Linux Enumeration of NICs
Version 4.1

By Robert Hentosh <*robert_hentosh@dell.com*>, Matt Domsch <Matt_Domsch@dell.com> and Narendra K <narendra_k@dell.com> . September 2009

Abstract

Linux naming of the hardware network interfaces may not align with BIOS and chassis labeling of the Ethernet ports. This is seen on Dell PowerEdge 1950, 1955, 2900, 2950 and newer servers when using a Linux 2.6 kernel-based product such as Red Hat Enterprise Linux 4, 5 and Novell/SuSE Linux Enterprise Server 9, 10 and 11. Solutions to align the Linux name with the expected name are presented.

Introduction

System administrators may expect the onboard network ports labeled Gb1, Gb2, etc... on a system to be assigned interface names eth0, eth1, etc... respectively. This would be one possible naming convention for network interfaces; however no industry standards currently exist to ensure such a convention. The method used in Linux to determine the number associated with a network port (i.e. eth0, eth1, eth2…) is complex and changed with the 2.6 and newer kernels.

The naming disconnect can be observed at two different times:

1. Install Time - OS Installation when network booting (e.g. using a PXE installation procedure);
2. System Runtime – Following OS installation

Dell engineers have developed workarounds for Red Hat Enterprise Linux 4 (RHEL4), 5 (RHEL5), and Novell/SuSE Linux Enterprise Server (SLES) 9, 10 and 11 for each of the above scenarios.

While the naming disconnect directly affects Dell PowerEdge 1950, 1955, 2900, 2950 and newer servers running Linux 2.6 kernels (eg. RHEL4, RHEL5 , SLES9, SLES10 and SLES 11), this behavior is not unique to Dell systems. The workarounds and solutions presented here work on Dell systems, and may work on additional systems.

The naming disconnect issues are not observed on Dell servers when using 2.4 kernel based distributions such as RHEL2.1, RHEL3, Red Hat Linux 7.2 and 7.3, or SLES8.

Background

How Linux assigns network interface names

The default name for Ethernet interfaces is based upon how Linux initializes them during device discovery. As Linux finds the network devices it will start numbering them starting with 0 and increasing sequentially. Device discovery is dependent on the device driver load order, PCI bus topology and the device driver code.

On RHEL4, the device driver load order is determined by the */etc/modprobe.conf* file. Device drivers assigned lower interface numbers in that file are loaded first. On RHEL5 device drivers are loaded by the

hotplug subsystem, i.e. by udev, in parallel and it affects the assignment of names. When Linux loads a single device driver it will initialize and find all devices supported by that single driver first.

On SLES, device drivers are loaded based on the PCI bus topology as discovered by the kernel. The PCI bus topology is composed of buses, bridges and devices. PCI devices must be connected to a PCI bus. The PCI buses are connected by PCI bridges to either other buses or to the system. The topology of the system can be viewed as a tree. Using this analogy, the devices are leaves, the system is the trunk, buses are branches and bridges exist where branches meet each other or the trunk. Searching for PCI devices in a system is accomplished by "walking the tree". The method of walking the tree was modified in the 2.6 and later kernels, thus changing the order in which devices are found.

And last, each device driver will search the PCI tree for all the devices it supports. Some drivers have a list of different devices it will support and search the tree for each device in the list. Other drivers will scan the tree and, for each device, see if it is in its list of supported devices. This will also change order of how devices are found and thus its interface name.

Changes in system configuration will also result in a different enumeration order. If a new network card is inserted into a PCI slot, its new position could be between two previous network devices. This may result in the new card taking over the name of a previous card in the system.

The root cause of NIC enumeration mismatches is that there is currently no industry standard to enable the OS to determine the physical labeling of Ethernet ports on the motherboard.

Workarounds

Workarounds have been identified for both install time and runtime NIC enumeration issues.

Install Time

At install time, the network interface used to download the OS kernel and initial ramdisk from the PXE server may not be the *default* interface (i.e., eth0) used by the OS installer to download the rest of the OS image. This leads to the installer trying to use the “wrong” interface to finish the download, often leading to a failed installation.

RHEL and SLES installers have long had the ability to be told *which* interface to use to download the rest of the image. On systems where the installer’s default interface isn’t the “right” interface, you can use these kernel command line options to resolve it:

- For RHEL4 Update 3 and higher, and in RHEL5, when using a PXE server, the NIC used for the initial PXE download can be used for the rest of the OS installation as well. In your PXE server’s configuration files, use these options:
  ```
  IPAPPEND 2
  APPEND ksdevice=bootif
  ```
  The OS installer then uses the same interface, regardless of its name, to finish the download. *Note:* your PXE server must use syslinux-2.07 or higher. RHEL3 provides syslinux-2.06 which does not have the IPAPPEND 2 option. RHEL4 and higher, SLES9 and higher have a sufficient version of syslinux.
- RHEL 4 Gold through Update 2, pass on the kernel command line:
  ```
  ksdevice=eth1
  ```
  This causes the installer to use eth1 instead of the default eth0. RHEL has the additional option:
  ```
  ksdevice=link
  ```

This causes the installer to use the first network device it finds that has "link" – i.e. is plugged into a network switch.

