Installation and Maintenance Instructions for Intel(R) R-WPA VNF Package

Contents

Contents .............................................................................................................................................. 1
Table of Figures .................................................................................................................................. 2
Overview ........................................................................................................................................... 3
Installation Steps .............................................................................................................................. 4
  1.  Host Setup .................................................................................................................................. 4
      1.1  System BIOS settings ............................................................................................................. 4
      1.1.1 Virtualization ..................................................................................................................... 4
      1.1.2 Hyper-threading .................................................................................................................. 4
      1.1.3 I/O Virtualization (SRIOV) .................................................................................................. 4
      1.2  Host Linux Kernel boot configuration ...................................................................................... 4
      1.2.1 Hugepage memory allocation .............................................................................................. 4
      1.2.2 Host CPU core isolation ....................................................................................................... 4
      1.2.3 Grub enable SRIOV .............................................................................................................. 5
      1.3  Linux package install on Host OS ........................................................................................... 5
      1.4  Load the R-WPA VNF package ............................................................................................... 6
      1.5  Load DPDK on the host ........................................................................................................... 6
      1.6  Build the R-WPA pf_init application ....................................................................................... 6
  2.  Control VNF (VNF-C) Installation ............................................................................................... 7
      2.1  Control VNF (VNF-C) VM Creation ......................................................................................... 7
      2.2  Disable firewall on VNF-C VM ............................................................................................ 7
      2.3  VNF-C VM Linux package install .......................................................................................... 8
  3.  Data Plane VNF (VNF-D) Installation ......................................................................................... 10
      3.1  Data Plane VNF (VNF-D) VM Creation ................................................................................ 10
      3.2  VNF-D VM Kernel boot configuration .................................................................................... 10
      3.2.1 Hugepage memory allocation in VNF-D VM ...................................................................... 10
      3.2.2 VNF-D VM core isolation .................................................................................................. 11
      3.3  Disable firewall on VNF-D VM ............................................................................................. 11
      3.4  VNF-D VM Linux package install ........................................................................................ 11
3.5 Install and build AESNI-MB crypto library ................................................................. 12
3.6 Install and build DPDK on the VNF-D VM .................................................................. 12
3.7 Install and build R-WPA dataplane application In the VNF-D VM .............................. 14
4. Libvirt Network Configuration ......................................................................................... 16
5. Start VNF VM’s ................................................................................................................ 17
  5.1 Adding interfaces to VM’s .............................................................................................. 17
  5.2 Start VNF VM’s ............................................................................................................ 19
  5.3 Running the hostapd application in the VNF-C VM ..................................................... 20
  5.4 Running R-WPA dataplane application in the VNF-D VM .......................................... 21
System Maintenance Guide ................................................................................................ 23
6. R-WPA VNF-D Compilation Options ............................................................................. 23
7. Monitoring Statistics ........................................................................................................ 25
8. Re-starting R-WPA VNF’s post installation .................................................................... 28
  8.1 Re-configuring R-WPA VNF’s ...................................................................................... 28
  8.2 Stopping VNF’s ........................................................................................................... 28
  8.3 Re-starting VNF’s ........................................................................................................ 29
Appendix A – Delay Simulation ......................................................................................... 30

Table of Figures
Figure 1: R-WPA VNF Architecture (Lab set-up) ................................................................. 3
Figure 2: Port Rx/Tx statistics ............................................................................................. 25
Figure 3: Station lookup and crypto statistics ..................................................................... 25
Figure 4: Uplink statistics .................................................................................................... 26
Figure 5: Downlink statistics ............................................................................................... 26
Figure 6: Control Plane Statistics ....................................................................................... 27
Figure 7: Uplink Cycle Statistics ........................................................................................ 27
Figure 8: Delay Simulation ................................................................................................. 30
Overview

The following guide explains, step by step, how to install the R-WPA VNF platform on an Intel Xeon Server with Fedora 24 OS. This guide contains instructions on the configuration and running of the R-WPA Data Plane and Control VNF’s. An overview of the R-WPA VNF configuration included in this guide is shown in Figure 1 below.

Figure 1: R-WPA VNF Architecture (Lab set-up)
Installation Steps

1. Host Setup

1.1 System BIOS settings
Before VNF software installation certain BIOS settings of the server must be enabled. The BIOS settings can be adjusted by entering the BIOS menu at boot time.

1.1.1 Virtualization
Virtualization must be enabled. This can be done by setting the "Intel(R) Virtualization Technology" field under the "Processor Configuration" sub-menu to 'Enabled'.

1.1.2 Hyper-threading
Hyper-threading must be enabled. This can be done by setting the "Intel(R) Hyper-Threading Technology" field under the "Processor Configuration" sub-menu to "Enabled".

1.1.3 I/O Virtualization (SRIOV)
I/O Virtualization must be enabled. This can be done by setting the "Intel(R) VT for Directed I/O" field under the "Integrated IO Configuration" sub-menu to "Enabled".

1.2 Host Linux Kernel boot configuration

1.2.1 Hugepage memory allocation
DPDK requires contiguous hugepage memory, and hugepages are also used by qemu-kvm to back VM’s.

Hugepages are added to the system by editing the Linux kernel grub settings file and adding the relevant kernel parameters.

