



## **Multilink**

Version: 543

Copyright 2007-2010 ImageStream Internet Solutions, Inc., All rights Reserved.



# Table of Contents

<b>Multilink.....</b>	<b>1</b>
Introduction.....	1
Requirements.....	1
PPP Configuration.....	1
Side 1: Normal customer side.....	1
Side 2: Testing side ( pseudo telco ).....	1
Multilink Frame.....	2
Side 1: Normal customer side.....	2
Side 2: Testing side ( pseudo telco ).....	2
Configuration Option Reference.....	3
bandwidth {bits_per_second}.....	3
description string.....	3
encapsulation {frame-relay hdlc none ppp raw l25 atmlaal5snaplaal5autopp}.....	4
ip address IPv4_Address IPv4_Netmask [secondary].....	4
frame-relay interface-dlci <DLCI>.....	5
ppp multilink.....	5
ppp multilink fragmentation {enabled disabled}.....	5
multilink-group group_id.....	5
service-module t1 clock source <line   internal>.....	6



# Multilink

## Introduction

Multilink interfaces simply take two or more physical interfaces and make a logical connection out of them. For example, two T1s can be tied together into a single three Mbps interfaces.

## Requirements

Multilink interfaces require a minimum of two physical lines. In addition, the telco must support multilink as well. For testing purposes, two Imagestream routers can be configured back to back.

## PPP Configuration

Basic configuration for the physical interfaces is the same, define the clock and encapsulation. ppp multilink enables multilink, and multilink-group 1 assigns the interface to a multilink group. After defining physical interfaces, configure a virtual interface to actually work with.

### Side 1: Normal customer side

```
!  
interface Serial1  
  service-module t1 clock source line  
  encapsulation ppp  
  ppp multilink  
  multilink-group 1  
!  
interface Serial2  
  service-module t1 clock source line  
  encapsulation ppp  
  ppp multilink  
  multilink-group 1  
!  
interface Multilink1  
  ip address 192.168.4.2 255.255.255.252  
!
```

The two interfaces are configured as bare ppp. Turn on multilink, and assign each one to a group. Then create an interface MultilinkX, where X is the group assigned to the physical interfaces.

### Side 2: Testing side ( pseudo telco )

```
!  
interface Serial1  
  service-module t1 clock source internal  
  encapsulation ppp  
  ppp multilink  
  multilink-group 1
```

```

!
interface Serial2
  service-module t1 clock source internal
  encapsulation ppp
  ppp multilink
  multilink-group 1
!
interface Multilink1
  ip address 192.168.4.1 255.255.255.252
!

```

service-module t1 clock source internal is the primary difference between Side 1 and Side 2. Normally the interface gets its clock timing from the line(telco) but in back to back setups, one side must generate the clock signal on the line.

## Multilink Frame

Multilink Frame requires only encapsulation frame-relay MFRX where X is a non-negative integer. The MFRX interface is then configured as a normal physical frame relay device. All physical interfaces with MFRX configured as the encapsulation are then added to that multilink group.

### Side 1: Normal customer side

```

!
interface Serial3
  encapsulation frame-relay MFR1
!
interface Serial4
  encapsulation frame-relay MFR1
!
interface MFR1
  encapsulation frame-relay
!
interface MFR1.1
  encapsulation frame-relay ietf
  frame-relay interface-dlci 591
  ip address 192.168.5.2 255.255.255.252
!

```

Set the encapsulation on the physical interfaces to MFR1. Then configure MFR1 as a frame-relay device. Configure MFR1.1 as a frame relay sub-interface with a DLCI of 591.

### Side 2: Testing side ( pseudo telco )

```

!
interface Serial3
  encapsulation frame-relay MFR1
!
interface Serial4
  encapsulation frame-relay MFR1
!
interface MFR1
  encapsulation frame-relay
  frame-relay mode dce

```

```

!
interface MFR1.1
 encapsulation frame-relay ietf
 frame-relay interface-dlci 591
 ip address 192.168.5.1 255.255.255.252
!