- For SLES 9 or 10, pass on the kernel command line:
  `netdevice=eth1`
  This causes the installer to use eth1 instead of the default eth0. This is not needed for SLES 11.

Of course, instead of the above, one could disable or remove all but one NIC in the system, such that there is only one NIC which the kernel will name eth0.

System Runtime

name_eths

Dell developed a program, **`name_eths`**, which reduces the complexity of the solutions presented here by automatically implementing the specific mechanisms outlined for each OS version. We strongly recommend you use **`name_eths`** to accomplish these tasks. You can find the program at `http://linux.dell.com/files/name_eths/`.

Name_eths must be run once, and run again if network cards are added to or removed from the system.

Name_eths presently can configure network device names on RHEL3, 4 and 5, Fedora Core 6 and earlier, SLES9, SLES10. For other Linux distributions, use one of the manual workarounds below.

Manual workarounds

A system administrator can use the Ethernet hardware address (MAC address) of a network device to assign a specific network interface name to a network port. Each Ethernet device has a unique hardware address associated with it. MAC addresses are normally hexadecimal numbers written in the form "`XX:XX:XX:XX:XX:XX`".

RHEL
Red Hat Enterprise Linux 3, 4 and 5 contain configuration files */etc/sysconfig/network-scripts/ifcfg-<network interface name>*. These files contain a line, `HWADDR=`, which specifies the hardware address of each port. This line can be modified, or added, if missing, to assign a specific network interface name to a specific network port. The **`name_eths`** program will do this for you.

SLES 9
SLES9 uses the configuration file in /etc/sysconfig/network/ifcfg-eth-id-<mac address> for each Ethernet port. Each file may contain a `PERSISTENT_NAME=` line which specifies the name to be given to the device. One difficulty with this is that, for SLES9 releases up to and including Service Pack 3, one cannot assign names to swap "eth0" and "eth1". Instead, you must rename both devices into another name space, e.g. "eth-external" and "eth-internal". By modifying a few of the startup scripts, this restriction can be lifted and you can use "eth0" and "eth1" as expected. The **`name_eths`** program provides these updated scripts and will modify the config files for you.

SLES 10
SuSE Linux Enterprise Server 10 uses the configuration files in */etc/udev/rules.d/* subdirectory to associate a hardware address with a Linux Ethernet device name. A configuration line in the file, *30-net_persistent_names.rules*, can be modified, or added, if missing, to assign a specific network interface name to a network hardware address associated to a specific network port. Below is an example line:

```
SUBSYSTEM=="net", ACTION=="add", SYSFS{address}=="00:02:b3:5b:75:f3",
IMPORT="/lib/udev/rename_netiface %k eth0"
```

The **`name_eths`** program will do this for you.

Read */usr/share/doc/packages/sysconfig/README.Persistent_Interface_Names* for more information.

An administrator can prevent network devices from being renamed when new Ethernet cards are added to a system by using the above methods. It will also allow the administrator to reconfigure the Linux device names to coincide with the names physically labeled on the machine.

SLES 11
On SLES 11 Ethernet interfaces are named as per the `/etc/udev/rules.d/70-persistent-net.rules` file. This file is generated when a new network device is detected, either during OS installation, or added to the system later. Users may edit this file as necessary to force device naming based on its MAC address.

Here is a sample `/etc/udev/rules.d/70-persistent-net.rules` file:

```
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="00:15:17:0b:1b:21", ATTR{type}=="1", KERNEL=="eth*",
NAME="eth0" ( Observe that this is the addon card )
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="00:19:b9:e1:3d:16", ATTR{type}=="1", KERNEL=="eth*",
NAME="eth1" (LOM1)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="00:19:b9:e1:3d:18", ATTR{type}=="1", KERNEL=="eth*",
NAME="eth2" (LOM2)
```

Notice here that the add-on card is being named "eth0", while the network ports labeled Gb1 and Gb2 on the chassis are named “eth1” and “eth2”.

“biosdevname” based solution:
The udev framework can integrate external tools like `biosdevname` (http://linux.dell.com/projects.shtml#biosdevname) that can provide a custom name to devices. `biosdevname` is a tool developed by Dell which provides the BIOS-given name of a device when supplied with the kernel-given name. The biosdevname rpm on SLES 11 installs `/sbin/biosdevname` and `/etc/udev/rules.d/71-biosdevname.rules`. The rule looks like this:

```
PROGRAM="/sbin/biosdevname --policy=all_ethN -i %k", ENV{INTERFACE_NAME}="%c"
```

When SLES11 is installed on a system:

1. `71-biosdevname.rules` is invoked and is supplied with the kernel given name of the device.
2. The rule results in biosdevname getting invoked and biosdevname exports what it thinks the device name is.
3. Then, `/lib/udev/rules.d/75-persistent-net-generator.rules` invokes `/lib/udev/write_net_rules` which writes this name to `/etc/udev/rules.d/70-persistent-net.rules`.

