

Configuring NIC and Setting Ethernet Interface Speed on Sun Solaris 10

Initial Configuration of the NIC

The network adapter can be brought up for use with the "plumb" switch in the ifconfig command. This is done by using:

```
ifconfig <NIC> plumb (or deplumb for the reverse)
ifconfig <NIC> ip <ip address> netmask <subnet mask> broadcast <broadcast address>
ifconfig -a {verify the NIC properties}
```

Setting NIC speed and duplex

Solaris is often unable to correctly auto-negotiate duplex settings with a link partner (e.g. switch), especially when the switch is set to 100Mbit full-duplex. You can force the NIC into 100Mbit full-duplex by disabling auto-negotiation and 100Mbit half-duplex capability.

Check Current State

To check the link status for every available network adapters on the system, as root use:

```
dladm show-dev
netstat -in # shows packet information, look for the Collis (Collisions)
column, if there is anything more than 0, then you know there is a problem
```

Test with Live Change

Make the changes to the running system as a test.

```
netstat -an # to determine your interface name, in this example it is bge0}

# You run these command only on the console.

ndd -set /dev/bge0 adv_100hdx_cap 0
ndd -set /dev/bge0 adv_100fdx_cap 1
ndd -set /dev/bge0 adv_autoneg_cap 0 # people on the internet seem to
indicate that we should use autoneg_cap last
```

Make the Changes Permanent

Full confusing details from Sun here, <http://docs.sun.com/app/docs/doc/805-4449/6j47dm3ih?l=en&q=system+hme&a=view>.

To set parameters so they remain in effect after you reboot the system there are at least 3 ways of doing so,

1. Use a boot script using ndd - can specify particular device.
2. Edit /kernel/drv/[device name].conf - Solaris 10 only. Looks like effects all devices.
3. Configure /etc/system - effects all devices in the system.

There are different types of devices,

- hme interface
- eri interface
- ce interface
- bge interface
- e1000g interface

Researching..

<http://www.google.ca/search?hl=en&q=e1000g+full+duplex+&meta=>
<http://docs.sun.com/app/docs/doc/816-5177/e1000g-7d?a=view>
<http://fixunix.com/solaris/142492-forcing-duplex-e1000g.html>

***** content below still being finalized *****

vi /etc/system

Add to the bottom,

```
set hme:bge0_adv_autoneg_cap=0
set hme:bge0_adv_100hdx_cap=0
set hme:bge0_adv_100fdx_cap=1
```

Or...

```
set bge:bge0_adv_autoneg_cap=0
set bge:bge0_adv_100hdx_cap=0
set bge:bge0_adv_100fdx_cap=1
```

Or...

```
set bge0:bge0_adv_autoneg_cap=0
set bge0:bge0_adv_100hdx_cap=0
set bge0:bge0_adv_100fdx_cap=1
```

Or...

```
set bge:bge_adv_autoneg_cap=0
set bge:bge_adv_100hdx_cap=0
set bge:bge_adv_100fdx_cap=1
```

Ok none of this except the manual method using ndd seem to work on the new T5120 (talk to Dimitri or Dickson Tin)

Method 1 (Dickson) - requires a reboot

Modify network configuration file appropriate to your kernel,

```
cd /platform/`uname -i`/kernel/drv/bge.conf {the command uname -i shows your kernel name}
```

* note the filename **bge.conf** will also depend on the hardware. using **dladm show-dev** determine the file name to use. For the e1000 then use filename e1000g.conf

This change affects all network adapters of the same manufacturer on the system.

```
adv_autoneg_cap = 0;
speed = 100;
full-duplex = 1;
```

Reboot the system,

```
shutdown -i6 -g0 -y
```

Method 2 (Dimitri) - works best for live systems

Example with bge0 interface.

1. Make the changes to the running system.

`nestat -an {to determine your interface name}`

You run these command only on the console.

`ndd -set /dev/bge0 adv_100hdx_cap 0`

`ndd -set /dev/bge0 adv_100fdx_cap 1`

`ndd -set /dev/bge0 adv_autoneg_cap 0`

2. Make kernel parameter changes to preserve the speed and duplex settings after a reboot.

`vi /etc/system`

Add:

`set hme:bge0_adv_autoneg_cap=0`

`set hme:bge0_adv_100hdx_cap=0`

`set hme:bge0_adv_100fdx_cap=1`

I don't understand below. At this point is not everything done? Tin

The /etc/system settings listed above are in the /platform/sun4u/kernel/drv/bge.conf file

Example: /etc/init.d/nddconfig

The contents of the file should be,

`#!/bin/sh`

`ndd -set /dev/bge0 instance 0`

`ndd -set /dev/bge0 adv_1000fdx_cap 0`

`ndd -set /dev/bge0 adv_1000hdx_cap 0`

`ndd -set /dev/bge0 adv_100fdx_cap 1`

`ndd -set /dev/bge0 adv_100hdx_cap 0`

`ndd -set /dev/bge0 adv_10fdx_cap 0`

`ndd -set /dev/bge0 adv_10hdx_cap 0`

`ndd -set /dev/bge0 adv_autoneg_cap 0`

Make a symbolic link,

```
ln -s /etc/init.d/nddconfig /etc/rc2.d/S31nddconfig
dmesg | grep bge0 # Make sure the network card shows up by searching in
display message
```