STORDIS
The Open Networking Expert

Getting started with STORDIS BF6064X and the Barefoot SDE

Contents

Overview

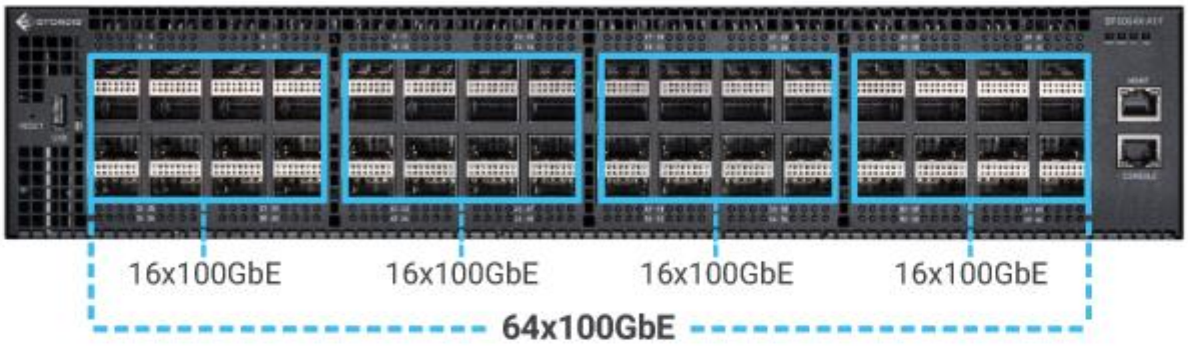
How you will receive your switch

The Stordis BF6064X comes with ONIE (Open Networking Installation Environment) preinstalled. This will serve as a basis to install a real operating system (OS) on the switch. It is not an OS itself.

To use the switch, an OS needs to be installed on it. You can either install a complete OS like SONiC or just a base OS to run your own P4 programs compiled on the switch with Barefoot's SDE.

This guide will focus on installing Barefoot's SDE on the switch, on top of Ubuntu server or ONL. It will also help you to get started in running a simple P4 program on the switch.

The switch from the outside



STORDIS Switch
Front view

STORDIS Switch BF6064X
Back view

Prerequisites

To be able to follow the installation steps detailed in this guide, you need the following:

- The switch, powered up (booting into ONIE)
- A serial connection from the switch's console port to your laptop and a terminal emulation program (e.g. PuTTY on Windows, minicom, picocom, screen or similar on Linux)
- A network connection to the management port of the switch, ideally with Internet access
- A copy of Barefoot's SDE installation tarball
- A copy of the BSP (Board Support Package) tarball for the switch
- A USB pen drive to put the Ubuntu or ONL installation files on

High-level overview of the installation steps

This guide will lead you through these steps to create a working installation on your switch:

1. Install either Ubuntu or ONL as the base OS
2. Install Barefoot’s SDE and BSP
3. Start the switching function with a P4 program
4. Configure ports on the switch and load the tables for your P4 program

Installation of the base OS

You can use either Ubuntu or ONL (Open Network Linux) as the base to install Barefoot's SDE.

ONL is easier to install because it can be installed using ONIE while Ubuntu will overwrite ONIE. On the other hand, some additional steps need to be done to make ONL work as expected.

Installation of Ubuntu server

This will resemble a normal server installation for Ubuntu, except you have to update grub.cfg to properly display output on the serial console. These steps have been tested with Ubuntu server version 16.4.4 LTS.

Make sure that you have a working serial console connection to your switch and that its management port is connected to your network.

