



# Bridge traffic between two or more Ethernet interfaces on Linux

---

## Content

- 1 Objective
- 2 Background
- 3 Scenario
- 4 Method (non-persistent)
  - 4.1 Overview
  - 4.2 Install the bridge utilities package
  - 4.3 Create the bridge
  - 4.4 Enable STP support if required
  - 4.5 Remove any IP addresses from the Ethernet interfaces
  - 4.6 Attach the Ethernet interfaces to the bridge
  - 4.7 Bring the bridge up
- 5 Methods (persistent)
- 6 Variations
  - 6.1 Binding an IP address to the bridge
- 7 Troubleshooting
- 8 Errors
  - 8.1 Can't add ppp0 to bridge br0: Invalid argument
- 9 Further reading

## Objective

To bridge traffic between two or more Ethernet interfaces on Linux

### Tested on

Debian (Lenny, Squeeze)  
Ubuntu (Lucid)

## Background

An Ethernet bridge is a device for forwarding packets between two or more Ethernets so that they behave in most respects as if they were a single network. It could be a physical device, but it is also possible for a bridge to be implemented entirely in software. The Linux kernel has the ability to perform bridging by means of the bridge module.

## Scenario

Suppose you have a machine with two Ethernet interfaces named `eth0` and `eth1`. Their respective MAC addresses are `02:00:00:00:00:00` and `02:00:00:00:00:01`. You wish to connect them using a bridge.

## Method (non-persistent)

### Overview

The method described here has six steps:

1. Install the bridge utilities package.
2. Create the bridge.
3. Remove any IP addresses from the Ethernet interfaces.
4. Enable STP support if required.
5. Attach the Ethernet interfaces to the bridge.
6. Bring the bridge and the Ethernet interfaces up.

Bridges created using this method will not persist beyond a reboot. See below if you require a persistent configuration.

## Install the bridge utilities package

Bridging is performed by a kernel module, but a userspace package is needed to configure it. This can be found in the `bridge-utils` package on Debian-based systems:

```
apt-get install bridge-utils
```

and similarly on Red Hat-based systems:

```
yum install bridge-utils
```

## Create the bridge

The bridge can be created using the `brctl addbr` command:

```
brctl addbr br0
```

Each bridge must be given a name. In this case the name `br0` has been chosen, however it is not necessary to follow any particular naming convention provided it does not clash with another network device.

## Enable STP support if required

If there is any possibility of the bridge creating a loop in the network then STP (Spanning Tree Protocol) support must be enabled. This must be done before the bridge is brought up, and to avoid accidents, preferably before any interfaces are attached to it. STP can be enabled using the `brctl stp` command:

```
brctl stp br0 on
```

Enabling STP should always be safe, but it is not necessarily desirable because of the substantial delay that can occur between a new link being added and it being able to pass traffic. For this reason you may want to leave STP disabled in simple cases (such as when bridging a set of virtual machines to a single physical interface).

## Remove any IP addresses from the Ethernet interfaces

Once an interface has been attached to a bridge it cannot be used for other purposes. In particular it cannot be used as an endpoint for Internet protocol traffic, so if the interface has been bound to any IP

addresses then those addresses should be removed before the interface is attached to a bridge. If they are not removed then spurious entries will be left in the routing table which can disrupt connectivity.

Addresses can be removed from an interface using the `ifconfig` command:

```
ifconfig eth0 0.0.0.0 down
ifconfig eth1 0.0.0.0 down
```

IPv6 addresses are automatically removed when an interface is brought down, but IPv4 addresses is not. This is the reason for explicitly setting the IPv4 address to zero.

## Attach the Ethernet interfaces to the bridge

To be useful the bridge must have at least two interfaces attached to it. This can be done using the `brctl addif` command:

```
brctl addif br0 eth0
brctl addif br0 eth1
```

The first argument is the name of the bridge and the second argument is the name of the Ethernet interface to be attached. More interfaces can be added if required.

A common mistake when administering a machine remotely via SSH is to incapacitate the network interface that the SSH connection is using. If you have followed the procedure described above and removed any addresses bound to the interface before attaching it to the bridge then there should be no surprises when you execute the `addif` commands. If not then a loss of connectivity could occur at this point.

## Bring the bridge up

Like the Ethernet interfaces, the bridge will not become operational until it is brought into the 'up' state. This can be done for all three of these devices using the `ifconfig` command:

```
ifconfig eth0 up
ifconfig eth1 up
ifconfig br0 up
```

The bridge should now be ready for use, however there may be a delay before traffic starts to flow (typically about 30 seconds if STP is enabled or half that if not).

## Methods (persistent)

The procedure for making a bridge persistent depends on which GNU/Linux distribution you are using. See:

- [Persistently bridge traffic between two or more Ethernet interfaces \(Debian\)](#)
- [Persistently bridge traffic between two or more Ethernet interfaces \(Red Hat\)](#)
- [Persistently bridge traffic between two or more Ethernet interfaces \(SUSE\)](#)

## Variations

## Binding an IP address to the bridge

As noted above, an Ethernet interface cannot usefully have an IP address if it is also attached to a bridge. However it is possible to achieve the same effect by binding an address to the bridge itself:

```
ifconfig br0 192.168.0.1
```

## Troubleshooting

See: [Troubleshooting Ethernet bridging on Linux](#)

## Errors

### Can't add ppp0 to bridge br0: Invalid argument

An error of the form:

```
can't add ppp0 to bridge br0: Invalid argument
```

probably indicates that the specified device is not capable of carrying Ethernet frames. In this instance an attempt has been made to bridge onto ppp0, which is probably a PPP interface using IPCP to carry Internet Protocol traffic. Because this does not use Ethernet framing it cannot be used for bridging.

(It is possible to bridge traffic using PPP, but it is necessary to use a different network control protocol called BCP. At the time of writing BCP was not supported by the mainline Linux kernel, but support could be added by means of a patch.)

## Further reading

- [bridge](#), The Linux Foundation (bridge module official website)
- Uwe Böhme, [Linux BRIDGE-STP-HOWTO](#), v0.04, January 2001

Tags: [bridging](#) | [ethernet](#)

© 2010–2016 [Graham Shaw](#), some rights reserved.