be **adsl0** to **adsl***x*, depending on what physical ADSL ports are actually available on the system.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named **pppoa0** to **pppoa15**.

Default

On-demand PPPoA connection is disabled.

Usage Guidelines

Use this command to direct the system to establish ADSL Point-to-Point Protocol over Asynchronous Transfer Mode (PPPoA) connections automatically just when traffic is sent.

When on-demand PPPoA connection is disabled, PPPoA links are created at boot time and remain up. If the link fails for any reason, the system brings the link back up immediately.

When on-demand PPPoA connection is enabled, the PPPoA link is brought up only when IP traffic needs to be sent on the link. If the link fails for any reason, it is brought back up again the next time traffic needs to be sent.

If you configure an on-demand PPPoA connection, you must also configure the idle timeout period, after which an idle PPPoA link will be disconnected. If a non-zero idle timeout period is not configured, the on-demand link will never be disconnected after the first time it is brought up. To configure the idle timeout period, use the **interfaces adsl <adslx> pvc <pvc-id> pppoa <num> idle-timeout <timeout>** command (see page 158).

If you configure an on-demand PPPoA connection, you must also configure remote-address. To configure the remote address, use the **interfaces adsl <adslx> pvc <pvc-id> pppoa <num> remote-address <ipv4>** command (see page 168).

Use the **set** form of this command to enable on-demand PPPoA connections.

Use the **delete** form of this command to disable on-demand PPPoA connections.

Use the **show** form of this command to show PPPoA connection configuration.

interfaces adsl <adslx> pvc <pvc-id> pppoa <num> default-route <param>

Enables or disables automatically adding a default route when a PPPoA link is brought up.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **default-route** *param*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **default-route**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoa 0-15 {
            default-route [auto|none]
         }
      }
   }
}
```

Parameters

adslx	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be **adsl0** to **adsl***x*, depending on what physical ADSL ports are actually available on the system.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named **pppoa0** to **pppoa15**.

param	Mandatory. Specifies whether a default route is automatically added when the PPP link comes up. **auto**: The PPP process automatically adds a default route to the remote end of the link. **none**: No default route is added.

Default

A default route to the remote endpoint is automatically added when the link comes up.

Usage Guidelines

Use this command to specify whether a default route pointing to the remote endpoint of a Point-to-Point Protocol over Asynchronous Transfer Mode (PPPoA) ADSL link is automatically added when the link comes up. The default route is only added if no other default route already exists in the system.

Use the **set** form of this command to enable or disable adding the default route.

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

Use the **show** form of this command to show the default route configuration.

interfaces adsl <adslx> pvc <pvc-id> pppoa <num> idle-timeout <timeout>

Specifies the length of time in seconds to wait before disconnecting an idle on-demand ADSL PPPoA session.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **idle-timeout** *timeout*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **idle-timeout**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **idle-timeout**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoa 0-15 {
            idle-timeout u32
         }
      }
   }
}
```

Parameters

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

num	Mandatory. The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named **pppoa0** to **pppoa15**.
timeout	Mandatory. The amount of time, in seconds, after which an idle connection will be closed. The range is 0 to 4294967295, where 0 means the connection is never closed.

Default

Idle connections are never disconnected.

Usage Guidelines

Use this command to set the idle timeout interval to be used with on-demand ADSL Point-to-Point Protocol over Asynchronous Transfer Mode (PPPoA) connections.

When on-demand PPPoA link connection is enabled, the link is brought up only when traffic is to be sent and is disabled when the link is idle for the interval specified by this command. On-demand PPPoA connection is enabled using the **interfaces adsl <adslx> pvc <pvc-id> pppoa <num> connect-on-demand** command (see page 154).

If this parameter is not set or is set to 0, an on-demand link will not be taken down when it is idle and after the initial establishment of the connection will behave like an ordinary PPPoA link.

Use the **set** form of this command to specify the idle timeout value.

Use the **delete** form of this command to restore default behavior for idle timeout.

Use the **show** form of this command to display idle timeout configuration.

interfaces adsl <adslx> pvc <pvc-id> pppoa <num> local-address <ipv4>

Assign an IP address to a PVC with PPPoA encapsulation on an ADSL interface.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **local-address** *ipv4*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **local-address**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **local-address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoa 0-15 {
            local-address ipv4
         }
      }
   }
}
```

Parameters

adslx	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be **adsl0** to **adsl***x*, depending on what physical ADSL ports are actually available on the system.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named **pppoa0** to **pppoa15**.

ipv4	Optional. The IPv4 address for the link.

Default

If not set, the local address is negotiated.

Usage Guidelines

Use this command to specify an IP address for an ADSL PVC with Point-to-Point Protocol over Asynchronous Transfer Mode (PPPoA) encapsulation.

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 adsl <adslx> pvc <pvc-id> pppoa <num> mtu <mtu>

Specify the Maximum Transmit Unit (MTU) size for a PVC with PPPoA encapsulation on an ADSL interface.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **mtu** *mtu*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **mtu**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **mtu**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoa 0-15 {
            mtu 8-8188
         }
      }
   }
}
```

Parameters

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

num	Mandatory. The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named **pppoa0** to **pppoa15**.
mtu	Optional. 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 a Point-to-Point Protocol over Asynchronous Transfer Mode (PPPoA) ADSL interface. 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> pppoa <num> name-server <param>

Specifies whether an ADSL PPPoA interface should obtain name server entries from the remote peer interface.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **name-server** *param*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **name-server**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoa 0-15 {
            name-server [auto|none]
         }
      }
   }
}
```

Parameters

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

num	Mandatory. The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named **pppoa0** to **pppoa15**.
param	Mandatory. Specifies whether the local PPPoA endpoint should obtain name server entries from the remote endpoint. Supported values are as follows: **auto**: The endpoint obtains name server entries from its peer. **none**: The endpoint uses the name server(s) configured for the local system.

Default

The interface obtains name server entries from its peer (i.e. **auto**).

Usage Guidelines

Use this command to define how a name server is defined when an ADSL Point-to-Point Protocol over Asynchronous Transfer Mode (PPPoA) link is brought up.

Use the **set** form of this command to set the way that name server entries are obtained by the PPPoA endpoint.

Use the **delete** form of this command to restore the default behavior for obtaining name server entries.

Use the **show** form of this command to show the PPPoA name server configuration.

interfaces adsl <adslx> pvc <pvc-id> pppoa <num> password <password>

Specifies the password to use to authenticate with the remote PPPoA endpoint.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **password** *password*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **password**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **password**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoa 0-15 {
            password text
         }
      }
   }
}
```

Parameters

adslx	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be **adsl0** to **adsl***x*, depending on what physical ADSL ports are actually available on the system.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named **pppoa0** to **pppoa15**.

password	Mandatory. The password used to authenticate the local endpoint with the remote PPPoA server.