1. Download the ISO image for Ubuntu server 16.4.4 LTS
2. Write it to a bootable USB stick (use e.g. Rufus on Windows or dd on Linux), use FAT32 format and ISO mode so you will be able to modify files on the stick
3. On the USB stick, edit file `boot\grub\grub.cfg` to make the following modifications:
    a. Under all the `menuentry` stanzas, there's a line starting with "`linux`".
    b. In these lines, replace the "`quiet`" keyword with "`console=ttyS0,115200n8`"
    c. Also append the same "`console=ttyS0,115200n8`" at the end of the line (after the `---`)
    d. It doesn't have to be done for all the entries, but mostly for the first `menuentry "Install Ubuntu Server"`. Basically, the entries that are modified will work later on.
    e. Save the file
    f. Unmount / eject the stick from your system
4. Insert the USB stick into the USB port on the switch
5. Reboot or power up the switch; when the prompt appears, press <Esc> or <Del> to enter the BIOS setup menu
6. When the Ubuntu Server Installation menu appears, choose Install Ubuntu server and follow the installation steps
7. On the "Primary Network Interface" menu, choose `ens1` which is the management port
8. If the Installer asks whether to install in UEFI mode, confirm it
9. Optionally, you can set up the disk partitions according to the recommendation in appendix A
10. For the "Software Selection", tick *OpenSSH server* to be able to SSH to the switch later
11. After successful install, it is recommended to update the installation to the latest versions (using apt-get). **Important:** if you first install the SDE and later update Ubuntu, you may run into problems if the kernel version changes. In that case, the SDE should be rebuilt again (at least the bf-drivers component)
12. Install pip (required for gRPC component in the SDE): `sudo apt-get install python-pip`
13. Reboot your system and test if the console access still works (there is output after "Booting in blind mode"). If it doesn't, follow the steps in appendix B to update GRUB again permanently

Installation of ONL

ONL can be installed with a standard ONIE NOS install. There are two options: installation from a USB pen drive or installation from a network server (TFTP or HTTP).

Usually you would also use a serial console connection to the switch, but if the management port is connected and can acquire a DHCP address, it can also be reached by telnet (no userid or password).

These steps assume that there is no operating system installed yet. If there is, uninstall it first with ONIE Uninstall OS.

Note the ONL installer image isn't currently available from the ONL project page (opennetlinux.org) yet. It has to be provided by Stordis.

Installation from USB drive

1. Copy the ONL installer image to a USB pen drive. It doesn't have to be bootable. Use FAT32 format.
2. Rename the file on the USB stick as "onie-installer.bin"
3. Plug the USB drive into the switch USB port
4. If switch hasn't been booted into ONIE Install OS yet, power it up or reboot it. It should automatically boot into ONIE installation mode
5. It should automatically discover the installer on the USB drive and start installation
6. It will automatically boot into ONL after successful installation

Installation from a network server

The ONIE installer natively supports TFTP or HTTP downloads, choose whatever server is available in your network environment. This guide assumes a TFTP server, but HTTP works in the same way.

Optionally, you can copy the installer file to the switch before you do the actual install. Using TFTP, this will speed up the download significantly because you can use larger buffers for TFTP. You can download the installer like this:

```
ONIE:/#tftp -l <local file name> -r <installer file name in TFTP directory of server> -g -b 16000 <IP address or DNS name of TFTP server>
```

Follow these steps for the installation:

1. Boot into ONIE Install OS mode
2. Hit Enter to activate the console
3. Stop the ONIE auto-discovery:
   - `ONIE:/#onie-discovery-stop`
4. Start the installation with this command:
   - `ONIE:/#onie-nos-install <protocol>://<installer file>`
   - *<protocol>* can be either *tftp* or *http*, or *file* if you have copied the installer file to the switch before
   - *<installer file>* would be something like *<server-address>/<path to file on server>*, or just the name of the installer file that you have previously copied to the switch
5. The switch will automatically boot into ONL after successful installation

After installation

After the reboot, enter the standard credentials at the login prompt:

User id: `root`

Password: `onl`

You will still need the serial console connection as the management interface needs to be configured first.