32 x 1GB hugepages should be allocated to the system by configuring the Linux grub file as shown below.

```
# vi /etc/default/grub

GRUB_CMDLINE_LINUX="... default_hugepagesz=1G hugepagesize=1G hugepages=32 ...
```

1.2.2 Host CPU core isolation
The CPU cores used by both DPDK and the VNF Virtual Machines must be isolated from the Linux kernel task scheduler.

This is done by adding the "isolcpus" setting to the kernel boot parameters as shown below to isolate the relevant cores. The core number values can be determined by examining the virsh xml for both the VNFC and VNFD VM’s. These are located at ‘scripts/vnfc/VNFC-RWPA.xml’ and ‘scripts/vnfd/VNFD-RWPA.xml’ within the RWPA VNF package.
Note also that the isolcpus settings will need to be adapted the specific core layout of the CPU package on the system. Run the DPDK utility below to check logical core layout for the CPU package.

```bash
# $DPDK_DIR/usertools/cpu_layout.py
```

E.g. to isolate cores 1 to 17, 19 to 35, 37 to 53 and 55 to 71, add the following settings to the GRUB_CMDLINE_LINUX line in the Linux grub file.

```bash
# vi /etc/default/grub
.
.
GRUB_CMDLINE_LINUX="... isolcpus=1-17,19-35,37-53,55-71 ..."
.
.
```

1.2.3 Grub enable SRIOV

To enable SRIOV iommu must be enabled in the kernel.

```bash
# vi /etc/default/grub
.
.
GRUB_CMDLINE_LINUX="... iommu=pt intel_iommu=on ..."
.
.
```

Compile the GRUB configuration and reboot the server for the Hugepages, CPU isolation and SRIOV settings to take effect.

```bash
# grub2-mkconfig -o /boot/grub2/grub.cfg
# reboot
```

1.3 Linux package install on Host OS

A number of standard Linux utilities must be installed to support DPDK and QEMU.

These can be installed through the Linux package manager.

```bash
# dnf remove vim
# dnf install vim git gcc libvirt qemu kernel-devel kernel-headers \
  elfutils-libelf-devel autoconf automake libtool numaclt-devel \
  libpcap-devel openssl-devel libtalloc-devel virt-manager python
```
1.4 Load the R-WPA VNF package
The R-WPA VNF installation package should be copied into the /root directory and the contents extracted.

1.5 Load DPDK on the host
DPDK also needs to be installed on the host. Install, checkout the correct version and build DPDK using the following commands. The required DPDK version for the Host is v18.02.

```
# cd /root
# git clone http://dpdk.org/git/dpdk
# cd /root/dpdk
# git checkout v18.02
# make install T=x86_64-native-linuxapp-gcc -j 11
```

1.6 Build the R-WPA pf_init application
In order to use SRIOV Virtual Functions in conjunction with DPDK on the guest, for optimal performance the Physical Functions (PF) should also be bound to DPDK. This requires running a basic DPDK application on the host to initialize PF’s. Source-code is provided for his application and can be built by changing to the ‘pf_init’ directory under the rwpa package and executing ‘make’. Note that DPDK environment variables will need to be set.

```
# cd /root/rwpa_vnf/pf_init
# export RTE_SDK=/root/dpdk
# make
```
2. Control VNF (VNF-C) Installation
The platform is now ready for VNF installation (starting with Control VNF).

In order to create Virtual Machines for the VNF’s a Fedora image is needed. Such images can be downloaded from "getfedora.org". The server image for Fedora 24 should be chosen.

2.1 Control VNF (VNF-C) VM Creation
The VNF-C VM can be created by following the steps outlined in this section.

Firstly, create the VM image and install the Fedora OS in the VM by following install instructions on the pop-up GUI.

NOTE: X11 forwarding must be enabled on the host OS for the installation GUI to be displayed.

Ensure that libvirtd is active by issuing the start command and then checking the status. Then run the following command to start virt-manager.

```
# systemctl start libvirtd
# systemctl status libvirtd
# virt-manager
```

A pop-up GUI will be presented. Under the ‘File’ tab select the option titled ‘New Virtual Machine’. This will create another window to allow the creation of a VM.

Choose the option ‘Local install media (ISO image or CDROM)’ and select the ‘Forward’ button.

In the next window click ‘Browse…’ to locate the Fedora image on your server. Once located select ‘Choose Volume’ to select this image. Check the box ‘Automatically detect operating system based on install media’ and then click ‘Forward’.

The next step is to allocate memory and CPUs to the VM. Allocate 4192MiB of memory and 2 CPUs. Then select ‘Forward’.

Check the box ‘Enable storage for this virtual machine’ and ‘Create a disk image for the virtual machine’. Allocate 20GiB of storage then select ‘Forward’.

On the final step enter ‘VNFC-RWPA’ into the ‘Name’ box, then click ‘Finish’.

A window will now appear to install Fedora. Follow the instructions from this window to install the OS.

Upon completion reboot the VM by selecting it and clicking ‘Power on the virtual machine’ using the virt-manager GUI. Enter credentials to log into the VM.

2.2 Disable firewall on VNF-C VM
Once logged into the VM, VNF-C requires some additional steps to be configured on the VM.

Disable firewall with the following commands:
2.3 VNF-C VM Linux package install

Before installing the required packages with the Linux package manager, note that if the system is behind a proxy this information must be configured in the OS.

