Table of contents

- Linux installation guide
  - Delivery
  - Prerequisites
    - Target platform
  - Install IP Virtual Card
    - Ubuntu
    - AlmaLinux
  - Uninstall IP Virtual Card
    - Ubuntu
    - AlmaLinux
  - KBDPDK (Optional)
    - Prerequisites
    - NIC Setup
      - Intel NIC
      - NVIDIA NIC
        - Requirements
        - Drivers install
        - Enable traffic shaping narrow linear profile
    - Install KBDPDK
      - Ubuntu
      - AlmaLinux
    - KBDPDK Clean Configuration
    - Uninstall KBDPDK
      - Ubuntu
      - AlmaLinux
  - VCS configuration
    - Interactive Mode
    - Silent Mode
    - Remarks
  - Network interface configuration
    - Socket mode
    - DPDK mode
      - Nvidia/Mellanox NIC

            - Other brand NIC
                - Example
                    - Ubuntu Server:
                    - Ubuntu Desktop and Almalinux (Using NetworkManager):
    - Performance considerations
        - Socket
        - BIOS
        - CPU governor
        - Conductor configuration
        - Route
    - Licensing
    - Synchronize multiple NICs
    - Virtual machine support
        - Configuration
- PTP
    - NTP disabling
    - Firewall

Linux installation guide

Delivery

The package contains all the resources needed to run the IP virtual card.

Once this package is installed, you will be able to run any application linked to the IP Virtual Card solution.

Prerequisites

Target platform

Hardware :

- CPU speed : minimum 2.1GHz
- CPU architecture : 64 bits
- NIC bandwidth : minimum 10Gb/s

Supported OS :

- Ubuntu 20.04
- Ubuntu 22.04
- AlmaLinux 9

Drivers :

- NIC drivers must be properly installed and up to date

Install IP Virtual Card

The installation is automated using the debian (.deb) or the Red Hat (.rpm) packages.

As an alternative, a tar.gz archive is available for manual installation.

Ubuntu

This package has dependencies to `build-essential, git, python3, dlmcli`.

```
sudo apt install ./ipvirtualcard.linux.[version].deb ./dlmcli.linux.[version].deb
```

AlmaLinux

This package has dependencies to `python3-pip, python3, dlmcli`.

```
sudo dnf install ipvirtualcard.linux.[version].rpm dlmcli.linux.[version].rpm
```

The ipvirtualcard package will :

- disable NTP
- install and register `ptp4l` as a service
- install and register the VirtualCardService as a service

The dlmcli package will: - Install the licensing manager

Uninstall IP Virtual Card

To uninstall the IP Virtual Card use the following commands:

Ubuntu

```
sudo apt remove ipvirtualcard dlmcli
```

AlmaLinux

```
sudo dnf remove ipvirtualcard
```

KBDPDK (Optional)

This section is only useful if you wish to use the IP Virtual Card with the DPDK kernel bypass instead of linux sockets.

Prerequisites

> To be able to install KBDPDK, you must first install some prerequisites.

```
# Install pip packages
sudo pip3 install pyelftools distro
```

NIC Setup

Intel NIC

No special setup is required for Intel NIC.

NVIDIA NIC

Requirements

- Network card :
  - ConnectX-5 :
    - Firmware: **16.21.1000** and above.
  - ConnectX-6 :
    - Firmware: **20.27.0090** and above.
  - ConnectX-6 Lx
    - Firmware: **20.27.0090** and above.
  - ConnectX-6 Dx
    - Firmware: **20.27.0090** and above.
  - BlueField-2
    - Firmware: **18.25.1010** and above.
- Minimal kernel version :
  - v4.14

Drivers install

1. Download the latest version of MLNX_OFED for your linux distribution : https://network.nvidia.com/products/infiniband-drivers/linux/mlnx_ofed/
2. Install required packages (only for Almalinux)

```
sudo dnf install perl.x86_64
sudo dnf install tk
```

3. Extract it on the target machine
4. Install the required libraries by installing Mellanox OFED/EN:

```
sudo ./mlnxofedinstall --upstream-libs --dpdk
```

5. Verify firmware version

```
ibv_devinfo
```

6. Restart drivers

```
sudo /etc/init.d/openibd restart
#OR
sudo service openibd restart
```

Enable traffic shaping narrow linear profile

Requirement:

- Hardware : ConnectX-6 Dx and higher
- OFED Version: 5.1-2 and higher
- Firmware Version : 22.28.2006 and higher
- IP Virtual Card License : Plus

