

## Release Notes

For

**NSC DP83820 Gigabit Ethernet Adapters**

**On FreeBSD**

Document Ref. No.: NSC\_GIGA\_1.2\_FreeBSD\_RN\_1.6a

Revision History

Version	Date
1.0	07/25/01
1.1	3/22/02
1.2	7/9/02

## TABLE OF CONTENTS

<b>1.0</b>	<b>INTRODUCTION.....</b>	<b>3</b>
1.1	OBJECTIVE.....	3
1.2	REFERENCES.....	3
<b>2.0</b>	<b>SOFTWARE REQUIREMENT .....</b>	<b>4</b>
<b>3.0</b>	<b>BUILD PROCEDURE .....</b>	<b>5</b>
3.1	BUILDING THE DRIVER.....	5
3.2	BUILDING THE DRIVER WITH DIFFERENT COMPILE OPTIONS .....	5
<b>4.0</b>	<b>INSTALLATION PROCEDURE.....</b>	<b>6</b>
4.1	METHOD 1 .....	6
4.2	METHOD 2 .....	7
<b>5.0</b>	<b>UNINSTALLATION PROCEDURE .....</b>	<b>7</b>
<b>6.0</b>	<b>CONFIGURING IP ADDRESS.....</b>	<b>8</b>
6.1	METHOD 1 .....	8
6.2	METHOD 2 .....	8
<b>7.0</b>	<b>CONFIGURING VLAN.....</b>	<b>9</b>
<b>8.0</b>	<b>UNCONFIGURING VLAN.....</b>	<b>11</b>
<b>9.0</b>	<b>CONFIGURING JUMBO FRAMES .....</b>	<b>11</b>
<b>10.0</b>	<b>FEATURES/BUG FIXES INCLUDED IN THIS RELEASE OF THE DRIVER.....</b>	<b>11</b>
<b>11.0</b>	<b>CONFIGURATION PARAMETERS.....</b>	<b>12</b>
<b>12.0</b>	<b>LIMITATIONS .....</b>	<b>12</b>

## **1.0 Introduction**

This document presents release notes for National Semiconductor's DP83820 Gigabit Ethernet network adapter card, on FreeBSD 4.3 and FreeBSD 4.5 OS.

### **1.1 Objective**

The objective of this document is to describe the following:

- Installation procedure.
- Build procedure.
- UnInstallation Procedure.
- Configuring IP address, vlan, Jumbo frame.
- Features/BUG fixes done in this release
- Configuration Parameters.
- Limitations.

### **1.2 References**

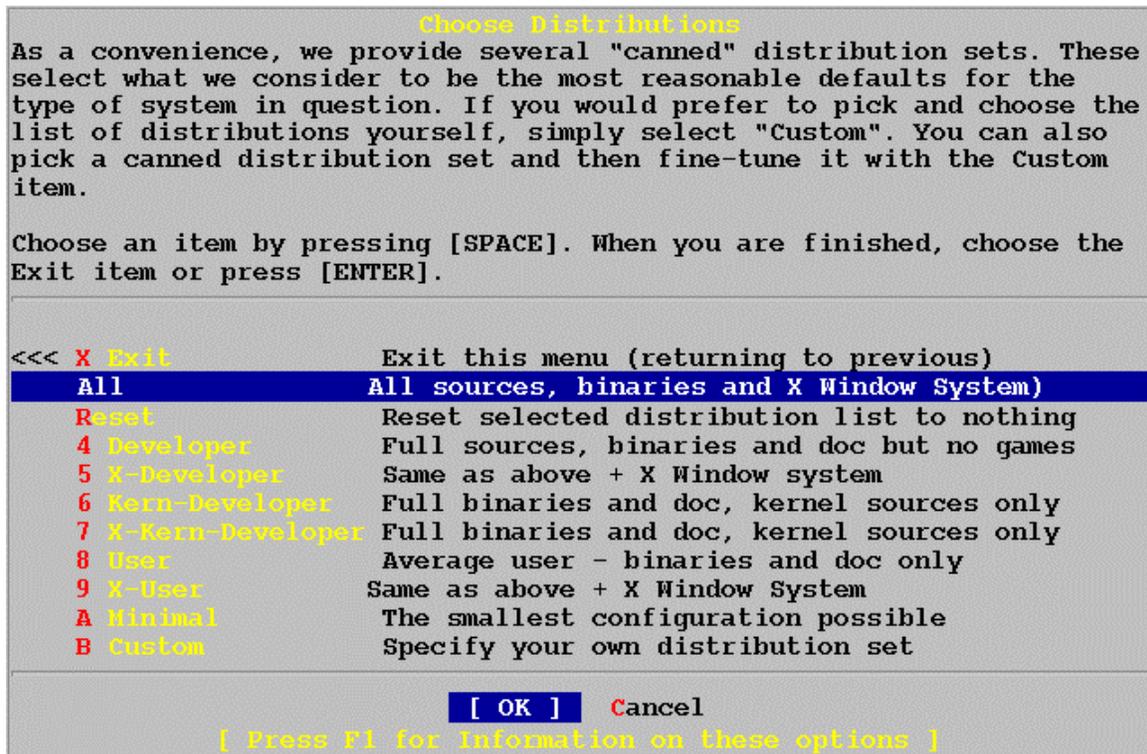
- FreeBSD 4.3 and FreeBSD 4.5 man pages.
- Comprehensive Design Description for Driver for NSC DP83820 gigabit Ethernet adapters On FreeBSD. Document Ref. No.: NSC\_GIGA\_1.0\_FreeBSD\_DD\_1.0
- TCP/IP Illustrated vol2. Authors – Gary R. Wright, W.Richard Stevens Publishers – Addison-Wesley.