This can be done by editing the file "/etc/dnf/dnf.conf" and setting the "proxy" entry to any proxies which are specified in the `bashrc` file.

```
# vi /etc/dnf/dnf.conf

proxy=....
```

Install packages using the Linux package manager and replace the default vim package which has minimal features.

```
# dnf remove vim
# dnf install vim gcc openssl-devel
```

Then load both the VNF-C hostapd package (provided separately by Intel) and the VNF-D R-WPA data plane package into the VNF-C VM, and extract files to the root directory.

The hostapd application can be compiled by changing to the hostapd subdirectory and executing make. The R-WPA package is required on the VNF-C as it contains some utility scripts for configuring Linux kernel interfaces.

Build the hostapd application and shut down the VM:

```
# cd /root/hostapd_2.6/hostapd
# make
# shutdown now
```

The disk image created now contains the necessary software. However virsh provides features for fine tuning core-mapping, memory and IO-devices. The rwpa_vnf package contains a virsh xml file specifying such details for the VNF-C VM. Import this file to virsh using the following commands. Note that these commands will redefine the VM which was created. The VM will use the VM disk image created and the xml definition from the rwpa_vnf package. The following commands should be executed on the host.
# virsh undefine VNFC-RWPA
# virsh define rwpa_vnf/scripts/vnfc/VNFC-RWPA.xml
3. Data Plane VNF (VNF-D) Installation

3.1 Data Plane VNF (VNF-D) VM Creation

The VNF-D VM can be created by following the steps outlined in this section.

Firstly, create the VM image and install the Fedora OS in the VM by following install instructions on the pop-up GUI.

NOTE: X11 forwarding must be enabled on the host OS for the installation GUI to be displayed.

Ensure that libvirtd is active by issuing the start command and then checking the status. Then run the following command to start virt-manager.

```bash
# systemctl start libvirtd
# systemctl status libvirtd
# virt-manager
```

A pop-up GUI will be presented. Under the ‘File’ tab select the option titled ‘New Virtual Machine’. This will create another window to allow the creation of a VM.

Choose the option ‘Local install media (ISO image or CDROM)’ and select the ‘Forward’ button.

In the next window click ‘Browse…’ to locate the Fedora image on your server. Once located select ‘Choose Volume’ to select this image. Check the box ‘Automatically detect operating system based on install media’ and then click ‘Forward’.

The next step is to allocate memory and CPUs to the VM. Allocate 10240MiB of memory and 6 CPUs. Then select ‘Forward’.

Check the box ‘Enable storage for this virtual machine’ and ‘Create a disk image for the virtual machine’. Allocate 20GiB of storage then select ‘Forward’.

On the final step enter ‘VNFD-RWPA’ into the ‘Name’ box, then click ‘Finish’.

A window will now appear to install Fedora. Follow the instructions from this window to install the OS.

Upon completion reboot the VM by selecting it and clicking ‘Power on the virtual machine’ using the virt-manager GUI. Enter credentials to log into the VM.

3.2 VNF-D VM Kernel boot configuration

Once logged into the VM, the VNF-D VM kernel boot parameters should be edited for the settings described below.

3.2.1 Hugepage memory allocation in VNF-D VM

Configure parameters to set up hugepages for the VNF-D application, by adding the following settings to the GRUB_CMDLINE_LINUX line, for example to configure 6 x 1GB huge-pages.
3.2.2 VNF-D VM core isolation

It is necessary to isolate the virtual CPU’s on which the data plane application will be running. This is one by adding the "isolcpus" setting to the kernel boot parameters as shown below.

Note that this will only be relevant if the physical cores on which the virtual CPU’s run are also isolated on the Host OS kernel through its grub configuration file as described in the Host CPU core isolation section.

e.g. to isolate cores 1 to 5 in the VNF-D VM add the following settings to the Linux grub configuration file.

```
# vi /etc/default/grub
.
.
GRUB_CMDLINE_LINUX="... isolcpus=1-5 ..."
.
.
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

Compile the GRUB configuration and the settings will take effect (on next startup)

3.3 Disable firewall on VNF-D VM

VNF-D VM requires the firewall to be disabled.

Disable firewall with the following commands:

```
# systemctl stop firewalld
# dnf remove firewalld
```

3.4 VNF-D VM Linux package install

Before installing the required packages with the Linux package manager, note that if the system is behind a proxy this information must be configured in the OS.

This can be done by editing the file "/etc/dnf/dnf.conf" and setting the "proxy" entry to any proxies which are specified in the `bashrc` file.
Install packages using the Linux package manager and replace the default \texttt{vim} package because it has minimal features.

```bash
# vi /etc/dnf/dnf.conf

# dnf remove vim
# dnf install vim gcc openssl-devel kernel-devel kernel-headers numactl-devel \ elfutils-libelf-devel python nasm
```

3.5 Install and build AESNI-MB crypto library

The Intel AESNI-MB cryptographic library is used to perform AES CCM encryption/decryption in the VNFD application. It is highly optimized and performs significantly better than standard implementations such as OpenSSL. The library may be downloaded at ‘https://github.com/intel/intel-ipsec-mb’ and built using the following commands. After building, an environmental variable should be set to point to the location of this installation. This is used by DPDK to locate the AESNI-MB cryptographic library.