The name of the management interface should be `ma1`, you can verify it by issuing an `ip address` or `ifconfig` command. Configure it as follows:

1. Add the following stanza to /etc/network/interfaces, using nano or vi editor:

   ```
   auto ma1
   iface ma1 inet dhcp
   ```

2. Save the file and exit the editor
3. Enable the interface:
   - `root@localhost:/#ifup ma1`
4. The switch will now try to acquire a DHCP address and should be reachable when it has received one

Note: to configure a static IP, add the appropriate settings in the interface stanza, refer to an Ubuntu guide on how to do it

Installation of the SDE

This guide assumes Ubuntu as the underlying OS. Any different steps pertaining to ONL are highlighted.

These instructions apply to SDE version 8.4.0

Note: At the time of this writing, the SDE doesn't work with ONL on this switch, some modifications have to be done still

Follow these steps:

1. to avoid locale problems during installation, add this to `/etc/environment` (and set/export in your current shell):

   ```
   LC_ALL=en_US.UTF-8
   LANG=en_US.UTF-8
   ```

2. transfer SDE `bf-sde-<x.x.x.x>.tgz` (or `.tar`) tarball to switch (e.g. to your home directory or any other directory that has enough space), e.g. using `scp`
3. untar file
   `tar xzf bf-sde-<x.x.x.x>.tgz` or `tar xf bf-sde-<x.x.x.x>.tar`
4. `cd` to the SDE top-level directory (`bf-sde-<x.x.x.x>`) that has just been created in the previous step
5. transfer shell scripts `sde_build.sh` and `set_sde.bash` to that directory
6. run `. ./set_sde.bash` to set up SDE environment (important: this script has to be sourced with the '.' at the beginning and not executed directly, otherwise the environment variables will not be set correctly)
7. (recommended) add setup script to shell profile (after it has been executed once, so $`SDE` environment variable is set):

   ```
   echo -e "\n# set up Capilano SDE environment\npushd $SDE >
   /dev/null\n. $SDE/set_sde.bash\npopd > /dev/null" >> ~/.bashrc
   ```

8. Make `sde_build.sh` script executable
   `chmod` 755 $SDE/sde_build.sh
9. build the SDE: (will automatically sudo and ask for password if needed).
   from `$SDE` directory, `./sde_build.sh -r`
   - Specify `-r` option to build for the "real" chip.
   - Optionally add `--no-examples` to skip building the P4 example programs

   - Optionally add `--switch-pkg p4_16` to later use the P4$_{16}$ version of `switch.p4`
   - Run `./sde_build.sh -h` to find about more options

10. takes about 40 minutes to build (without examples)
11. at the end of the process, it will again sudo and ask for password to configure the DMA memory pool

Install the BSP

The Board Support Package (BSP) contains the switch hardware specifics to supplement the SDE. The BSP package for this switch needs to be obtained from Stordis.

Follow these steps:

1. transfer BSP package tarball e.g. `bf-delta-ag9064v1-platforms-20180511.tar.gz` to the switch (to home directory)
2. untar file
   ```
   tar xzf bf-delta-ag9064v1-platforms-20180511.tar.gz
   ```
3. setup `BSP` and `BSP_INSTALL` environment variables (`BSP` = directory where the `bg-<xxxx>-platforms` directory resides (e.g. home directory), `BSP_INSTALL=$SDE_INSTALL`):
   ```
   export BSP=`pwd`
   export BSP_INSTALL=$SDE_INSTALL
   ```
4. (recommended) add environment variables to `.bashrc`: From the directory that contains the `bg-<xxxx>-platforms` directory, execute:
   ```
   echo -e "export BSP=`pwd`\nexport BSP_INSTALL=$SDE_INSTALL" >>
   ~/.bashrc
   ```
5. (recommended) set an environment variable for the platform directory that you want to install from (only used during installation in the subsequent steps):
   ```
   export BSP_PLATFORM=$BSP/bf-<xxxx>-platforms
   ```
6. Install dependencies:
   ```
   cd $BSP_PLATFORM
   ./install_pltfm_deps.sh
   ```
7. run `autoreconf` and `autoconf` in the `$BSP_PLATFORM` directory
8. Build and install the BSP
   ```
   cd $BSP_PLATFORM
   ./configure --prefix=$BSP_INSTALL --enable-thrift
   make
   make install
   ```

Start up the switching functions

These steps need to be performed after each reboot to set up the hardware and start switching. They might be added to the startup configuration (`rc.local` or similar).