## 2.0 Software Requirement

OS Version: FreeBSD 4.3 and FreeBSD 4.5

In order to build the driver, and configure vlan, Option 4 should be selected during the FreeBSD 4.3 OS installation:

The FreeBSD OS installation screen snapshot is shown in the following figure:



## 3.0 Build Procedure

### 3.1 Building the driver

Step 1: Login as root

Step 2: Untar **dp83820.tar** in the build directory  
`#tar -xvf dp83820.tar`

Step 3: Go to **dp83820** directory.  
`#cd dp83820`

Step 4: Execute the following:  
`#make clean`  
`#make`

### 3.2 Building the driver with different compile options

Step 1: Login as root

Step 2: Untar **dp83820.tar** in the build directory  
`#tar -xvf dp83820.tar`

Step 3: Go to **dp83820** directory.  
`#cd dp83820`

#### **Building the driver with debug messages:**

Edit the '**makefile**' and add the following line exactly as given below.

```
CFLAGS += -D_DP83820_ -g -DNSMDEBUG -DNSCDEBUG
```

#### **Enabling the ChecksumOffload in the NIC:**

Edit the '**makefile**' and add the following line exactly as given below.

For enabling Transmit checksum offload to the NIC HW.

```
CFLAGS += -DTXCHECKSUM -DTPCHECKSUM_HOTFIX  
-DUDPCHECKSUM_HOTFIX
```

For enabling Receive checksum offload to the NIC HW.

```
CFLAGS += -DRXCHECKSUM
```

## Vlan Feature:

Step 1: Vlan Pseudo interfaces are configured into the kernel  
(Please refer configuring vlan section 7.0)

Step2 :Edit the '**Space.h**' and change the variable name  
'**VlanInsertionMode**' according to the requirements.

[The Space.h file is there in the directory where the dp83820.tar file is extracted.  
Please look at the Space.h file for the list of tunables  
Configurable by the user at compile time]

The valid Vlan options for the variable  
'**VlanInsertionMode**' are:

- a. **VLAN\_INSERTION\_OFF** - Vlan packets are not accepted by the driver.
- b. **VLAN\_INSERTION\_NIC** - Enables the vlan tagging in the NIC. This is the default mode.
- c. **VLAN\_INSERTION\_OS** - OS will insert the VLAN Tag directly

For example:

```
VlanInsertionMode = VLAN_INSERTION_NIC;
```

**NOTE:** If vlan is not configured into the kernel then the flag **VLAN\_8021P** must be removed from the makefile **CFLAGS** definition.

Step 4: Execute the following:

```
#make clean  
#make
```

## 4.0 Installation Procedure

### 4.1 Method 1

The installation procedure described here will automatically load the driver during every system start up.

Step 1: Login as root

Step 2: Untar dp83820.tar in the build directory  
#tar -xvf dp83820.tar

Step 3: Go to **dp83820** directory.

```
#cd dp83820
```

Step 4: If user wants to re-build the driver, refer **build procedure section 3.0**.