Default

None.

Usage Guidelines

Use this command to set the authentication password for a Point-to-Point Protocol over Asynchronous Transfer Mode (PPPoA) ADSL endpoint.

Authentication is optional from the system's point of view; however, most service providers require it.

The password is used in conjunction with the user ID to authenticate the local system to the remote endpoint. The user ID is set by using the **interfaces adsl <adslx> pvc <pvc-id> pppoa <num> remote-address <ipv4>** command (see page 168). The authentication protocol is determined by the remote endpoint. Use the **set** form of this command to set the password.

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

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

interfaces adsl <adslx> pvc <pvc-id> pppoa <num> remote-address <ipv4>

Sets the IP address of the remote end of a PPPoA-encapsulated link on an ADSL interface.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **remote-address** *ipv4*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **remote-address**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **remote-address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoa 0-15 {
            remote-address ipv4
         }
      }
   }
}
```

Parameters

adslx	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be **adsl0** to **adsl***x*, depending on what physical ADSL ports are actually available on the system.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named **pppoa0** to **pppoa15**.

ipv4	Mandatory. The IP address of the remote end of the PPPoA 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 a Point-to-Point Protocol over Asynchronous Transfer Mode (PPPoA) connection on an ADSL interface.

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.

interfaces adsl <adslx> pvc <pvc-id> pppoa <num> user-id <user-id>

Specifies the user ID to use to authenticate with the remote PPPoA endpoint.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **user-id** *user-id*

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **user-id**

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **user-id**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535|auto] {
         pppoa 0-15 {
            user-id text
         }
      }
   }
}
```

Parameters

adslx	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be **adsl0** to **adsl***x*, depending on what physical ADSL ports are actually available on the system.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
num	Mandatory. The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named **pppoa0** to **pppoa15**.

user-id	Optional. The user ID to be used by the local endpoint to authenticate itself to the remote endpoint.

Default

None.

Usage Guidelines

Use this command to set the user ID for authenticating with the remote PPPoA endpoint.

Authentication is optional from the system’s point of view; however, most service providers require it.

The user ID is used in conjunction with the password to authenticate the local system to the remote endpoint. The password is set by using the **interfaces adsl <adslx> pvc <pvc-id> pppoa <num> password <password>** command (see page 166). The authentication protocol is determined by the remote endpoint. Use the **set** form of this command to set the user ID.

Use the **delete** form of this command to remove the user ID.

Use the **show** form of this command to display user ID configuration.

show interfaces pppoa

Displays IP layer information about PPPoA interfaces.

Syntax

show interfaces pppoa

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display IP-layer information about PPPoA interfaces.

Chapter 6: Classical IPoA

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

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

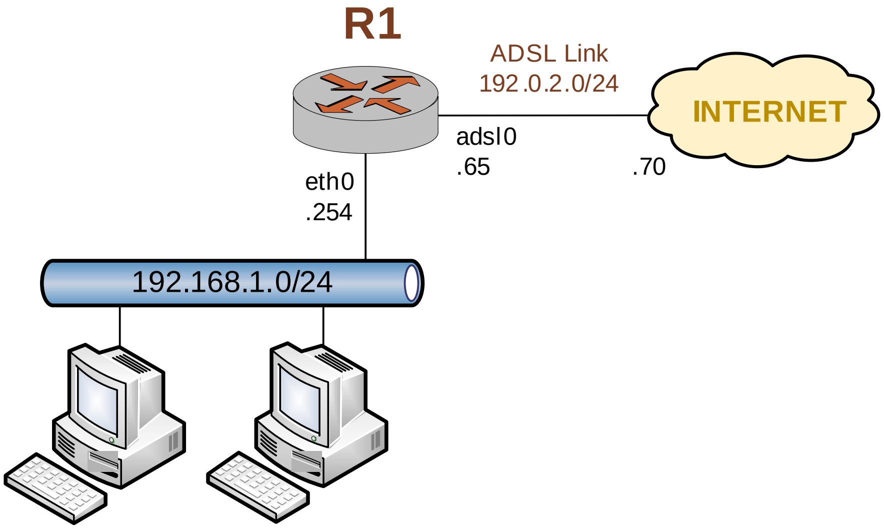
Classical IPoA Overview

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 6-1 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 6-1 Typical ADSL network configuration



Example 6-1 sets up a Classical IPoA encapsulation on interface adsl0. In this example:

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

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 6-1 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]`

Example 6-1 Creating and configuring an ADSL interface for Classical IPoA encapsulation

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
	      }
	   }
	vyatta@R1#

Classical IPoA Commands

This chapter contains the following commands.

Configuration Commands	
interfaces adsl <adslx> pvc <pvc-id> classical-ipoa	Specifies RFC 1577 Classical IPoA encapsulation for a PVC on an ADSL interface.
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.
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.
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.
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.
Operational Commands	
clear interfaces connection <adslx>	Brings a Classical IPoA–encapsulated DSL interface down then up.
connect interface <adslx>	Brings a Classical IPoA–encapsulated DSL interface up.
disconnect interface <adslx>	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 *Vyatta WAN Interfaces Reference Guide*.
Firewall	Commands for configuring firewall on Classical-IPoA encapsulated interfaces are described in the *Vyatta Security Reference Guide*.
OSPF	Commands for configuring the Open Shortest Path First routing protocol on Classical-IPoA encapsulated interfaces are described in the *Vyatta OSPF Reference Guide*.

RIP	Commands for configuring the Routing Information Protocol on Classical-IPoAencapsulated interfaces are described in the *Vyatta RIP Reference Guide*.
QoS	Commands for configuring quality of service on Classical-IPoA encapsulated interfaces are described in the *Vyatta Policy and 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

adslx	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 **adsl***x*.

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

adslx	Mandatory. The name of the interface. This is the name of a Classical IPoA-encapsulated DSL interface; that is the interface name can be **adsl***x*.

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

adslx	Mandatory. The name of the interface. This is the name of a Classical IPoA-encapsulated DSL interface; that is the interface name can be **adsl***x*.