Configure Transceiver signals

These commands set the LPMode (Low Power mode) signal to 0 on all ports and ResetL to 1 (a value of 0 causes the transceiver to reset).

1. install I2C drivers
   ```
   sudo modprobe i2c-dev
   sudo modprobe i2c-i801
   ```
2. to verify:
   ```
   sudo i2cdetect -l
   i2c-0    smbus          SMBus I801 adapter at
   f000           SMBus adapter
   sudo i2cdetect -y 0
   ```

```
0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- 08 -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- 44 -- -- -- 48 -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: 70 -- -- -- -- -- -- --
```

3. Set QSFP28 Port LPMode to 0

```
sudo i2cset -y 0 0x70 0x20
sudo i2cset -y 0 0x32 0xE 0x0
sudo i2cset -y 0 0x32 0xF 0x0
sudo i2cset -y 0 0x34 0x2 0x0
sudo i2cset -y 0 0x34 0x3 0x0
sudo i2cset -y 0 0x34 0x4 0x0
sudo i2cset -y 0 0x35 0x2 0x0
sudo i2cset -y 0 0x35 0x3 0x0
sudo i2cset -y 0 0x35 0x4 0x0
```

4. Set QSFP28 Port ResetL to 1

```
sudo i2cset -y 0 0x70 0x20
sudo i2cset -y 0 0x32 0x14 0xff
sudo i2cset -y 0 0x32 0x15 0xff
sudo i2cset -y 0 0x34 0xB 0xff
sudo i2cset -y 0 0x34 0xC 0xff
sudo i2cset -y 0 0x34 0xD 0xff
sudo i2cset -y 0 0x35 0xB 0xff
sudo i2cset -y 0 0x35 0xC 0xff
sudo i2cset -y 0 0x35 0xD 0xff
```

Start SDE drivers

This assumes you have built your P4 program (which is out of the scope for this document).

1. start kernel mode driver
   `sudo $SDE_INSTALL/bin/bf_kdrv_mod_load $SDE_INSTALL`
2. cd to SDE home directory
   `sde`
3. Start `bf_switchd` with your P4 program (note: no `.p4` extension must be given!)
   `./run_switchd.sh -p <p4program>`

if all goes well, after some initialization, you should receive the `bf-shell>` prompt.

You might want to check the output of the above `run_switchd.sh` command to make sure you are running on the real hardware and not on a model:

```
Operational mode set to ASIC
```

Setting up ports

To get to the `bf-sde` shell, enter `ucli` at the `bf-shell>` prompt.

The ports that you want to use need to be added to the SDE configuration (some P4 programs like `switch.p4` do this already).

You do this by entering the `ucli` command at the `bf-shell>` prompt. You will then see the `bf-sde>` prompt. Most commands can be entered at the `pm` submenu (**note:** to leave the submenu, type "`..`" (two dots). "`Exit`" will leave the `bf-sde` environment altogether):

```
bf-sde> pm
bf-sde.pm>
```

To add a port, enter its front panel number together with the lane (0-3 for breakout configuration, 0 if no breakout), the desired speed (will also define breakout configuration) and FEC mode, e.g.:

```
bf-sde.pm> port-add 1/0 100G RS
bf-sde.pm> port-add 2/2 10G NONE
```

Ports that you have added must be enabled as well:

```
bf-sde.pm> port-enb 1/0
bf-sde.pm> port-enb 2/2
```

To verify what you have added, use the `show` command (also under the `pm` submenu):

```
bf-sde.pm> show
-----+----+---+----+------+----+---+---+---+----------------+-------
---------+-
PORT |MAC |D_P|P/PT|SPEED |FEC |RDY|ADM|OPR|FRAMES RX       |FRAMES
TX       |E
-----+----+---+----+------+----+---+---+---+----------------+-------
---------+-
1/0  |31/0|188|1/60| 100G | RS
|YES|ENB|DWN|                 0|                 0|
2/2  |30/2|186|1/58| 10G  |NONE|NO
|ENB|DWN|                 0|                 0|
```

This will also show you the mapping of the physical front ports to the logical port numbers that the Tofino chip knows. The first column (PORT) shows the port number as written on the front panel, followed by the lane (for breakout configuration). If there is no breakout configuration, only lane 0 is there.