Otherwise use the existing **dp83820.ko**

Step 5: Copy the **dp83820.ko** to **/modules**

```
#cp dp83820.ko /modules
```

Step 6: Edit the file **/boot/loader.conf** and at the end of the file, add the following line ( Use the upper or lower case as given below since it is case sensitive)

```
dp83820_load="YES"
```

Step 7: Reboot the system

```
#init 6
```

## 4.2 Method 2

The driver loaded using this installation procedure will not be loaded automatically after a system reboot.

Step 1: Login as root

Step 2: Untar **dp83820.tar** in the build directory

```
#tar -xvf dp83820.tar
```

Step 3: Go to **dp83820** directory.

```
#cd dp83820
```

Step 4: If user wants to re-build the driver, refer **build procedure section 3.0**.

Otherwise use the existing **dp83820.ko**

Step 5: Load the driver using the following command:

```
#kldload ./dp83820.ko
```

Step 6: Verify the module is successfully loaded by typing '**kldstat**' at the prompt which shows the currently loaded modules.

```
#kldstat
```

## 5.0 UnInstallation Procedure

Step 1: Login as root

Step 2: If a VLAN interface has been configured, then first unconfigure the VLAN interface associated with this NIC as explained in Section 8, otherwise jump to Step 3.

Step 3: Unload the driver using the following command:

```
#kldunload dp83820.ko
```

**NOTE :** Do **not** unload the driver before unconfiguring the VLAN interface. This can lead to a panic. Refer to Section 8 to unconfigure the VLAN interface associated with this NIC before unloading the driver.

## 6.0 Configuring IP address

### 6.1 Method 1

If you want to configure the IP address permanently, follow this method.

Step 1: Login as root

Step 2: Follow the procedure described in Section 4 to install the driver and confirm that the dp83820 driver is loaded by using the **kldstat** command.  
#kldstat

Step 3: Get the **<interface\_name>** by using the **ifconfig** command  
#ifconfig -a

The **<interface\_name>** will be 'dp**xx**'  
Where 'xx' will be a number.

If you have a single NIC, the interface name will be 'dp**0**'

Step 4: Edit the **/etc/rc.conf** file and add the following line at the end of the file.  
ifconfig\_<interface\_name>="inet <ip\_address> netmask 255.255.255.0"

Where **<ip\_address>** is the IP address to be assigned.

Step 5: Reboot the system  
#init 6

### 6.2 Method 2

The IP address configured using this method will only be effective till system reboot.

Step 1: Login as root

Step 2: Follow the procedure described in Section 4 to install the driver and confirm that the dp83820 driver is loaded by using the **kldstat** command.  
#kldstat

Step 3: Get the **<interface\_name>** by using the **ifconfig** command  
#ifconfig -a

The **<interface\_name>** will be 'dp**xx**'  
Where 'xx' will be a number.

If you have a single NIC, the interface name will be 'dp**0**'

Step 4: Execute the following command:  
#/sbin/ifconfig <interface\_name> <ip\_address>

## 7.0 Configuring VLAN

Step 1: Login as root

Step 2: Building a new kernel  
Before you can create and use the VLAN device, you must configure the kernel to use the VLAN device driver.

Change the directory to where the kernel config file resides in  
**/usr/src/sys/i386/conf**

```
#cd /usr/src/sys/i386/conf
```

copy the **GENERIC** kernel config file to another file called **VLAN**.  
#cp GENERIC VLAN

Edit the file **VLAN** and add the following line exactly as shown below at the end of the file.

```
pseudo-device vlan 2 #Pseudo vlan interfaces
```

(This will add support for 2 VLAN interfaces, however, you can add more if you wish by changing the value '2' to the number of interfaces you want to support.)

Then execute the following commands at the prompt.

```
#/usr/sbin/config VLAN
```

The '**config**' command builds a set of system configuration files.

If any error messages are produced by '**config**' the problems in the configuration file should be corrected and '**config**' command should be run again.

If '**config**' command is successful then the source directory for the new kernel to be compiled will be specified.

It will also prompt you to do a '**make depend**' command

Go to build directory

```
#cd ../../compile/VLAN
```

Compile the new kernel this will take some time so please be patient.

```
#make depend
```

```
#make
```

Installs the new kernel

```
#make install
```

Reboot the system

```
#init 6
```

Step 3: Configuring the VLAN interfaces

Once you have recompiled the kernel and rebooted the system, you need to configure the VLAN interface:

The format for configuring VLAN is  
`#ifconfig vlanxx vlan yy vlandev <interface_name>`

where:

**vlanxx** is the name of the VLAN interface you want to configure and **xx** is the number used to identify the VLAN interface. The VLAN interfaces present in the system can be viewed by issuing the 'ifconfig' command.