```bash
# cd ~
# git clone https://github.com/intel/intel-ipsec-mb.git
# cd intel-ipsec-mb
# git checkout v0.48
# make all
# export AESNI_MULTI_BUFFER_LIB_PATH=/root/intel-ipsec-mb
```

3.6 Install and build DPDK on the VNF-D VM

In addition to loading the R-WPA VNF package on the VM in the root directory it is also necessary to install and build DPDK.

As on the host, go to the root directory in the VNF-D VM, install the DPDK source code through git and then build it.
Note that while the same version of DPDK is used in the VNF-D VM, some modifications must be made to DPDK configuration options which are specific to the R-WPA dataplane application which runs in the VNF-D VM.

The correct DPDK version to use for the VNF-D VM is v18.02.

```
# cd /root
# git clone http://dpdk.org/git/dpdk
# git checkout v18.02
# cd dpdk
```

Make modifications to the following lines in “config/common_base”

```
# vi config/common_base
.
.
CONFIG_RTE_LIBRTE_PMD_QAT=Y
.
CONFIG_RTE_LIBRTE_PMD_AESNI_MBR=Y
.
CONFIG_RTE_LIBRTE_PMD_OPENSSL=y
.
CONFIG_RTE_LIBRTE_IP_FRAG_MAX_FRAG=2
.
.
```

DPDK can be built using the following command.

Note use of “-j 11” to allow parallel make jobs which speeds up compilation time.

```
# make install T=x86_64-native-linuxapp-gcc -j 11
# export RTE_SDK=$PWD
```

It is useful to include environmental variables in the ‘/root/.bashrc’ file as they will then persist across reboots. Do this by editing the file.
3.7 Install and build R-WPA dataplane application in the VNF-D VM

The R-WPA VNF package should then be loaded into the VNF-D VM and files extracted into the root directory.

Change to the R-WPA data plane source-code folder and build the data path application.

```bash
# vi ~/.bashrc
.
.
export RTE_SDK=/root/dpdk
export AESNI_MULTI_BUFFER_LIB_PATH=/root/intel-ipsec-mb
.
.
# cd /root/rwpa_vnf/rwpa_dp/
# make
```

Note two sub-directories under rwpa_dp, the config directory which contains default application configuration files and the certs directory which contains security files required for the TLS connection between the VNF-D and VNF-C VM's. The default.cfg file should be configured with required IP/MAC addresses for VNF-D, VNF-C and WAG; the other configuration files may be used unmodified. As regards the cert files, these should match equivalent files in the VNF-C VM and may be re-generated if required (on both sides).

After building and configuring the R-WPA data plane application the VNF-D VM must be shut down to allow the VM's to be rebooted with the network interfaces configured for correct operation of the VNF's.

```bash
# shutdown now
```

The disk image created now contains the necessary software. However virsh provides features for fine tuning core-mapping, memory and IO-devices. The rwpa_vnf package contains a virsh xml file specifying such details for the VNF-D VM. Import this file to virsh using the following commands. Note that these commands will redefine the VM which was created. The VM will use the VM disk image created and the xml definition from the rwpa_vnf package. The following commands should be executed on the host.
Note that a different xml file is used to define the VNF-D VM when Intel Quick Assist Technology (QAT) is used to accelerate cryptographic operations.

An Intel Quick Assist Technology PCIe card should be installed on the R-WPA VNF server in this case.

And the following commands are executed.

```
# virsh undefine VNFD-RWPA
# virsh define rwpa_vnf/scripts/vnfd/VNFD-RWPA.xml
```

```
# virsh undefine VNFD-RWPA
# virsh define rwpa_vnf/scripts/vnfd/VNFD-RWPA-QAT.xml
```
4. Libvirt Network Configuration

A libvirt network bridge is used for communication between VNF-C and VNF-D VM’s. This network must be defined using virsh. The definition is included in the ‘rwpa_vnet.xml’ file located under the ‘scripts/host’ directory of the rwpa package. Define and start the network using the following commands.

```
# cd rwpa_vnf/scripts/host
# virsh net-define rwpa_vnet.xml
# virsh net-autostart rwpa_vnet
# virsh net-start rwpa_vnet
```
5. Start VNF VM's

5.1 Adding interfaces to VM's.

The final configuration to make using virsh is to the I/O devices for each VM which will differ depending on the PCI enumeration of the NIC from platform to platform. To make this easier a ‘vf_test.sh’ script is provided. Before continuing, the NIC interfaces that will be used must be configured in the ‘ports.sh’ script under the ‘scripts/host’ directory of the rwpa_vnf package.

Edit the ‘ports.sh’ script specifying the following variables. Running the commands ‘ifconfig’ and ‘lspci | grep Eth’ will provide the information required to set these variables.

"phy0" variable - specifies the pci address of the interface connecting the VNF-D to the WAG.

"phy1" variable - specifies the pci address of the interface connecting VNF-C to the Radius Server/CMTS/AP which also connects VNF-D to the CM<TĂś>TS/AP.

“iface0” variable – specifies the linux interface name of the interface connecting the VNF-D to the WAG.

“iface1” variable – specifies the linux interface name of the interface connecting VNF-C to the Radius Server/CMTS/AP which also connects VNF-D to the CM<TĂś>TS/AP.

If using Intel Quick Assist Technology (QAT) the argument ‘-q’ should be passed to the ‘vf_test.sh’ script. This will output an extra device pci address to be added to the VNF-D. This is the QAT virtual function. The pci address of the QAT card should be assigned to the "qat0" variable in the ‘scripts/host/qat.sh’ file. The pci address of the QAT card can be found by running ‘lspci | grep QAT’.

After specifying the interfaces, run the ‘vf_test.sh’ script under the ‘scripts/host’ directory. The script will print out some log messages and then print out the list of pci devices that should be used by the VM’s. Note that the order in which devices are added is important and devices should be added in the order they are printed out. See below for sample output from the ‘vf_test.sh’ script.