`lib/udev/rules.d/75-persistent-net-generator.rules` will look like this:

```
# PCI device 0x14e4:0x164c (bnx2) (custom name provided by external tool)
      SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
      ATTR{address}=="00:19:b9:e1:3d:16", ATTR{type}=="1", KERNEL=="eth*",
      NAME="eth0" (LOM1)

# PCI device 0x14e4:0x164c (bnx2) (custom name provided by external tool)
```

```
      SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
      ATTR{address}=="00:19:b9:e1:3d:18", ATTR{type}=="1", KERNEL=="eth*",
      NAME="eth1" (LOM2)

# PCI device 0x8086:0x107d (e1000e) (custom name provided by external tool)
      SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
      ATTR{address}=="00:15:17:0b:1b:21", ATTR{type}=="1", KERNEL=="eth*",
      NAME="eth2" # this is the add-on card
```

Issues with this solution:

There are a few issues with this implementation. The main issue here is `biosdevname` is not being used during installation. This leads to generation of rules with duplicate names for two Ethernet devices with different MAC addresses. For example:

```
      SUBSYSTEM=="net", ACTION=="add",
      DRIVERS=="?*",ATTR{address}=="00:1c:23:c3:5c:21", ATTR{type}=="1",
      KERNEL=="eth*",NAME="eth1"

      # PCI device 0x14e4:0x165a (tg3) (custom name provided by external tool)
      SUBSYSTEM=="net", ACTION=="add",
      DRIVERS=="?*",ATTR{address}=="00:10:18:2a:07:c1", ATTR{type}=="1",
      KERNEL=="eth*",NAME="eth1"
```

Scenario 1: Mistakes while renaming devices can cause names such as “eth1_rename_ren” to be used. SLES 11 does not install `biosdevname` by default, therefore does not use it during normal OS installation. Following OS installation, the `70-persistent-net.rules` file will contain the usual rules. Thereafter when `biosdevname` is installed and it attempts to assign names to devices, it may assign a name which conflicts with a name already in `70-persistent-net.rules`. This leads to a name clash and results in names such as `eth1_rename_ren` being assigned.

Scenario 2: Addition of a NIC into a PCI slot leads to changed names.
If devices are named eth*X* (where *X* is a digit) consecutively (e.g. eth0, eth1, eth2, eth3), and thereafter a new network card is added to the system into a PCI slot, `biosdevname` may suggest that the names of all the devices should change.

Workaround for this issue:

1. This can be worked around by deleting the `/etc/udev/rules.d/70-persistent-net.rules` file every time a NIC is added to the system and on the next reboot `/etc/udev/rules.d/70-persistent-net.rules` file is generated afresh. This has the disadvantage of losing persistency.
2. This can also be worked around by recreating the boot initrd with:
    a. `/sbin/biosdevname`
    b. `/etc/udev/rules.d/71-biosdevname.rules`
    c. `/lib64/libpci.so.3.0.1`

    This would generate the correct `/etc/udev/rules.d/70-persistent-net.rules` file for the first time which would address a static scenario of cards not being moved between slots.

    Note: Above mentioned workarounds might fail in a scenario where network drivers for all the network cards present in the system are not loaded before biosdevname runs for the first time

Solution for this issue:

A clean solution to this issue is to not use the eth*X* names for devices, but instead to use a different naming convention. `biosdevname` includes several naming policies, including one which suggests that names be based on the physical location of the card (eth_s0_1, eth_s0_2, eth_s1_1) corresponding to the first and second network devices on the motherboard and the network device in PCI slot 1. By using an alternate naming policy, such renames can be eliminated.

The `biosdevname` package can be installed from the SLES 11 DVD media and is not part of default installs.

Identifying the network interface's port

The `ethtool` command can be used to discover which physical NIC port on a server (either embedded NIC or PCI-e/PCI-X NIC adapter cards) is currently associated with which Linux network device name. Run the command:

```
ethtool –p eth0
```

to cause the port for the indicated network interface device (in this example, `eth0`) to be identified in some manner. This typically results in the blinking of one or more of the LEDs associated with that Ethernet port.

Short-Term Solution

The Linux kernel starting with kernel 2.6.19-rc3 and higher has code to recognize the affected Dell PowerEdge systems and automatically renumbers the PCI devices (and thus the network ports) in ascending PCI domain/bus/device/function order (breadth-first sort). A new kernel command line parameter, pci=bfsort forces breadth-first device sorting on any system, and pci=nobfsort disables breadth-first device sorting on any system. This same fix is included in the Red Hat Enterprise Linux 4 Update 5 kernel and later Red Hat Enterprise Linux 4 updates, Red Hat Enterprise Linux 5 and later updates.

Long-Term Industry Standard Solution

At present there is no formal method by which BIOS may communicate to the operating system the names of the devices as it knows them. This leads to the above outlined naming disconnects and provided workarounds. Dell, through our participation in standards bodies such as the DMTF SMBIOS Working Group the ACPI Forum, is working towards a formal documented solution to this issue. Once adopted into the standard and implemented in both BIOS and operating system code, it is expected that a uniform and widely accepted solution will be provided to the industry.