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

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

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

adslx	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be **adsl0** to **adsl***x*, depending on what physical ADSL ports are actually available on the system.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
ipv4	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

adslx	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be **adsl0** to **adsl***x*, depending on what physical ADSL ports are actually available on the system.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
mtu	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

adslx	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be **adsl0** to **adsl***x*, depending on what physical ADSL ports are actually available on the system.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
prefix	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

adslx	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be **adsl0** to **adsl***x*, depending on what physical ADSL ports are actually available on the system.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
ipv4	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 7: Bridged Ethernet

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

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

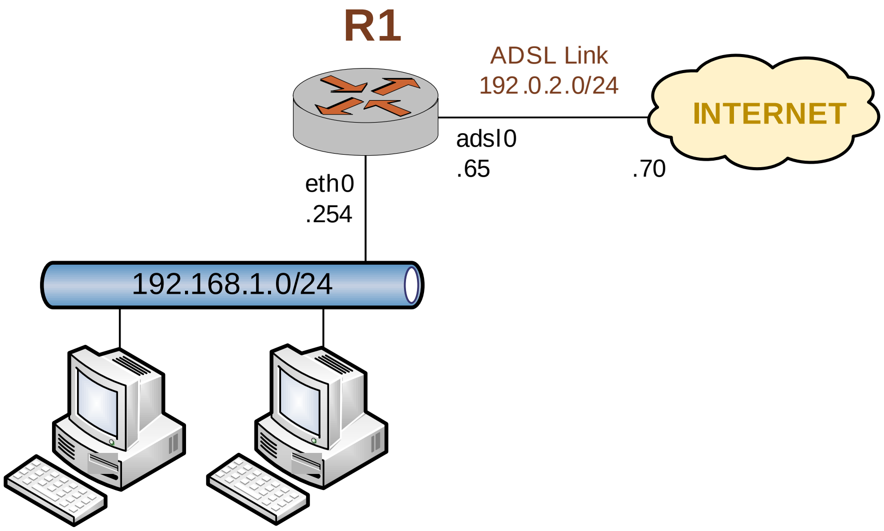
Bridged Ethernet Overview

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 7-1 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 7-1 Typical ADSL network configuration



Example 7-1 sets up a Bridged Ethernet encapsulation on interface adsl0. 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 7-1 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 adsl adsl0 pvc 0/35`**
	`[edit]`

Set the line encapsulation to Bridged Ethernet.	`vyatta@R1#` **`set interfaces adsl adsl0 pvc 0/35 bridged-ethernet`**
	`[edit]`

Example 7-1 Creating and configuring an ADSL interface for Bridged Ethernet encapsulation

Assign the local IP address to the interface.	vyatta@R1# **set interfaces adsl adsl0 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 adsl0 pvc 0/35 bridged-ethernet prefix-length 24** [edit]
Commit the configuration.	vyatta@R1# **commit** [edit]

View the configuration.	vyatta@R1# **show interfaces adsl adsl0**
	 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 *Vyatta Security Reference Guide*.
OSPF	Commands for configuring the Open Shortest Path First routing protocol on Bridged Ethernet encapsulated interfaces are described in the *Vyatta OSPF Reference Guide*.
RIP	Commands for configuring the Routing Information Protocol on Bridged Ethernet encapsulated interfaces are described in the *Vyatta RIP Reference Guide*.
QoS	Commands for configuring quality of service on Bridged Ethernet encapsulated interfaces are described in the *Vyatta Policy and QoS Reference Guide*.

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

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

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

adslx	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be **adsl0** to **adsl***x*, depending on what physical ADSL ports are actually available on the system.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair (e.g., 0/35) or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
ipv4	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

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

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

adslx	Mandatory. Multi-node. The identifier for the ADSL interface you are defining. This may be **adsl0** to **adsl***x*, depending on what physical ADSL ports are actually available on the system.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair (e.g., 0/35) or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* 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.
ipv4	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.

Chapter 8: Multilink Interfaces

This chapter describes commands for working with multilink interfaces.

This chapter presents the following topics:

- Multilink Interface Commands

Multilink Interface Commands

This chapter contains the following commands.

Configuration Commands	
interfaces multilink <mlx>	Defines the characteristics of Point-to-Point Protocol encapsulation on a serial interface.
interfaces multilink <mlx> authentication	Specifies the authentication parameters for a multilink interface.
interfaces multilink <mlx> description <desc>	Specifies a description for a virtual interface on a multilink interface.
interfaces multilink <mlx> lcp-echo-failure <value>	Specifies the LCP echo failure threshold for a multilink interface.
interfaces multilink <mlx> lcp-echo-interval <interval>	Specifies the LCP echo interval for a multilink interface.
interfaces multilink <mlx> logging <state>	Specifies whether to enable or disable logging of debugging messages for the multilink process.
interfaces multilink <mlx> mrru <mrru>	Specify the MRRU size for a multilink interface.
interfaces multilink <mlx> mtu <mtu>	Specify the MTU size for a multilink interface.
interfaces multilink <mlx> vif 1 address local-address <ipv4>	Sets the IP address for a virtual interface on a multilink interface.
interfaces multilink <mlx> vif 1 address prefix-length <prefix>	Specifies the prefix defining the network served by a virtual interface on a multilink interface.
interfaces multilink <mlx> vif 1 address remote-address <ipv4>	Specifies the IP address of the remote endpoint on a multilink connection.
interfaces multilink <mlx> vif 1 description <desc>	Sets the description for a virtual interface on a multilink interface.
Operational Commands	
clear interfaces multilink	Clears counters for multilink interfaces
show interfaces multilink	Displays information about multilink interfaces.

Commands for using other system features with multilink interfaces can be found in the following locations.

Related Commands Documented Elsewhere	
OSPF	Commands for configuring the Open Shortest Path First routing protocol on multilink interfaces are described in the *Vyatta OSPF Reference Guide*.
RIP	Commands for configuring the Routing Information Protocol on multilink interfaces are described in the *Vyatta RIP Reference Guide*.

clear interfaces multilink

Clears counters for multilink interfaces

Syntax

clear interfaces multilink [*ml0..ml23*]

Command Mode

Operational mode.

Parameters

ml0..ml23	Clears the statistics on the specified multilink interface. *Multilink interfaces are numbered ml0 (“em ell zero”) through ml23 (“em ell twenty-three”)*

Usage Guidelines

Use this command to clear statistics for a specified multilink interface.

If no multilink interface is specified then statistics are cleared on all multilink interfaces.

Examples

Example 8-1 clears statistics on all multilink interfaces.

Example 8-1 “clear interfaces multilink”: Clearing multilink statistics

```
vyatta@R1> clear interfaces multilink
PPP statistics flushed
PPP statistics flushed
vyatta@R1>
```

Example 8-2 clears statistics on a specific multilink interface.