```
# cd /root/rwpa_vnf/scripts/host
# ./vf_test.sh
.
.
.
PCI Devices to use on VNFC VM are:
07:02.0

PCI Devices to use on VNFD VM are:
07:02.1
05:00.0
```
Adding PCI devices to the VM can be done by editing the virsh xml. The virsh xml contains a section for adding host devices under the “<hostdev>” element. Add devices to the VNF-C by executing the command below. This will open up the virsh xml in a text editor. Search for the “hostdev” elements. There will be three. Edit the fields shown in green in the box below to reflect the correct pci addresses output by the ‘vf_test.sh’ script. Once edited the xml can be saved and closed by typing the following characters “:wq” and pressing return. Ensure that the pci addresses are entered in the virsh xml in the same order that they are output by the ‘vf_test.sh’ script.

```xml
<hostdev mode='subsystem' type='pci' managed='yes'>
    <source>
        <address domain='0x0000' bus='0x00' slot='0x0a' function='0x0'/>
    </source>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x0a' function='0x0'/>
</hostdev>
```

The same steps must be taken to add devices to the VNF-D VM. The xml for the VNF-D VM can be edited using the command below. Similar to the VNF-C the “<hostdev>” elements must be edited with the correct pci addresses outputted by the ‘vf_test.sh’ script. Once editing is complete save and close the xml by typing “:wq” and pressing return.

If using QAT use the \texttt{VNFD-RWPA-QAT.xml} file instead of \texttt{VNFD-RWPA.xml} for the following commands.

```bash
# vim scripts/vnfc/VNFC-RWPA.xml
```
Once updated redefine the new VM definitions which will include host PCI devices using the following commands.

```bash
# virsh undefine VNFD-RWPA
# virsh define rwpa_vnf/scripts/vnfd/VNFD-RWPA.xml
# virsh undefine VNFC-RWPA
# virsh define rwpa_vnf/scripts/vnfc/VNFC-RWPA.xml
```

### 5.2 Start VNF VM’s

The pf_init application must be started before starting VM’s as it is required to initialize the physical interface connected to the CMTS/AP. Do this by executing the following commands and replacing ‘$phy1’ with the correct pci address as described in the previous section.

```bash
# cd /root/dpdk
# modprobe uio
# insmod x86_64-native-linuxapp-gcc/kmod/igb_uio.ko
# ./usertools/dpdk-devbind.py -b igb_uio $phy1
# cd /root/rwpa_vnf/pf_init/build
# ./pf_init -l 2 -w $phy1 -socket-mem 256,256 -- -p 0x1 -m \ 00:00:00:00:00:05 &
```

If using an Intel Quick Assist Technology card, the following additional commands are required to initialize the physical device on the host with the QAT Linux kernel modules

```bash
# cd /root/dpdk
# modprobe qat_dh895xcc
# modprobe intel_qat
# ./usertools/dpdk-devbind.py -b dh895xcc $qat0
```

Virtual Machines for the VNF-C and VNF-D can then be started using virsh commands outlined below. The IP address of the admin interface of each VM can also be queried using virsh.
5.3 Running the hostapd application in the VNF-C VM

The hostapd application provides the core functionality of the R-WPA Control VNF (VNF-C). The following instructions describe how to launch the hostapd application.

Firstly log in as root user with the password as configured when setting up the VNF-C VM.

Then the network interfaces of the VM must be assigned IP addresses to work correctly with the default configurations of the AP’s and VNF-D.

This can be done by copying ifcfg files for each interface from the "scripts/vnfc" directory as shown below, and the networking service must be restarted for these changes to take effect. Ensure that the IP address assigned to the interface ‘ens10’ is suitable for the Network in which the R-WPA VNF is being installed. Interface ‘ens10’ will be connected to the CMTS/Home Gateway/Radius Server. ‘ens8’ is used for communication with VNF-D and the default IP address should be sufficient.

