Virtual Linux Interface (VLI)

bintec-Dm 803-I

Legal Notice

Warranty

This publication is subject to change.

bintec offers no warranty whatsoever for information contained in this manual.

bintec is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

Chapter 1 Introduction

1.1 Architecture

Some of our devices incorporate a dual core microprocessor where each core can be used independently. One executes OS while the other can execute an applications server based on the Linux operating system over which general purpose applications run as they do in a PC. To achieve this, these devices can be equipped with internal storage via USB or SATA.

The device behaves as a conventional router that runs OS, but it is capable of rerouting traffic destined to the applications server over a virtual communication link established via shared memory. The IP that the applications server is going to use is configured in the OS. The VLI module (Virtual Linux Interface) receives the traffic destined to this IP and sends it to the other core. In the applications server, the VTI driver is responsible for OS communications. As a result, a vti0 network device is generated. In the case of the applications server, the vti0 device is similar to an Ethernet device but communication are carried out at the IP layer and not the link layer. The OSautomatically configures the applications server network by assigning it an IP and configuring a default route so all the traffic generated in the Linux is sent through the vti0 device. The OS receives traffic generated by the applications server and reroutes it as locally-generated traffic.

The following figure shows the architecture scheme:

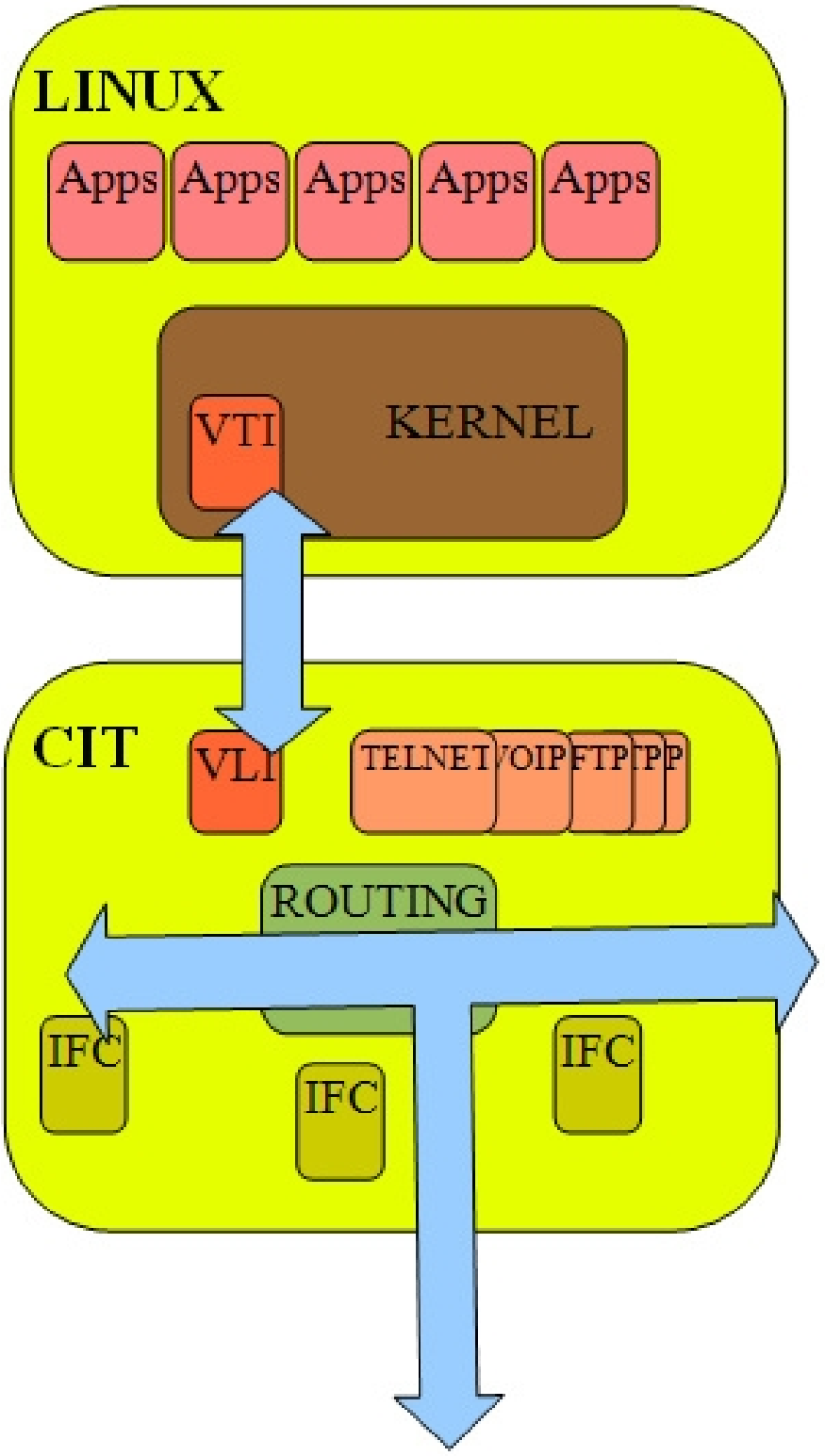

Fig. 1: Architecture scheme

1.2 IP Communications in the Applications Server

You need to configure an IP and allocate it to the applications server for it to work. This IP is configured in the OS and will be the one the applications running in the server use as source in the IP packets generated. Depending on how this IP is configured, there are two possible operating modes: 1) the applications server shares the IP with the OS; 2) the applications server has an exclusive IP. Additionally, you can configure additional commands so that the server receives traffic destined to multicast groups or even traffic that goes through the OS and isn't destined for the latter. The different possibilities are further detailed below.