Example 8-2 “clear interfaces multilink”: Clearing multilink statistics on one interface

```
vyatta@R1> clear interfaces multilink ml0
PPP statistics flushed
vyatta@R1>
```

interfaces multilink <mlx>

Defines the characteristics of Point-to-Point Protocol encapsulation on a serial interface.

Syntax

set interfaces multilink *mlx*

delete interfaces multilink *mlx*

show interfaces multilink *mlx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   multilink ml0..ml23 {
   }
}
```

Parameters

mlx	Mandatory. The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”).

Default

None.

Usage Guidelines

Use this command to define a multilink bundle. A multilink bundle allows the bandwidth of individual links to be combined into a single virtual link.

Multilink bundling is currently supported only for point-to-point protocol (PPP) links. You can create up to two multilink bundles and each bundle can include eight individual PPP links.

To combine multiple serial interfaces into a single multilink bundle you create both the multilink interface and the individual serial interfaces. Individual links are assigned to the bundle using the **multilink** parameter of the **interfaces serial <wanx> ppp authentication** command (see page 60).

When PPP connections are bundled into a multilink, the settings on the multilink override the settings on the individual PPP link. The exceptions is authentication (authentication settings specified for individual PPP links override authentication settings for the multilink) and MTU/MRU/MRRU.

A transmitted packet may not be larger than the remote device is willing to receive. The actual MTU is the smaller of the configured MTU of the local device and the configured MRU of the remote device; this value is determined by MRU negotiation when the link is established.

The interaction between MTU/MRU in PPP links and MTU/MRRU in a multilink bundle is as follows:

- If MTU is unconfigured in both the member PPP link and the multilink bundle, the default for member links is used.
- If MTU is set in member links but not in the multilink bundle, the configured value for member links is used. These must match for every PPP link in the bundle.
- If MTU is set in the multilink bundle, it overrides any value (default or configured) for member links.
- MRRU (for the multilink bundle) and MRU (for member links) are configured independently and used separately during MRU negotiation. If neither is set, the MRU default value is used for MRU and the MRRU default value is used for MRRU.

In multilink bundles, if an individual member link goes down, the multilink bundle remains up, and if the member link becomes operational again it will become a member of the same bundle. If all member links fail, the multilink bundle will also fail, but will become operational again if any of the member links comes back up.

Use the **set** form of this command to define multilink settings on an interface.

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

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

interfaces multilink <mlx> authentication

Specifies the authentication parameters for a multilink interface.

Syntax

set interfaces multilink *mlx* **authentication** [**password** *password* | **peer-password** *password* | **peer-system-name** *name* | **peer-user-id** *user-id* | **refuse-type** *type* | **system-name** *name* | **type** *type* | **user-id** *user-id*]

delete interfaces multilink *mlx* **authentication**

show interfaces multilink *mlx* **authentication**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   multilink ml0..ml23 {
      authentication {
         password: text
         peer-password: text
         peer-system-name: text
         peer-user-id: text
         refuse-type: [none|chap|pap|papchap|mschap|mschap-v2|eap]
         system-name: text
         type: [none|chap|pap|papchap|mschap|mschap-v2|eap|any]
         user-id: text
      }
   }
}
```

Parameters

mlx	Mandatory. The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”).
password *password*	Optional. Sets the password this system will use when authenticating itself to a peer.
peer-password *password*	Optional. Sets the password this system will accept from a peer.

peer-system-name *name*	Optional. The system name this system will accept from a peer.
peer-user-id *user-id*	Optional. The user ID this system will accept from a peer.
refuse-type *type*	Defines authentication types that will be refused during authentication negotiations. Used when the Vyatta system is acting as the client side of the communication. **none**: Does not refuse any type of authentication; that is, the system will authenticate to the peer any type of authentication requested, including not using authentication. **chap**: Refuses CHAP authentication if offered by the remote peer. **pap**: Refuses PAP authentication if offered by the remote peer. **papchap**: Refuses PAP or CHAP authentication if offered by the remote peer. **mschap**: Refuses MS-CHAP authentication if offered by the remote peer. **mschap-v2**: Refuses MS-CHAP v2 authentication if offered by the remote peer. **eap**: Refuses EAP authentication if offered by the remote peer. The default is none.
system-name *name*	Optional. The system name this system will use when authenticating itself to a peer.

type *type*	Optional. Sets the authentication required from the remote peer. Used when the Vyatta system is acting as the server side of the communication. Supported values are as follows: **none**: The remote peer is not required to authenticate itself. **chap**: The remote peer must authenticate using the Challenge Handshake Authentication Protocol (CHAP), as defined in RFC 1994. **pap**: The remote peer must authenticate using the Password Authentication Protocol (PAP). The client authenticates itself by sending a user ID and a password to the server, which the server compares to the password in its internal database. **papchap**: The remote peer must authenticate using either PAP or CHAP as the authentication method. **mschap**: The remote peer must authenticate using the Microsoft Challenge Handshake Authentication Protocol (MS-CHAP), which is the Microsoft version of CHAP and is an extension to RFC 1994. **mschap-v2**: The remote peer must authenticate using version 2 of MS-CHAP. **eap**: The remote peer must authenticate using Extensible Authentication Protocol (EAP), which is an authentication framework frequently used in mobile networks and point-to-point connections. **any**: The peer is required to authenticate itself (that is, **none** is refused), but any supported method of authentication offered by the remote peer is accepted. The default is none.
user-id *user-id*	Optional. The user ID this system will use when authenticating itself to a peer.

Default

None.

Usage Guidelines

Use this command to set the authentication parameters for a multilink interface. These authentication requirements must be satisfied before network packets are sent or received.

Use the **set** form of this command to set the authentication parameters.

Use the **delete** form of this command to remove authentication configuration or restore default information.

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

interfaces multilink <mlx> description <desc>

Specifies a description for a virtual interface on a multilink interface.

Syntax

set interfaces multilink *mlx* **description** *desc*

delete interfaces multilink *mlx* **description**

show interfaces multilink *mlx* **description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   multilink ml0..ml23 {
      description: text
   }
}
```

Parameters

mlx	Mandatory. The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”).
desc	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 virtual interface on a multilink interface.

Use the **set** form of this command to set 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 multilink <mlx> lcp-echo-failure <value>

Specifies the LCP echo failure threshold for a multilink interface.

Syntax

set interfaces multilink *mlx* **lcp-echo-failure** *value*

delete interfaces multilink *mlx* **lcp-echo-failure**

show interfaces multilink *mlx* **lcp-echo-failure**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   multilink ml0..ml23 {
      lcp-echo-failure: u32
   }
}
```

Parameters

mlx	Mandatory. The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”).
value	Optional. Sets the LCP echo failure threshold. The failure threshold is the maximum number of LCP echo-requests that can be sent without receiving a valid LCP echo-reply. If this threshold is exceeded, the peer is considered to be dead and the connection is terminated. The value specified must be a non-zero number. The default is 3. Deleting this value does not disable LCP echoes, but instead restores the default value. If this parameter is set, the **lcp-echo-interval** parameter must also be set.

