# 12.2.7 Network Interface Bonding Facts

A Bonding Bandwidth On Demand Interoperability Group, referred to as bonding or aggregation, combines multiple network interfaces into a single bonding interface. A bonding interface allows you to increase throughput by increasing bandwidth and traffic load sharing. It provides high availability through redundant network connections, link integrity monitoring, and failover.

This lesson covers the following topics:

- Aggregation overview
- Installation
- Bonding configuration
- Slave configuration

## Aggregation Overview

An aggregation (bonding) combines a group of ports into a link aggregation group, or LAG. A Linux bonding driver aggregates two or more network interface controllers into a single logical bonded interface. The LAG (bonding) is referred to as the master; the network interfaces associated with the bond are referred to as slaves. The Link Aggregation Control Protocol (LACP) is used to include or remove individual links from the LAG. Be aware that:

- A scheduling algorithm defines which packets are sent along which link.
- IEEE standard for link aggregation is 802.1AX.
- All links must be identical (Ethernet 10/100/1000/10G, etc.).
- All links must be configured the same way (duplex settings, VLAN configurations, and queuing features).
- There is a maximum of eight individual links in a LAG group.
- An even number of links usually provides better load balancing.
- Bonding interfaces function according to whether the modes are hot standby or load balancing.
- Bonded devices are directly connected to a dedicated switch device, or another system, such as a computer running Linux.

## Installation

Most kernels ship with the bonding driver already available as a module and the ifenslave program installed. If your distribution does not or you need to compile bonding from source, be aware that:

- The current version of the bonding driver is available in the drivers/net/bonding subdirectory of the most recent kernel source.
- You can add bonding to the kernel with modprobe.
- When you configure the driver as a module, you will be able to pass parameters to the driver or configure more than one bonding device.
- After you build and install the new kernels and modules, install the ifenslave control utility. ifenslave should correspond to the kernel you are using.
- Options for the bonding driver are supplied as parameters to the bonding module at load time, usually specified in either the /etc/modules.conf or /etc/modprobe.conf configuration file.

## Bonding Configuration

To create a bonding interface, you give the parameters for the Linux bonding driver to the kernel bonding module at load time. They may be given as command line arguments to the **insmod** or **modprobe** command, but are usually specified in a configuration file in the /etc/sysconfig/network-scripts/ directory called ifcfg-bond*n* where *n* is a number that identifies the interface.

When bonding using active/passive, the term passive is also known as the slave.

The following table identifies parameters you can include.

| Parameters | Description | Example |
|---|---|---|
| DEVICE | Specifies a number to identify the bond. | **DEVICE=bond1** identifies the bond as bond1. |
| TYPE | Specifies driver type. | **TYPE=Ethernet** indicates an Ethernet connection. |
| STARTMODE | Specifies when the driver will be started. The options are **onboot** and **manual**.<br>• **onboot** activates the bond when the system is started.<br>• **manual** requires the boot to be started | **STARTMODE=ONBOOT** activates the bond at startup. |
| USERCTL | Allows or prohibist system users from making changes to the bond. | **USERCTL=no** allows only root to make changes. |

| | | |
|---|---|---|
| NM_CONTROLLED | Indicates whether the bond will be controlled by NetworkManager. | **NM_CONTROLLED=n**o specifies NetworkManager is not used to manage the bond. |
| MTU | Specifies the maximum transmission unit (MTU). | **MTU=9000** |
| BOOTPROTO | Determines the protocol type to initialize the device. There are three options:<br>• **dhcp** causes the system to search for the DHCP server.<br>• **bootp** indicates the boot protocol looks for a DHCP server.<br>• **static** or **none** indicates DHCP will not be used. | **BOOTPROTO=static** indicates DHCP will not be used.<br>**BOOTPROTO=dhcp** indicates that DHCP will be used. |
| IPADDR | Specifies the IP address for the bond. | **IPADDR=179.254.0.2** uses 179.254.0.2 as the IP address for the bond. |
| BONDING_OPTS | Bonding options include the following:<br>• **arp_interval s**pecifies the ARP link monitoring frequency in milliseconds.<br>• **downdelay s**pecifies the time, in milliseconds, to wait before disabling a slave after a link failure has been detected. This option is valid only for the miimon link monitor. The downdelay value should be a multiple of the miimon value or it will be rounded down to the nearest multiple. The default value is 0.<br>• **lacp_rate s**pecifies the rate to transmit LACPDU (LACP data unit) packets in 802.3ad mode. Slow, or 0, transmits LACPDUs every 30 seconds. Fast, or 1, transmits LACPDUs every second. Slow is the default setting.<br>• **maxbonds** specifies the number of bonding devices to create for this instance of the bonding driver.<br>• **miimon** specifies the MII link monitoring frequency in milliseconds. This determines how often the link state of each slave is | Use **BONDING_OPTS=("mode=0 miimon=100 downdelay=400("** to use rr for load balancing, activate link monitoring every 100 milliseconds, and wait 500 milliseconds before disabling a failed link.<br><br>**Use BONDING_OPTS=("mode=1 primary=eth0 LACPDU=1 maxbonds=8("** to specify active backup, eth0 as the primary slave, a maximum of 8 bonds, and transmit LACPDU packets each second. |

inspected for link failures. A value of zero disables MII link monitoring.