Procedure:

1. Download the latest MFT (Mellanox Firmware Tools) tool package : https://network.nvidia.com/products/adapter-software/firmware-tools/
2. Extract it on the target machine
3. Install MFT tools

```
#In mft tool extracted folder
sudo ./install.sh
```

4. Get the pci address of your nics

```
lspci | grep Mellanox
01:00.0 Ethernet controller: Mellanox Technologies MT28841
01:00.1 Ethernet controller: Mellanox Technologies MT28841
```

5. For each NIC, enable firmware functionality

```
sudo mlxconfig -d [PCI_ADDR] s REAL_TIME_CLOCK_ENABLE=1

sudo mlxconfig -d [PCI_ADDR] s ACCURATE_TX_SCHEDULER=1
#For example for nic 01:00.0
sudo mlxconfig -d 01:00.0 s REAL_TIME_CLOCK_ENABLE=1
sudo mlxconfig -d 01:00.0 s ACCURATE_TX_SCHEDULER=1
```

6. Restart your machine

Install KBDPDK

The installation of kbdpdk is done by using the provided deb or rpm package. Before running the install, a compatibility check will be executed through the deb or rpm package.

Ubuntu

```
sudo apt install ./kbdpdk.linux.[Version].deb
```

AlmaLinux

```
sudo dnf config-manager --set-enabled crb
sudo dnf install kbdpdk.linux.[Version].rpm
```

KBDPDK Clean Configuration

This script cleans the kbdpdk configuration on your system:

```
sudo python3 kbconfig_cleanup.py
```

Uninstall KBDPDK

To uninstall KBDPDK, use the following commands:

Ubuntu

```
sudo apt remove kbdpdk
```

AlmaLinux

```
sudo dnf remove kbdpdk
```

VCS configuration

The VCS configuration is done through the `ipvc_configure.py` script. This script requires Python (v3.8 and higher) to be installed on the system.

The script works in 2 different modes : interactive and silent.

Interactive Mode

In interactive mode, the script will ask you to enter the configuration parameters one by one. After entering all the parameters, the script will ask you if you want to save the configuration in a file. Then it will ask you if you want to apply the configuration to the system. By default, the configuration file is saved in the current directory and is named `ipvc_config.cfg`.

```
sudo ipvc_configure.py
```

Silent Mode

In silent mode, the script will use the configuration file passed as an argument. Using the following command, the script will apply the configuration to the system:

```
sudo ipvc_configure.py --config [CONFIG_FILE]
```

Remarks

In dpdk mode, when using IOMMU on the machine, you must configure all ports of the NIC you want to use. Even if you only want to only use a subset of this NIC ports.

Network interface configuration

It is the responsability of the user to configure the network interfaces used by the IP Virtual Card.

Socket mode

In socket mode, the IP Virtual Card uses the standard network interfaces. Those interfaces can be configured using any network manager tool.

DPDK mode

Nvidia/Mellanox NIC

When using those Nics, the user can set the nic interface in the same way as done in Socket mode.

Other brand NIC

When using Intel NICs or other Brand NIC, The ip virtual card create a virtual interface for each port bound to VCS.

Below some examples of how to configure the virtual interfaces on different Linux distributions.

Example

Ubuntu Server:

It is recommanded to use netplan to configure permanently the network interfaces.

Ubuntu Desktop and Almalinux (Using NetworkManager):

```
#Static IP configuration:

#Create NetworkManager connection
sudo ip addr add 10.0.0.1/24 dev veth0
sudo ip link set veth0 u
p
#Make the configuration persistent
sudo nmcli connection modify veth0 connection.autoconnect yes
sudo nmcli connection modify veth0 ipv4.method manual
sudo nmcli connection modify veth0 ipv4.addresses 'x.x.x.x
'
sudo nmcli connection down veth0
sudo nmcli connection up veth0

#Dynamic IP configuration:

#Create NetworkManager connection
sudo ip addr add 10.0.0.1/24 dev veth0
sudo ip link set veth0 u
p
#Make the configuration persistent
sudo nmcli connection modify veth0 connection.autoconnect yes
sudo nmcli connection modify veth0 ipv4.method aut
o
sudo nmcli connection modify veth0 ipv4.addresses '
'
sudo nmcli connection modify veth0 ipv6.method "disabled
"
sudo nmcli connection down veth0
sudo nmcli connection up veth0
```

Performance considerations

Socket

Redist package installs a script that is launched at each startup and that improves the socket performance.
You can find that script here : `/usr/share/deltacast/vcs/network_sysctl.sh`

Explanations :
`sysctl -w net.core.rmem_max=33554432`
This sets the max OS receive buffer size for all types of connections.