Default

A maximum of 3 LCP echo-requests can be sent without receiving a valid LCP echo-reply.

Usage Guidelines

Use this command to specify the LCP echo failure threshold for a multilink interface.

Use the **set** form of this command to set the LCP echo failure threshold.

Use the **delete** form of this command to restore the default LCP echo failure threshold configuration.

Use the **show** form of this command to view LCP echo failure threshold configuration.

interfaces multilink <mlx> lcp-echo-interval <interval>

Specifies the LCP echo interval for a multilink interface.

Syntax

set interfaces multilink *mlx* **lcp-echo-interval** *value*

delete interfaces multilink *mlx* **lcp-echo-interval**

show interfaces multilink *mlx* **lcp-echo-interval**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   multilink ml0..ml23 {
      lcp-echo-interval: u32
   }
}
```

Parameters

mlx	Mandatory. The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”).
interval	Optional. Sets the LCP echo interval, which is the number of seconds between LCP echoes. LCP echoes are used to determine whether the connection is still operational. The value specified must be a non-zero number. The default is 3. Deleting this value does not disable LCP echoes, but instead restores the default value. Specifying a low value for this parameter allows fast detection of failed links. The value set for this parameter must match the value set on the peer.

Default

LCP echo-requests are sent at 3-second intervals.

Usage Guidelines

Use this command to specify the LCP echo interval for a multilink interface.

Use the **set** form of this command to set the LCP echo interval.

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

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

interfaces multilink <mlx> logging <state>

Specifies whether to enable or disable logging of debugging messages for the multilink process.

Syntax

set interfaces multilink *mlx* **logging** *state*

delete interfaces multilink *mlx* **logging**

show interfaces multilink *mlx* **logging**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   multilink ml0..ml23 {
      authentication {
         logging: [on | off]
      }
   }
}
```

Parameters

mlx	Mandatory. The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”).
state	Enables logging of debugging messages for the PPP process. Supported values are as follows: **on**: Enables debugging for PPP connections. Trace-level messages are sent from the PPP process to the system log. **off**: Disables debugging for PPP connections. Note that logging creates additional system load and may degrade performance.

Default

Logging of debugging messages is disabled.

Usage Guidelines

Use this command to enable or disable logging of debugging messages for the multilink process.

Use the **set** form of this command to specify whether to enable or disable debugging on a multilink interface.

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

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

interfaces multilink <mlx> mrru <mrru>

Specify the MRRU size for a multilink interface.

Syntax

set interfaces multilink *mlx* **mru** *mru*

delete interfaces multilink *mlx* **mru**

show interfaces multilink *mlx* **mru**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   multilink ml0..ml23 {
      mrru: 8-8188
   }
}
```

Parameters

mlx	Mandatory. The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”).
mrru	Optional. Sets the Maximum Reconstructed Receive Unit (MRRU). This is the maximum size for a received packet on a multilink bundle, analogous to the MRU for individual links. The range is 8 to 8188. The default is 1600. A value of 296 (40 bytes for the TCP/IP header + 256 bytes of data) works well on very slow links. Note that for IPv6 connections, the MRRU must be at least 1280.

Default

The default is 1600.

Usage Guidelines

Use this command to specify the Maximum Reconstructed Receive Unit (MRRU) for a multilink interface. This is the maximum packet size the interface is willing to receive.

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

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

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

interfaces multilink <mlx> mtu <mtu>

Specify the MTU size for a multilink interface.

Syntax

set interfaces multilink *mlx* **mtu** *mtu*

delete interfaces multilink *mlx* **mtu**

show interfaces multilink *mlx* **mtu**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   multilink ml0..ml23 {
      mtu: 8-8188
   }
}
```

Parameters

mlx	Mandatory. The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”).
mtu	Optional. Sets the Maximum Transmit Unit (MTU). Unless the peer requests a smaller value (by means of MRU negotiation), packets larger than this number are fragmented. The range is 8 to 8188. The default is 1500. Note that for IPv6 connections, the MTU must be at least 1280.

Default

The default is 1500.

Usage Guidelines

Use this command to specify the Maximum Transmit Unit (MTU) for a Point-to-Point Protocol (PPP) 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 value.

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

interfaces multilink <mlx> vif 1 address local-address <ipv4>

Sets the IP address for a virtual interface on a multilink interface.

Syntax

set interfaces multilink *mlx* **vif 1 address local-address** *ipv4*

delete interfaces multilink *mlx* **vif 1 address local-address**

show interfaces multilink *mlx* **vif 1 address local-address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   multilink ml0..ml23 {
      vif 1 {
         address {
            local-address: ipv4
         }
      }
   }
}
```

Parameters

mlx	Mandatory. The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”).
1	The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be **1**.

ipv4	Optional if specified on the peer; mandatory otherwise. The IPv4 address for this vif. If multiple PPP interfaces are all endpoints for a multi-link PPP bundle, all links in the bundle must share the same IP addresses. If an interface is to form part of a multi-link bundle, the IP address need not be explicitly assigned. In this case, the IP address must be received from the remote PPP peer.

Default

None.

Usage Guidelines

Use this command to specify an IP address for a virtual interface on a multilink 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 multilink <mlx> vif 1 address prefix-length <prefix>

Specifies the prefix defining the network served by a virtual interface on a multilink interface.

Syntax

set interfaces multilink *mlx* **vif 1 address prefix-length** *prefix*

delete interfaces multilink *mlx* **vif 1 address prefix-length**

show interfaces multilink *mlx* **vif 1 address prefix-length**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   multilink ml0..ml23 {
      vif 1 {
         address {
            prefix-length: u32
         }
      }
   }
}
```

Parameters

mlx	Mandatory. The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”).
1	The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be **1**.
prefix	Optional if specified on the peer; mandatory otherwise. 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 a virtual interface on a multilink 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 multilink <mlx> vif 1 address remote-address <ipv4>

Specifies the IP address of the remote endpoint on a multilink connection.