```

When operating back to back, one side must have frame-relay mode dce. This tells the interface to act as the telco side of the line.

## Configuration Option Reference

### **bandwidth {*bits\_per\_second*}**

#### **Description**

Sets the intended bandwidth in bits per second. This command does not set internal clock speeds. See baud to set internal clocking for interfaces.

When used in conjunction with ATM QoS commands this value is used to calculate the correct ATM QoS settings. When this command is used on a Frame-relay interface this is used to setup rate-limiting on the interface.

#### **Parameter**

*bits\_per\_second* - Non-negative integer which represents how much data an interface can push.

#### **Examples**

**bandwidth 2000000** - Sets the interface to 2 Mbps  
**bandwidth 100000000** - Sets the interface to 100Mbps

### **description *string***

#### **Description**

Used to add a comment (description) for tracking what is attached to a particular interface.

#### **Parameter**

*string* - Any alpha-numeric characters are allowed.

#### **Example**

**description Link to CO** - Documents this interface as connecting to CO

## encapsulation

**{frame-relay|hdlc|none|ppp|raw|x25|atm|aal5snap|aal5autopp}**

### Description

Set the protocol for a serial interface.

### Parameters

*frame-relay* - Frame Relay protocol (Uses IETF standard)

*hdlc* - High-Level Data Link Control (HDLC) protocol for serial interface. This encapsulation method provides the synchronous framing and error detection functions of HDLC without windowing or retransmission

*none* - Raw character device read/write for external applications

*ppp* - Point-to-Point Protocol

*raw* - Raw IP encapsulation

*x25* - X.25 protocol

*atm* - ATM VC multiplex encapsulation

*aal5snap* - ATM Aal5snap encapsulation (valid only on an atm sub-interface)

*aal5autopp* - ATM Aalautopp encapsulation (valid only on an atm sub-interface)

### Examples

**encapsulation frame-relay** - Set the protocol on the interface to frame-relay.

**encapsulation raw** - No protocol, an external program will process the raw data from the line.

## ip address *IPv4\_Address IPv4\_Netmask* [secondary]

### Description

To set IP addresses for an interface, use the ip address command.

### Parameters

*IPv4\_Address* - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia

*IPv4\_Netmask* - Four integers from zero to 255 separated by periods

secondary - Specifies additional IP addresses (aliases)

### Examples

**ip address 10.1.1.199 255.0.0.0** - Assigns the interface the IP address 10.1.1.199 with a Class A network mask

**ip address 192.168.1.254 255.255.255.0 secondary** - Assigns the interface the alias IP address 192.168.1.254 with a Class C network mask



## frame-relay interface-dlci <DLCI>

### Description

Set the DLCI number of the frame relay link. Normally your DLCI is provided by telco, in a back to back or testing setup, any positive integer can be used.

### Parameter

<DLCI> - Identifying number assigned by telco

### Examples

*frame-relay interface-dlci 50* - Associate DLCI 50 to this interface.

## ppp multilink

### Description

Enables multilink on a device using the PPP protocol.

## ppp multilink fragmentation {enabled|disabled}

### Description

### Parameters

enabled - Turn on the feature(Default)

disabled - Turn off the feature

### Examples

**ppp multilink fragmentation disabled** - Turns off multilink fragmentation ( Common when dealing with Cisco routers )

## multilink-group *group\_id*

### Description

Sets the group number of the multilink device, this number names the multilink interface which binds the group together. i.e. group 1 would be bound to Multilink1

### Parameter

*group\_id* - Identifies the multilink group this interface is part of.

## Examples

**multilink-group 3** - Assigns the interface to group 3

**service-module t1 clock source** <line / *internal*>

## Description

Set the clock clock or line sync for an interface

## Parameter

line - Set the interface to sync from the line.

*internal* - Set the interface to generate sync on the line.

## Example

*service-module t1 clock source internal* - Sets the internal CSU/DSU to generate clock signal for the connection