- **mode** specifies if the links function as either hot standby or load balancing services. The behavior of the single logical bonded interface is specified by the bonding driver mode. The default parameter is balance-rr. Modes can be specified by numbers as follows:
    - **mode=0** Balance Round Robin (balance-rr): this mode transmit network packets in sequential order from the first available network interface (NIC) slave through the last. This mode provides load balancing and fault tolerance.
    - **mode=1** Active backup: only one NIC slave in the bond is active. A different slave becomes active if the active slave fails. This mode provides fault tolerance.
    - **mode=2** Balance XOR: this mode transmits network packets based on a hash of the packet's source and destination. The same NIC slave is used for each destination MAC address, IP address, or IP address and port combination. This mode provides load balancing and fault tolerance.
    - **mode=3** Broadcast: this mode transmits network packets on all slave network interfaces. This mode provides fault tolerance.
    - **mode=4** 802.3ad Dynamic link aggregation: each aggregation group shares the same speed and duplex settings. This mode is similar to the XOR mode and supports the same

| | | |
|---|---|---|
| | balancing policies. The link is set up dynamically between two LACP-supporting peers.<br>◦ **mode=5** Adaptive transmit load balancing (balance-tlb): This mode does not require any special network switch support. The outgoing network packet traffic is distributed according to the current load on each network interface slave.<br>◦ **mode=6** Adaptive load balancing (balance-alb): this mode includes balance-tlb plus receive load balancing (rlb) for IPV4 traffic. It does not require any special network switch support. The receive load balancing is achieved by ARP negotiation.<br>• **primary** specifies which slave is the primary device. The primary device will always be the active slave while it is available. Only when the primary is offline will another device be used.<br>• **updelay** specifies the time, in milliseconds, to wait before enabling a slave after a link recovery has been detected. The updelay value should be a multiple of the miimon value. | |
| SLAVE | Identifies the slave interfaces for the bonding. | **BONDING_SLAVE0=eth0** indicates the interface name. eth0 is a slave in the bond. **BONDING_SLAVE1=bus-pci-0000:06:08.1** indicates that the interface name bus-pci-0000:06:08.1 is a slave in the bond. |

Follow these guidelines when configuring a bonding:

- Configure the LAG on both sides of the link.
- Set the interfaces on both sides of the link to the same speed.
- Configure and apply firewall filters on a LAG (optional).

- Use IP addresses in the load balancing algorithm for better performance when all communication is between two devices, such as a router and a firewall.

## Slave Configuration

Bonding also requires configuration file for bonding interfaces, or slaves. These configuration files are in the /etc/sysconfig/network-scripts/ directory. Create a file named ifcfg-*xxxn* where *xxx* is the type of interface and *n* is a number that identifies the interface. For example, ifcfg-eth0 identifies the first Ethernet interface. The following table identifies parameters you can include in the file:

| Parameter | Description | Example |
|---|---|---|
| DEVICE | Specifies a name to identify the interface. | **DEVICE=eth0** identifies the first Ethernet interface created. |
| USERCTL | Allows or prohibit system users from making changes to the interface. | **USERCTL=no** allows only root to make changes. |
| ONBOOT | Specifies whether the interface will be started at boot. | **ONBOOT=yes** starts the interface when the system is booted. |
| MASTER | Identifies the master associated with the slave interface. | **MASTER=bond0** indicates the master for the slave is bond0. |
| SLAVE | Specifies that the interface is a slave. | **SLAVE=yes** indicates the interface is a slave. |
| BOOTPROTO | Determines the protocol type to initialize the device. The protocol must be the same of the protocol specified for the master. There are three options:<br>• **dhcp** causes the system to search for the DHCP server.<br>• **bootp** indicates the boot protocol looks for a DHCP server.<br>• **static** or **none** indicates DHCP will not be used. | **BOOTPROTO=static** indicates DHCP will not be used<br>**BOOTPROTO=dhcp** indicates that DHCP will be used. |