Syntax

set interfaces multilink *mlx* **vif 1 address remote-address** *ipv4*

delete interfaces multilink *mlx* **vif 1 address remote-address**

show interfaces multilink *mlx* **vif 1 address remote-address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   multilink ml0..ml23 {
      vif 1 {
         address {
            remote-address: ipv4
         }
      }
   }
}
```

Parameters

mlx	Mandatory. The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”).
1	The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be **1**.
ipv4	Optional if specified on the peer; mandatory otherwise. An IPv4 address representing the network address of the remote peer.

Default

None.

Usage Guidelines

Use this command to specify the IP address of the remote endpoint in a multilink link.

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 multilink <mlx> vif 1 description <desc>

Sets the description for a virtual interface on a multilink interface.

Syntax

set interfaces multilink *mlx* **vif 1 description** *desc*

delete interfaces multilink *mlx* **vif 1 description**

show interfaces multilink *mlx* **vif 1 description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   multilink ml0..ml23 {
      vif 1 {
         description: text
      }
   }
}
```

Parameters

mlx	Mandatory. The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”).
1	The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be **1**.
desc	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 virtual interface on a multilink interface.

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

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

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

show interfaces multilink

Displays information about multilink interfaces.

Syntax

show interfaces multilink [*ml0..ml23*]

Command Mode

Operational mode.

Parameters

ml0..ml23	Shows detailed information for the specified multilink interface. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”).

Usage Guidelines

Use this command to view the operational status of a multilink interface.

When used with no option, this command displays summary information for all available multilink interfaces.

Note that an MLPPP link that is negotiating is considered to be in an “active” state.

Examples

Example 8-3 shows summary information for all configured multilink bundles.

Example 8-3 “show interfaces multilink”: Displaying summary multilink information

```
vyatta@R1> show interfaces multilink
ml0: <POINTOPOINT,MULTICAST,NOARP,UP,10000> mtu 1540 qdisc
pfifo_fast qlen 3
    link/ppp
    inet 3.3.3.1 peer 3.3.3.2/32 scope global ml0
    mrru 1560
RX:  bytes    packets    errors    dropped    overrun      mcast
        78          5         1          1         0          0
TX:  bytes    packets    errors    dropped    carrier collisions
        72          5         0          0         0          0
    Multilink members:
    wan0  : active

ml1: <POINTOPOINT,MULTICAST,NOARP,UP,10000> mtu 1540 qdisc
pfifo_fast qlen 3
    link/ppp
    inet 3.3.3.2 peer 3.3.3.1/32 scope global ml1
    mrru 1560
RX:  bytes    packets    errors    dropped    overrun     mcast
        72          5         0          0          0         0
TX:  bytes    packets    errors    dropped    carrier collisions
       109          7         0          0          0         0

    Multilink members:
    wan1  : active
```

Example 8-4 shows information for a single multilink bundle.

Example 8-4 “show interfaces multilink”: Displaying detailed information for a multilink bundle

```
vyatta@R1> show interfaces multilink ml0
ml0: <POINTOPOINT,MULTICAST,NOARP,UP,10000> mtu 1540 qdisc
pfifo_fast qlen 3
    link/ppp
    inet 3.3.3.1 peer 3.3.3.2/32 scope global ml0
    mrru 1560
RX:  bytes    packets    errors    dropped    overrun     mcast
        78          5         1          1          0         0
TX:  bytes    packets    errors    dropped    carrier collisions
        72          5         0          0          0         0

    Multilink members:
    wan0  : active

wan0: <POINTOPOINT,NOARP,UP,10000> mtu 1450 qdisc pfifo_fast
qlen 100
    link/ppp
    multilink ml0
RX:  bytes    packets    errors    dropped    overrun     mcast
       367         16         0          0          0         0
TX:  bytes    packets    errors    dropped    carrier collisions
       343         15         0          0          0         0


PPP data:
IN.BYTES  :         78
IN.PACK   :          5
IN.VJCOMP :          0
IN.VJUNC  :          0
IN.VJERR  :          0
OUT.BYTES :         72
OUT.PACK  :          5
OUT.VJCOMP:          0
OUT.VJUNC :          0
OUT.NON-VJ:          5
```

Chapter 9: Tunnel Interfaces

This chapter lists the commands for configuring GRE and IP-in-IP routable tunnel interfaces.

This chapter presents the following topics:

- Tunnel Commands

Tunnel Commands

This chapter contains the following commands.

Configuration Commands	
interfaces tunnel <tunx>	Defines a tunnel interface.
interfaces tunnel <tunx> address <ipv4net>	Sets a primary or secondary IP address for a tunnel interface.
interfaces tunnel <tunx> description <descr>	Specifies a description for a tunnel interface.
interfaces tunnel <tunx> disable	Disables a tunnel interface without discarding configuration.
interfaces tunnel <tunx> encapsulation	Sets the encapsulation for a tunnel interface.
interfaces tunnel <tunx> key <key>	Defines an authentication key for a tunnel interface.
interfaces tunnel <tunx> local-ip <ipv4>	Sets the IP address for the local endpoint of a tunnel.
interfaces tunnel <tunx> mtu <mtu>	Sets the MTU size for a tunnel interface.
interfaces tunnel <tunx> remote-ip <ipv4>	Sets the IP address for the remote endpoint of a tunnel.
interfaces tunnel <tunx> tos <tos>	Specifies the value to be written into the ToS byte of the tunnel packet's IP header.
interfaces tunnel <tunx> ttl <ttl>	Defines the time-to-live (TTL) value to be written into the tunnel packet's IP header.
Operational Commands	
clear interfaces tunnel counters	Clears tunnel interface statistics.
show interfaces tunnel	Displays information about tunnel interfaces.

Commands for using other system features with tunnel interfaces can be found in the following locations.

Related Commands Documented Elsewhere	
Firewall	Commands for configuring firewall on tunnel interfaces are described in the *Vyatta Security Reference Guide*.
OSPF	Commands for configuring the Open Shortest Path First routing protocol on tunnel interfaces are described in the *Vyatta OSPF Reference Guide*.
RIP	Commands for configuring the Routing Information Protocol on tunnel interfaces are described in the *Vyatta RIP Reference Guide*.

clear interfaces tunnel counters

Clears tunnel interface statistics.

Syntax

clear interfaces tunnel [*tunx*] **counters**

Command Mode

Operational mode.

Parameters

tunx	Optional. Clears information for the specified tunnel interface. The range is **tun0** to **tun23**.

Default

None.

Usage Guidelines

Use this command to clear statistics for tunnel interfaces.

interfaces tunnel <tunx>

Defines a tunnel interface.