```
# cd /root/rwpa_vnf/scripts/vnfc
# cp ifcfg-ens8 /etc/sysconfig/network-scripts/
# cp ifcfg-ens10 /etc/sysconfig/network-scripts/
# systemctl restart network
```

Running "ifconfig" will confirm the network interfaces have been configured correctly.

The hostapd application runs in the VNF-C VM, and must be configured through the "wpapt.conf" file that is located in the "hostapd" subdirectory. Detailed instructions for configuring hostapd using this configuration file can be found in the hostapd package. The "wpapt.conf" contains a sample configuration. Importantly, all Access Points must be configured in this file along with the IP address of the radius server.

Routes to the Radius Server and Home Gateways may also have to be configured on the VNF-C VM. This can be done using the ‘ip route’ utility.

After configuration is complete the hostapd application can be launched by running the setup script in the "scripts/vnfc" directory.
Note that passing the "-ddt" option to hostapd will result in more debug output to be printed to the console which may or may not be of interest.

Log out of the VNF-C VM for the next step.

5.4 Running R-WPA dataplane application in the VNF-D VM

The R-WPA dataplane application provides the core functionality of the R-WPA Dataplane VNF (VNF-D). The following instructions describe how to launch the R-WPA dataplane application.

Firstly log in with root username and password as configured when setting up the VNF-D VM.

Then the Linux network interfaces of the VM must be assigned IP addresses to work correctly with the default configurations of the AP’s and VNF-C VM.

This can be done by copying ifcfg files for each interface from the "scripts/vnfd" directory as shown below, and the networking service must be restarted for these changes to take effect. Interface ‘ens8’ is used for communication with VNF-C so the default IP address should be sufficient.

```bash
# cd /root/rwpa_vnf/scripts/vnfc
# ./setup.sh
# cd /root/rwpa_vnf/scripts/vnfd
# cp ifcfg-ens8 /etc/sysconfig/network-scripts/.
# systemctl restart network
```

Running "ifconfig" will confirm the network interfaces have been configured correctly.

Running "ifconfig" will confirm the network interfaces have been configured correctly.

The default configuration of the VNF-D may need to be updated to reflect IP and MAC addresses, in particular that of the WAG. This can be done by editing the default.cfg file located under the ‘rwpa_vnf/rwpa_dp/config’ directory and updating the ‘ADDRESSES’ section. The entries listed below are the ones most likely to require updating.

If using QAT the configuration file is named ‘default-qat.cfg’.

```bash
# vim /root/rwpa_vnf/rwpa_dp/config/default.cfg
.
.
[ADDRESSES]
vnfd_ip_to_ap = "VNFD IP address for connection to CMTS/CPE"
vnfd_ip_to_wag = "VNFD IP address for connection to WAG"
wag_tun_ip = "IP address of WAG"
wag_tun_mac = "MAC address of first hop on route to WAG"
.
```
Once configuration is completed the R-WPA data plane application can be launched by running the setup script in the "scripts/vnfd" directory.

```
# cd /root/rwpa_vnf/scripts/vnfd
# ./setup.sh
```

If using QAT card the ‘-q’ argument should be passed to this script.

```
# cd /root/rwpa_vnf/scripts/vnfd
# ./setup.sh -q
```

The R-WPA VNF’s are now running and will accept AP and client station connections. The final step is to start the Access Points.
6. R-WPA VNF-D Compilation Options

The R-WPA application provides a number of useful build options to enable different levels of statistic monitoring.

Build options which may be of interest are as follows:

(Note that all are disabled by default)

RWPA_DEBUG_BUILD=1

This option builds the R-WPA data plane application with the lowest level of gcc compiler optimizations.

RWPA_EXTRA_DEBUG=1

Provides additional debug statistics to show location of dropped packets. This option will effect performance as it includes printing console output within the pipeline.

RWPA_STATS_CAPTURE=1

This option enables statistics capture. By default statistics will be printed to screen every 3 seconds. Statistics included by this option are port statistics, station lookup statistics, cryptographic statistics, uplink statistics, downlink statistics and control statistics. Statistics capture is designed in a way that minimizes the performance impact. Use of this option will not be as invasive as the RWPA_EXTRA_DEBUG compile option.

RWPA_STATS_CAPTURE_PORTS_OFF=1

This option can disable port statistic capture.

RWPA_STATS_CAPTURE_STA_LOOKUP_OFF=1

This option disables station lookup statistics capture.

RWPA_STATS_CAPTURE_CRYPTO_OFF=1

This option disables crypto statistics capture.

RWPA_STATS_CAPTURE_UPLINK_OFF=1

This option disables statistics capture on the uplink thread.

RWPA_STATS_CAPTURE_DOWNLINK_OFF=1

This option disables statistics capture on the downlink thread.

RWPA_STATS_CAPTURE_CONTROL_OFF=1

This option disables statistics capture of control plane messages.

RWPA_CYCLE_CAPTURE=1
This option is used to enable cycle count statistics capture. Compiling with this option will capture cycle counts for different function calls and make this information available to the user. This option is particularly useful to show the cycle count breakdown of the R-WPA pipeline. This option can be set to RWPA_CYCLE_CAPTURE=1 or RWPA_CYCLE_CAPTURE=2. Setting to ‘1’ shows the number of cycles used overall on the Uplink and Downlink threads while setting to ‘2’ gives a detailed breakdown of the R-WPA pipeline showing the cost of stages on both Uplink and Downlink. Note that to use this option RWPA_STATS_CAPTURE=1 must also be set when compiling.

RWPA_AP_TUNNELLING_GRE=1

By default the R-WPA data plane application uses a UDP tunnel to the Access Point. By enabling this option a GRE tunnel is used between R-WPA data plane and Access Points. Note that the tunneling used needs to match that used for the software version on the Access Point.

The following is an example of how to compile with these options.