The third column (`D_P`) is the port number that the Tofino chip (and your P4 program) see. The next column shows the pipeline `P` and port number within the pipeline `PT`. Note that `P*128+PT = D_P`

The `RDY` (ready) column shows most likely NO when there is no (supported) transceiver or DAC cable in the port. You can show all transceivers with these commands:

1. go to top-level menu
   ```
   bf-sde> ..
   ```
2. go to `qsfp` submenu (note this is actually a shortcut to `bf_pltm.qsfp`):
   ```
   bf-sde> qsfp
   ```
3. type `show` command:
   ```
   bf-sde.bf_pltfm.qsfp> show
   port  id    vendor           part_num       eth   ext-eth Om-len
   Cu-Len  Qsfp-type
                                               cmpl    cmpl
   ---------------------------------------------------------------
   -----------------
     1  0x11 FINISAR CORP     FTLC9551REPM    0x80  0x02   OM3 70
   ```

```
m  50 m    Optical
----------------------------------------------------------------
-----------------
```

The ADM and OPR columns tell you if the port is enabled and operational (up or down), respectively.

You can disable (port-dis) and delete (port-del) unneeded ports as well.

Appendix A – Recommended disk partition for Ubuntu installation

This is a recommendation from Barefoot and not mandatory to set up.

This will create the following partitions on the 64GB SSD:

- 98MB EFI system partition
- 20.9 GB EXT4 partition for root file system
- 36GB EXT4 partition for /data file system
- 7GB swap area

Note: all of the partitions created below should be created at the beginning of free space

When being asked for partitioning of your disk, do the following:

1. Choose Manual as the Partitioning Method
2. Select disk sda (64GB) and confirm to create a new empty partition table
3. Select the Free Space entry under that disk and create a new partition with 98 MB in size
4. Select the “Use As” line and choose “EFI System Partition”
5. After that, select “Done setting up the partition”
6. Select the Free Space entry under that disk and create a new partition with 20.9 GB in size
7. Change the Reserved Blocks to 1%
8. Select “Done setting up the partition”
9. Select the Free Space entry under that disk and create a new partition with 36 GB in size
10. Select the Mount Point menu entry, and choose “Enter manually”. Enter “/data”
11. Change the Reserved Blocks to 0%
12. Select “Done setting up the partition”
13. Select the Free Space entry under that disk and create a new partition with the remaining space (should be 7 GB)
14. Select the “Use As” line and choose “Swap area”
15. Select “Done setting up the partition”
16. Scroll down and select “Finish partitioning and write changes to disk”
17. Done

Appendix B – Update GRUB to support serial console output

When Ubuntu reboots after installation, you will be presented with the GRUB boot selector screen. To continue to be able to use your console connection, you need to make changes to the boot options.

1. Leave the first entry (Ubuntu) selected and press "e" to edit the boot settings.
2. in the file that will open, look for the line starting with `linux` It should end with `ro`
3. to that line, append: `console=tty0 console=ttyS0,115200n8`
4. press Ctrl-X to continue booting

To make this permanent, the GRUB configuration needs to be updated as well:

- Edit `/etc/default/grub` as follows:

```
GRUB_CMDLINE_LINUX_DEFAULT=""
GRUB_TERMINAL='serial console'
GRUB_CMDLINE_LINUX="console=tty0 console=ttyS0,115200n8"
GRUB_SERIAL_COMMAND="serial --speed=115200 --unit=0 --word=8 --
parity=no --stop=1"
```

- Update grub configuration

```
# update-grub
```