Syntax

set interfaces tunnel *tunx*

delete interfaces tunnel [*tunx*]

show interfaces tunnel [*tunx*]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   tunnel tun0..tun23 {
   }
}
```

Parameters

tunx	Mandatory. Multi-node. An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**. You can define multiple tunnel interfaces by creating multiple **tunnel** configuration nodes.

Default

None.

Usage Guidelines

Use this command to create a tunnel interface for encapsulating traffic.

Use the **set** form of this command to create a tunnel interface.

Use the **delete** form of this command to remove a tunnel interface and all its configuration.

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

interfaces tunnel <tunx> address <ipv4net>

Sets a primary or secondary IP address for a tunnel interface.

Syntax

set interfaces tunnel *tunx* **address** *ipv4net*

delete interfaces tunnel *tunx* **address** [*ipv4net*]

show interfaces tunnel *tunx* **address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   tunnel tun0..tun23 {
      address: ipv4net
   }
}
```

Parameters

tunx	Mandatory. The name of the tunnel interface you are configuring. The range is **tun0** to **tun23**.
ipv4net	Multi-node. An IPv4 network address in the format *ip-address*/*prefix*. You can define more than one IP address for a tunnel interface by creating multiple **address** configuration nodes.

Default

None.

Usage Guidelines

Use this command to assign a primary or secondary IP address to a tunnel interface. At least one address must be configured for the tunnel interface to function.

Use the **set** form of this command to create an IP address for a tunnel interface. Note that you cannot use set to change an existing address; you must delete the address to be changed and create a new one.

Use the **delete** form of this command to remove an IP network address for a tunnel interface. At least one address must remain for the tunnel to function.

Use the **show** form of this command to view address configuration for a tunnel interface.

interfaces tunnel <tunx> description <descr>

Specifies a description for a tunnel interface.

Syntax

set interfaces tunnel *tunx* **description** *descr*

delete interfaces tunnel *tunx* **description**

show interfaces tunnel *tunx* **description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   tunnel tun0..tun23 {
      description: text
   }
}
```

Parameters

tunx	Mandatory. The name of the tunnel interface you are configuring. The range is **tun0** to **tun23**.
descr	A mnemonic name or description for the interface. The default is an empty string.

Default

None.

Usage Guidelines

Use this command to record a brief description for a tunnel interface. If the description contains spaces, it must be enclosed in double quotes.

Use the **set** form of this command to record a brief description description for the tunnel interface.

Use the **delete** form of this command to remove a description for the tunnel interface.

Use the **show** form of this command to view a description for the tunnel interface.

interfaces tunnel <tunx> disable

Disables a tunnel interface without discarding configuration.

Syntax

set interfaces tunnel *tunx* **disable**

delete interfaces tunnel *tunx* **disable**

show interfaces tunnel *tunx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   tunnel tun0..tun23 {
      disable
   }
}
```

Parameters

tunx	Mandatory. The name of the tunnel interface you are configuring. The range is **tun0** to **tun23**.

Default

The tunnel interface is enabled.

Usage Guidelines

Use this command to disable a tunnel interface without discarding configuration

Use the **set** form of this command to disable the tunnel interface.

Use the **delete** form of this command to enable the tunnel interface.

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

interfaces tunnel <tunx> encapsulation

Sets the encapsulation for a tunnel interface.

Syntax

set interfaces tunnel *tunx* **encapsulation** {**gre** | **ipip** | **sit**}

delete interfaces tunnel *tunx* **encapsulation**

show interfaces tunnel *tunx* **encapsulation**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   tunnel tun0..tun23 {
      encapsulation [gre|ipip|sit]
      }
   }
}
```

Parameters

tunx	Mandatory. The name of the tunnel interface you are configuring. The range is **tun0** to **tun9**.
gre	Uses Generic Routing Encapsulation (GRE) to encapsulate transported packets.
ipip	Uses IP-in-IP to encapsulate transported packets.
sit	Uses Simple Internet Transition (SIT) encapsulation.

Default

GRE is the encapsulation type.

Usage Guidelines

Use this command to set the encapsulation type for a tunnel.

The Generic Routing Encapsulation (GRE) protocol provides a simple-general purpose mechanism for encapsulating packets from a wide variety of network protocols to be forwarded over another protocol. The original packet (the “passenger” packet) can be one of many arbitrary network protocols—for example a multicast packet, an IPv6 packet, or a non-IP LAN protocol such as AppleTalk, Banyen VINES, or Novell IPX. The delivery protocol can be one of a number of routable IP protocols.

The IP-in-IP encapsulation protocol is used to tunnel between networks that have different capabilities or policies. For example, an IP-in-IP tunnel can be used to forward multicast packets across a section of a network (such as an IPsec tunnel) that does not support multicast routing. An IP-in-IP tunnel can also be used to influence the routing of the packet, or to deliver a packet to a mobile device using Mobile IP.

The SIT encapsulation is typically used to tunnel IPv6 across an IPv4 network.

Use the **set** form of this command to set the encapsulation type for a tunnel interface.

Use the **delete** form of this command to remove restore the default encapsulation type for a tunnel interface.

Use the **show** form of this command to view encapsulation configuration for a tunnel interface.

interfaces tunnel <tunx> key <key>

Defines an authentication key for a tunnel interface.

Syntax

set interfaces tunnel *tunx* **key** *key*

delete interfaces tunnel *tunx* **key**

show interfaces tunnel *tunx* **key**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   tunnel tun0..tun23 {
      key: 0-999999
   }
}
```

Parameters

tunx	Mandatory. The name of the tunnel interface you are configuring. The range is **tun0** to **tun23**.
key	A key for authenticating the local endpoint to the remote endpoint. The key must match on both ends of the connection for the tunnel to be established.

Default

No key is configured; authentication is not required.

Usage Guidelines

Use this command to provide a simple password-like numeric key for authenticating tunnel endpoints to one another. For the tunnel to be established, keys must be identical at both ends of the tunnel.

Use the **set** form of this command to specify a key for the tunnel interface.

Use the **delete** form of this command to remove the key for the tunnel interface.

Use the **show** form of this command to view the key for the tunnel interface.

interfaces tunnel <tunx> local-ip <ipv4>

Sets the IP address for the local endpoint of a tunnel.