`#ifconfig -a`

**yy** is the VLAN Tag to be associated with the interface.

**<interface\_name>** is the name of the NIC on which you want to configure VLAN. This can be found by following the instructions in Section 6.1 Step 3.

For Example:

`#!/sbin/ifconfig vlan0 vlan 2 vlandev dp0`

This configures the interface **vlan0** on the NIC **dp0** with a VLAN tag of **2**.

Step 4: The next step is to assign it an IP Address.

`#!/sbin/ifconfig vlanxx inet <ip_address> netmask 255.255.255.0`

where:

**vlanxx** is the VLAN interface as explained above.

**ip\_address** is the IP Address that you want to assign to the VLAN Interface.

For Example:

`#!/sbin/ifconfig vlan0 inet 192.168.204.65 netmask 255.255.255.0`

Step 5: Now enable the NIC by issuing the command

`#!/sbin/ifconfig <interface_name> up`

where :

**<interface\_name>** is the NIC interface as explained above.

For Example:

`#!/sbin/ifconfig dp0 up`

Step 6: while building the driver with vlan feature section 3.2.

If Vlan option is **VLAN\_INSERTION\_NIC** then we must issue the following command to enable the vlan interfaces.

`#!/sbin/ifconfig vlanxx link0`

where :

**vlanxx** is the VLAN interface as explained above.

For Example:

`#!/sbin/ifconfig vlan0 link0`

**NOTE :** Execute Step 6 **only** if **VLAN\_INSERTION\_NIC** has been used for initialization during the building of driver section 3.2.

## 8.0 UnConfiguring VLAN

Step 1: Login as root

Step 2: To unconfigure the vlan interface execute the following command.

```
#!/sbin/ifconfig vlanxx -vlandev <interface_name>
```

where:

<interface\_name> is the NIC interface as explained in Section 7 Step 3.

**vlanxx** is the VLAN interface as explained in Section 7 Step 3.

For Example:

```
#!/sbin/ifconfig vlan0 -vlandev dp0
```

## 9.0 Configuring Jumbo Frames

Step 1: Login as root

Step 2: Set the Jumbo Frame size for an interface using the following command.

```
#!/sbin/ifconfig <interface_name> mtu <size>
```

where:

<interface\_name> is the NIC interface as explained in Section 7 Step 3.

<size> is the size of the Jumbo Frame in bytes to set.

For Example:

```
#!/sbin/ifconfig dp0 mtu 4096
```

This will set the Jumbo Frame size for the interface **dp0** to **4096** bytes.

**NOTE:** The maximum Jumbo Frame size that can be set is 16370 bytes.

## 10.0 Features/Bug Fixes included in this Release of the driver

- Multiple NIC support(No limit)
- TBI NIC support with Activity LED control.
- Transmit and Receive Checksum off-load for IP, TCP and UDP packets
- Software hot fix for checksum for TCP and UDP Packets
- 8 Byte fix for latency timer problem in the NIC.
- Multiple Vlan
- Jumbo frame upto 16K.

## 11.0 Configuration Parameters

SL No	Parameter	User Configurable/Level	Configuration Method
1	Speed	Yes/Runtime	ifconfig
2	Duplex	Yes/Runtime	ifconfig
3	Vlan configuration	Yes/Runtime	ifconfig( NOTE: since the NIC is capable of inserting and extraction of vlan tags on its own the vlan interface must be set to LINK0)
4	MTU	Yes/Runtime	ifconfig(NOTE: Receive buffers are also changed according to the MTU size, the maximum MTU we can change is 16k)
5	Jumbo Frames	Yes/Runtime	The Maximum Receive frame size supported will be same as MTU if MTU is changed at runtime. (Refer to item 4 in this table).
6	Tx/Rx Descriptor Count	Yes/Compile time	The file Space.h has the list of tunable parameters.
7	NCBIT setting	Yes/Compile time	Same as above
8	VlanInsertionMode	Yes/Compile time	Enabling/Disabling of vlan tagging in the NIC, Please refer the build procedure sec 4.2

## 12.0 Limitations

The v1.2 release of DP83820 driver does not support the following features:

- Checksum Off-loading.
- Setting the following User configurable parameters:
  - Transmit buffer count
  - Receive buffer count

- Max packet size
- Power Management.