```
# make RWPA_STATS_CAPTURE=1 RWPA_CYCLE_CAPTURE=1
```
7. Monitoring Statistics

When the R-WPA data plane application is compiled with the RWPA_STATS_CAPTURE=1 or RWPA_CYCLE_CAPTURE=1 build options statistics will be printed to the console output. Capture of statistics is designed to have minimal impact on the VNF-D performance. Some screen clippings are included in this section to show the output that can be provided by using these options at compilation time.

![Figure 2: Port Rx/Tx statistics](image)

Figure 2 above shows the basic port Rx/Tx statistics. These ports are the VNFD ports. All uplink and downlink traffic can be monitored by these port statistics.

![Figure 3: Station lookup and crypto statistics](image)

Figure 3 shows Station lookup statistics and information on the encrypt/decrypt operations.

Figure 4, 5 and 6 that follow show Uplink, Downlink and Control Plane statistics respectively.
### UPLINK

<table>
<thead>
<tr>
<th>PACKET DROPS</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet Decap Errors</td>
<td>0</td>
</tr>
<tr>
<td>Reassembly Errors</td>
<td>0</td>
</tr>
<tr>
<td>Station Not Found</td>
<td>0</td>
</tr>
<tr>
<td>No Station Key</td>
<td>0</td>
</tr>
<tr>
<td>Replay Detected</td>
<td>0</td>
</tr>
<tr>
<td>Decryption Errors</td>
<td>0</td>
</tr>
<tr>
<td>Eth Convert Errors</td>
<td>0</td>
</tr>
<tr>
<td>Data Packet Encap Errors</td>
<td>0</td>
</tr>
<tr>
<td>Ctrl Packet Encap Errors</td>
<td>0</td>
</tr>
<tr>
<td>Unexpected Packet Type</td>
<td>0</td>
</tr>
</tbody>
</table>

*Figure 4: Uplink statistics*

### DOWNLINK

<table>
<thead>
<tr>
<th>PACKET DROPS</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet Decap Errors</td>
<td>0</td>
</tr>
<tr>
<td>Station Not Found</td>
<td>0</td>
</tr>
<tr>
<td>No Station Key</td>
<td>0</td>
</tr>
<tr>
<td>Wifi Convert Errors</td>
<td>0</td>
</tr>
<tr>
<td>Encryption Errors</td>
<td>0</td>
</tr>
<tr>
<td>Fragmentation Errors</td>
<td>0</td>
</tr>
<tr>
<td>Packet Encap Errors</td>
<td>0</td>
</tr>
<tr>
<td>Broad/Multicast Packet</td>
<td>0</td>
</tr>
<tr>
<td>Unexpected Packet Type</td>
<td>0</td>
</tr>
</tbody>
</table>

*Figure 5: Downlink statistics*
Figure 6: Control Plane Statistics

Figure 7 below shows output when RWPA_CYCLE_CAPTURE=2 is set at compilation time. Seen in the diagram is a breakdown of the Uplink pipeline and the associated cycle costs for each stage.

<table>
<thead>
<tr>
<th>Cycle States</th>
<th>Total Calls</th>
<th>Total Cycles</th>
<th>Cycles per Call</th>
<th>Cycles per Mb/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPLINK</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--UL_PMD_RX (exc empty reads)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--incl empty reads</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--UL_INITIAL_PKT_CLASSIFY</td>
<td>--461082561</td>
<td>--858961170</td>
<td>--78</td>
<td>--0</td>
</tr>
<tr>
<td>--UL_UP_FRAME_DECAP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--UL_VAP_HDR_PARSE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--UL_VAP_PKTLOAD_REASSEMBLE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--UL_VAP_HDR_DECAP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--UL_VAP_TLV_DECAP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--UL_IEEE80211_PKT_PARSE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--UL_STA_LOCK</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--UL_STA_decrypt DATA_SET</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--UL-instagram packet</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--UL_STA_decrypt DATA_UPDATE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--UL_CRYPTO</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--UL_CRYPTO ENQUEUE</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
</tr>
<tr>
<td>--UL_CRYPTO dequeue</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
</tr>
<tr>
<td>--UL_STA_UPDATE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--UL_IEEE80211_PKT_CLASSIFY</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--UL_POST_IEEE80211_PKT_CLASSIFY</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--UL_IEEE80211_DATA_PKT_PROCESS</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
</tr>
<tr>
<td>--UL_IEEE80211 TO OTHER CONV</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
</tr>
<tr>
<td>--UL_IEEE80211 ENCAP</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
</tr>
<tr>
<td>--UL_PM_TX</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
</tr>
<tr>
<td>--UL_IEEE80211 EAPOL_PKT_PROCESS</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
</tr>
<tr>
<td>--UL_COMP_DECAP</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
</tr>
<tr>
<td>--UL_VARP_PKT Encap</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
</tr>
<tr>
<td>--UL_VARP_PKT ENCAP</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
</tr>
<tr>
<td>--UL_TLS_TX</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
<td>--0</td>
</tr>
</tbody>
</table>

Figure 7: Uplink Cycle Statistics
8. Re-starting R-WPA VNF’s post installation
Post installation, the R-WPA VNF’s may need to be re-started, due to server re-boot or for re-configuration when Access Points are added/removed.

This can be done by following the simple sequence of instructions described in this section.

8.1 Re-configuring R-WPA VNF’s
Note that not all these configuration steps are not required across restarts if the physical GRX AP’s remain the same. If this is the case ignore the rest of this section and continue from section 8.2 Stopping VNF’s.

If the Access Points serviced by the R-WPA VNF are changed then a number of re-configuration steps are required on the VNF server host before re-starting VNF’s.

8.1.1 Re-configure VNF-C
The AP’s serviced by the R-WPA VNF must be included in the “wpapt.conf” configuration file located in the VNF-C VM under the “hostapd_2.6/hostapd” directory. Details are included within the hostapd package and an example configuration is supplied. Note that this file will not have to change after initial installation unless the Radius Server changes or the GRX AP’s are changed.

8.1.2 Re-configure VNF-D
The configuration of the VNF-D may need to be updated to reflect changes, in particular that of the WAG. This can be done by editing the default.cfg file located under the ‘rwpa_vnf/rwpa_dp/config’ directory and updating the ‘ADDRESSES’ section. The entries listed below are the ones most likely to require updating. This file can be updated within the host rwpa_vnf package as the automated start script will copy this file to the VNF-D guest.

If using QAT make updates to the ‘default-qat.cfg’ file instead.