`sysctl -w net.core.wmem_max=33554432`
This sets the max OS send buffer size for all types of connections.

`sysctl -w net.core.rmem_default=65536`
This sets the default OS receive buffer size for all types of connections.

`sysctl -w net.core.wmem_default=65536`
This sets the default OS send buffer size for all types of connections.

`sysctl -w net.ipv4.route.flush=1`
This will ensure that immediately subsequent connections use these values.

BIOS

According to our observations, C-states, P-states or any energy-saving parameters must be disabled in the BIOS.

This ensures that the computer is running at its peak performances.

Not following the recommendations can lead to unstable or non-compliant streams.

CPU governor

We strongly recommend `performance` to disable `ondemand` CPU scaling daemon to ensure best performances:
`sudo systemctl disable ondemand`

To avoid reboot, you can manually change the actual scaling governor:
`echo performance | sudo tee /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor`

Conductor configuration

The CPU core associated to a conductor must not be used by any process.
To avoid kernel interruption on this core, it can be isolated at boot via the option `isolcpus` or with the KBAPI configuration (in case of DPDK mode).

If the hyper-threading is activated, the same guideline must be applied to the associated logical core.

Route

In order to improve the performances during a TX stream emitting toward a multicast address, we advise to set a general multicast route associated with the used NIC.

This can be done thanks to the following command :
`sudo route add -net 224.0.0.0/4 dev xxxx`,
where ‘xxxx’ is the name of the NIC.

Without this route, the packet interval time (PIT) will not be stable and this can cause non-compliance to the ST2110-21 packet pacing standard (overflow in the VRX bucket).

Licensing

The IP Virtual Card solution is secured by a license manager called **dlmcli**.

To uniquely identify the platform on which the IP Virtual Card runs, dlmcli takes several parameters into account. One of those parameters is the MAC address of one of the NICs present in the machine. To avoid license issues in case of network configuration change, we recommend to force dlmcli to use the MAC address of a specific NIC that should never be removed or used by IP Virtual Card (motherboard integrated NIC by example). For that, use the following argument while you add the first license:
`--select-custom-mac ############`

If a custom MAC address is not provided with the first license entry, dlmcli will warn you and list all the available NIC MAC address.

To add a license in online mode , use the following command:

`dlmcli activate ####-####-####-####-####-####-####-#### [--select-custom-mac ############]`
In order to add a license in online mode, the system time must be correct.

To add a license in offline mode , use the following command:
`dlmcli activate --offline requestfile.bin ####-####-####-####-####-####-####-#### [--select-custom-mac ############]`
Provide the processed requestfile.bin to DELTACAST. In return, DELTACAST will provide you a response file. Then use the following command:
`dlmcli process responsefile.bin`

To update the licensing information in VCS, without having to restart it, call the `VMIP_RefreshLicensing()` function.
You can also compile and run the `sample_refresh_licensing` .

To remove, transfer, repair, unlock, or perform any other operation on a licence, please contact DELTACAST.

Synchronize multiple NICs

When using multiple NICs, if **traffic shaping narrow linear** is enabled, You **MUST** synchronize all the NICs on the interface synchronized with PTP4L.

To synchronize the NICs, a Python utility is available in the VCS folder:

This script must be run with admin rights.

The script creates all the necessary phc2sys services.

```
/usr/share/deltacast/vcs/config_nic_sync.py
USAGE:
# Create and load all the required services

config_nic_sync.py [ptp_domain] [main_interface] [follower_interface] ... [follower_interface]

config_nic_sync.py clean    # Clean previous configuration

config_nic_sync.py start    # Start all sync services

config_nic_sync.py restart  # Restart all sync services

config_nic_sync.py stop     # Stop all sync services
```

Example of setup:

```
/usr/share/deltacast/vcs/config_nic_sync.py 127 eno1 eno2 eno3 eno4
```

Virtual machine support

Configuration

The virtual machine support is only available with network cards configured in PCI passthrough. Socket and DPDK mode are both available in virtual machine.

PTP

NTP disabling

In order to have a proper PTP synchronization, the NTP service must be disabled. The installation of the IP Virtual Card will automatically disable NTP.

Firewall

In order to make PTP work, you have to be sure that the ptp packets are not blocked by any kind of firewall installed on your machine. If it is the case, you must add a firewall rule to allow those packets or PTP will not work.