Syntax

set interfaces tunnel *tunx* **local-ip** *ipv4*

delete interfaces tunnel *tunx* **local-ip**

show interfaces tunnel *tunx* **local-ip**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   tunnel tun0..tun23 {
      local-ip: ipv4
   }
}
```

Parameters

tunx	Mandatory. The name of the tunnel interface you are configuring. The range is **tun0** to **tun23**.
ipv4	Mandatory. The IPv4 address to be used as the tunnel endpoint on the local router. The IP address must already be configured for the interface.

Default

None.

Usage Guidelines

Use this command to specify the IP address to use as the local endpoint of the tunnel. The IP address must be one of those configured using the **interfaces tunnel <tunx> address <ipv4net>** command (see page 243).

Use the **set** form of this command to set address of the local endpoint of the tunnel.

Use the **delete** form of this command to remove the local endpoint of the tunnel. Note that the tunnel will not function without both endpoints configured.

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

interfaces tunnel <tunx> mtu <mtu>

Sets the MTU size for a tunnel interface.

Syntax

set interfaces tunnel *tunx* **mtu** *mtu*

delete interfaces tunnel *tunx* **mtu**

show interfaces tunnel *tunx* **mtu**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    tunnel tun0..tun23 {
        mtu: mtu
    }
}
```

Parameters

tunx	Mandatory. The name of the tunnel interface you are configuring. The range is **tun0** to **tun23**.
mtu	Optional. The MTU, in octets, for the tunnel interface. The range is 0 to 8042, where 0 means fragmentation is never performed. The default is 1476.

Default

Tunnel interface packets have an MTU of 1476.

Usage Guidelines

Use this command to set the maximum transfer unit (MTU) for encapsulated packets traversing the tunnel.

This MTU is applied to the packets embedded in the encapsulating protocol; it is not the MTU of the “carrier” packets themselves. The MTU of carrier packets is dictated by the MTU of the physical interface transmitting and receiving the tunnel packets.

Use the **set** form of this command to set the MTU value for encapsulated packets.

Use the **delete** form of this command to restore the default MTU value for encapsulated packets.

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

interfaces tunnel <tunx> remote-ip <ipv4>

Sets the IP address for the remote endpoint of a tunnel.

Syntax

set interfaces tunnel *tunx* **remote-ip** *ipv4*

delete interfaces tunnel *tunx* **remote-ip**

show interfaces tunnel *tunx* **remote-ip**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   tunnel tun0..tun23 {
      remote-ip: ipv4
   }
}
```

Parameters

tunx	Mandatory. The name of the tunnel interface you are configuring. The range is **tun0** to **tun23**.
ipv4	Mandatory. The IPv4 address to be used as the tunnel endpoint on the remote router. The IP address must already be configured for the interface.

Default

None.

Usage Guidelines

Use this command to specify the IP address to use as the remote endpoint of the tunnel.

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

Use the **delete** form of this command to remove the remote endpoint of the tunnel. Note that the tunnel cannot be established without both endpoints configured.

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

interfaces tunnel <tunx> tos <tos>

Specifies the value to be written into the ToS byte of the tunnel packet’s IP header.

Syntax

set interfaces tunnel *tunx* **tos** *tos*

delete interfaces tunnel *tunx* **tos**

show interfaces tunnel *tunx* **tos**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   tunnel tun0..tun23 {
      tos: tos
   }
}
```

Parameters

tunx	Mandatory. The name of the tunnel interface you are configuring. The range is **tun0** to **tun23**.
tos	Optional. The value to be written into the ToS byte in tunnel packet IP headers (the carrier packet). The range is 0 to 99, where 0 means tunnel packets copy the ToS value from the packet being encapsulated (the passenger packet). The default is 0.

Default

The ToS byte of the encapsulated packet is copied into the ToS byte of the tunnel packet’s IP header.

Usage Guidelines

Use this command to specify the value to be written in the 8-bit Type of Service (ToS) byte of the IP header for packets traversing a tunnel interface. The ToS byte of a packet's IP header specifies the forwarding behavior to be applied to the packet.

Use the **set** form of this command to specify the ToS value to write into a tunnel packet's IP header.

Use the **delete** form of this command to restore the default behavior for the ToS byte.

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

interfaces tunnel <tunx> ttl <ttl>

Defines the time-to-live (TTL) value to be written into the tunnel packet's IP header.

Syntax

set interfaces tunnel *tunx* **ttl** *ttl*

delete interfaces tunnel *tunx* **ttl**

show interfaces tunnel *tunx* **ttl**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   tunnel tun0..tun23 {
      ttl: 0-255
   }
}
```

Parameters

tunx	Mandatory. The name of the tunnel interface you are configuring. The range is **tun0** to **tun23**.
ttl	Optional. The value to be written into the TTL field in tunnel packet IP headers (the carrier packet). The range is 0 to 255, where 0 means tunnel packets copy the TTL value from the packet being encapsulated (the passenger packet). The default is 0.

Default

The ToS byte of the encapsulated packet is copied into the ToS byte of the tunnel packet's IP header.

Usage Guidelines

Use this command to specify the value to be written in the TTL field of the IP header for packets traversing a tunnel interface. The TTL field of a packet's IP header used to limit the lifetime of an IP packet and to prevent indefinite packet looping.

Use the **set** form of this command to specify the TTL value to write into a tunnel packet's IP header.

Use the **delete** form of this command to restore the default behavior for the TTL field.

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

show interfaces tunnel

Displays information about tunnel interfaces.

Syntax

show interfaces tunnel [*tunx* [**brief**] | **detail**]

Command Mode

Operational mode.

Parameters

tunx	Optional. Displays information for the specified tunnel interface. The range is **tun0** to **tun23**.
brief	Optional. Displays a brief status of the specified tunnel.
detail	Optional. Displays a detailed status of the tunnel interfaces.

Default

Information is displayed for all tunnel interfaces.

Usage Guidelines

Use this command to view the operational status of tunnel interfaces.

Examples

Example 9-1 shows operational status for the GRE tunnel interface tun0.

Example 9-1 “show interfaces tunnel”: Displaying tunnel configuration

```
vyatta@vyatta:~$ show interfaces tunnel
   tun0@NONE: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1476 qdisc noqueue
   link/gre 192.168.20.2 peer 192.168.20.3
   inet 192.168.20.1/24 brd 192.168.20.255 scope global tun0
   RX:  bytes   packets   errors   dropped   overrun   mcast
          0     0         0        0         0         0
   TX:  bytes   packets   errors   dropped   carrier   collisions
          0     0         0        0         0         0
```

Glossary of Acronyms

ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
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
DLCI	data-link connection identifier
DMI	desktop management interface
DMZ	demilitarized zone
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/Ouput
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
IS-IS	Intermediate System–to–Intermediate System
ISP	Internet Service Provider
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
MAC	medium access control
MIB	Management Information Base
MLPPP	multilink PPP
MPLS	Multiprotocol Label Switching

MPLS EXP	MPLS experimental
MPLS TE	MPLS Traffic Engineering
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
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
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
Rx	receive
SNMP	Simple Network Management 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