```
# vim /root/rwpa_vnf/rwpa_dp/config/default.cfg
.
.
[ADDRESSES]
vnfd_ip_to_ap = "VNFD IP address for connection to CMTS/CPE"
vnfd_ip_to_wag = "VNFD IP address for connection to WAG"
wag_tun_ip = "IP address of WAG"
wag_tun_mac = "MAC address of first hop on route to WAG"
.
```

8.2 Stopping VNF’s
Once re-configured using instructions in the previous section, the VNF’s can be stopped using the script “rwpa_stop.sh” located under the “scripts/host” directory.

This script shuts down VNF-C, VNF-D VM’s. Once stopped the VNF’s can be restarted as described in the following section.
8.3 Re-starting VNF’s

The VNF’s can be re-started by running a single script called “rwpa_start.sh” which is located under the “scripts/host” directory. This single script re-starts VNF’s.

Note that the ‘ports.sh’ file should be updated correctly for this script to work. And the ‘qat.sh’ file should also be updated correctly if using QAT.

If using QAT pass the ‘-q’ argument to this script.

```bash
# cd /root/rwpa_vnf/scripts/host
# ./rwpa_start.sh
# cd /root/rwpa_vnf/scripts/host
# ./rwpa_start.sh -q
```
Appendix A – Delay Simulation

Network delays between Customer Premise Equipment and the R-WPA VNFs (VNF-C and VNF-D) can be simulated using scripts in the ‘scripts/host/delay_sim’ directory of the rwpa_vnf package. These scripts simulate network delays from VNF-C to CPE, VNF-D to CPE and VNF-C to VNF-D as shown in Figure 8 below.

![Delay Simulation Diagram](image)

*Figure 8: Delay Simulation*
To start the R-WPA VNFs using delays the delays first need to be specified in the ‘update_delays.sh’ script found in the ‘scripts/host/delay_sim’ directory.

“delay_vnfc_cpe” variable – specifies delay in microseconds between VNF-C and CPE

“delay_vnfd_cpe” variable – specifies delay in microseconds between VNF-D and CPE

“ip_vnfd” variable – specifies the IP address of the VNFD interface connected to CPE

“ip_radius” variable – specifies the IP address of the Radius Server

“ip_home_gw” variable – specifies the IP address of the CPE (Home Gateway or AP)

The delay from VNF-C to VNF-D will be determined as ‘delay_vnfc_cpe− delay_vnfd_cpe’ thus it is required that delay_vnfc_cpe is larger than delay_vnfd_cpe.

Also the ‘ports_delay.sh’ should be configured similar to the ‘ports.sh’ file described in section 4.1. The only additional variable is described below.

“phy2” variable – specifies the pci address of the spare NIC interface on the system which will be used to route traffic through the delay simulator VM.

“iface2” variable – specifies the linux interface name of the spare NIC interface on the system

“phy3” variable – specifies the pci address of the spare NIC interface on the system which will be used to route traffic through the delay simulator VM.

“iface3” variable – specifies the linux interface name of the spare NIC interface on the system

Once the ‘update_delays.sh’ and ‘ports_delay.sh’ files have been configured correctly the R-WPA VNFs can be started by executing the commands below. By running the ‘rwpa_start_delay.sh’ script the VNF-C, VNF-D and Delay VNF VM’s are all started. Once they are booted the VNF-C, VNF-D and Delay applications are started within each VM. The delay values present in the ‘update_delays.sh’ script when the ‘rwpa_start_delay.sh’ script is executed will be used by the delay simulation application.

```
# cd /root/rwpa_vnf/scripts/host/delay_sim
# ./rwpa_start_delay.sh
```

For delays to be updated the VNFs and AP’s do not need to be fully rebooted. Simply edit the ‘update_delays.sh’ script with the new delay values and then execute the script. This allows for updating of the script without restarting the whole setup.

```
# ./update_delays.sh
```

When delay testing is complete shutdown the VNF’s using the provided script.

```
# cd /root/rwpa_vnf/scripts/host
# ./rwpa_stop_delay.sh
```