1.2.1 Applications Server shares IP with the OS

The applications server IP can be the one configured in OS, in an interface, or as an internal or management IP. In this case, only TCP or UDP packets destined to this IP that aren't lost in the OS will be sent to the server. I.e. the OS must not be using this port or have an established TCP/UDP connection. In TCP connections initiated by the applications server and by the OS, source ports have different ranges so they don't overlap.

This operating mode has the advantage of simplifying the IP address planning as you don't have to provide an additional IP for the applications server. However, since the server only receives TCP and UDP traffic through ports that are not being used by the OS, you really have to plan which ports to use for each service.

In this configuration, when a PING is sent to the shared IP, the OS responds, telnet connects to the OS through port 23, and (if there is web service in port 8080 of the applications server), a web connection to this port will be created.

1.2.2 The Applications Server has an exclusive IP

If you configure an IP address for the applications server that is not present in a OS interface and is not the internal IP or the management IP, then all packets destined to this IP are sent to the server. This way, the applications server responds to a PING destined to its IP. In the OS's routing table, a host route to this IP appears so it can be processed as local traffic.

1.2.3 The Applications Server receives Multicast Traffic

You can configure multicast group addresses for groups the applications server wants to join. To do this, you need to enable the IGMP Proxy feature in the OS. The server receives the IP packets destined to the configured multicast groups. You can also send packets addressed to multicast IPs.

1.2.4 The Applications Server monitors OS traffic

Even if the traffic is, in principle, not destined to the server, you can configure an access list to select part of the traffic routed by the OS so that a copy of this traffic is sent to the applications server. This way, applications (such as an intrusion detection system, IDS) can monitor the traffic going through the router.

1.2.5 The Applications Server intercepts traffic from the OS

Similarly to the situation described above, part of the traffic going through the OS can be classified so it is sent to the applications server. However, in this case, instead of sending a copy, the original packet is sent interrupting the processing in the OS. This feature allows you to execute applications that alter the normal OS routing (i.e. a transparent proxy-cache web).

1.2.6 Configuring the Applications Server's DNS servers

At the OS, you can configure the DNS server(s) used by the applications server. You can also configure the OS itself as an applications server's DNS acting as a DNS cache.

1.3 Managing the Applications Server

The applications server is a Linux distribution we designed based on Debian, installed in an .iso file.

From the monitoring and management menu, you can carry out various operations related to the applications server.

Through the **application list** command, you can see all the storage devices that have been detected and their status: installed server, installed, unknown content.

The **application install** command allows you to install the applications server starting from an installer file (.iso). This installer can be obtained through the network or from a storage device connected to the external USB connector.

Through the **application boot** command, you can configure whether the installed applications server should boot when the device is switched on or not.

Some operations, such as disk checking or installation, can only be executed if the applications server is not booted.

You can use the **application check** command to check the integrity of, and potentially repair, an internal storage device.

Chapter 2 Configuration

2.1 Configuring VLI

To access the VLI configuration menu, go to the router's configuration console. To access the menu, execute the following sequence of commands:

```
*config

configuration environment
Config>feature vli

-- VLI configuration --
VLI config>
```

This can also be accessed from the device's dynamic configuration console:

```
*running-config

Config$feature vli

-- VLI configuration --
VLI config$
```

2.2 Configuration Commands

This section describes the VLI configuration commands and includes all their possible options. You can eliminate or restore the default value for any command by entering NO in front of it.

2.2.1 [NO] APPLICATION

This command configures various parameters in the VLI application.

2.2.1.1 APPLICATION ADDRESS

Configures the IP address used by the applications server. This IP can be shared with the OS or be an exclusive (as described in sections **2.1** and **2.2** of the previous chapter). In cases where the IP is shared, you can specify the interface where the IP is obtained from so that the applications server uses the first active IP in said interface. As a result, the server can use an IP received through DHCP or PPP.

The OS automatically configures the IP in the applications server so that applications run over it and use this IP as source for the generated packets.

Syntax:

```
VLI Config$[no] application address ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
```

2.2.1.2 APPLICATION DNS-SERVER

Configures one or several DNS servers used by the applications server as the primary and secondary names server.

Syntax:

```
VLI Config$[no] application dns-server <a.b.c.d>
```

2.2.1.3 APPLICATION EPHEMERAL-PORTS

Configures the range of ephemeral ports used by the applications server. These ports are used as source for the TCP and UDP packets sent as client. The default range goes from 40960 to 61000. This range is reserved so that it cannot be used in the OS and to avoid collisions in cases where the IP is shared.

Syntax:

```
VLI Config$[no] application ephemeral-ports min <1024..65535> max <1024..65535>
```

2.2.1.4 APPLICATION TRAFFIC-DIVERT

Configures an access list to divert traffic to the applications server. If this command is configured, it analyses the traffic the OS is going to route to see if it matches the access list. If it does, the traffic is rerouted to the server interrupting the processing executed by the OS. As a result, matching packets are not routed.

Syntax:

```
VLI Config$[no] application traffic-divert access-list <1..9999>
```

2.2.1.5 APPLICATION TRAFFIC-EXPORT ACCESS-LIST

Configures the access list to export traffic to the applications server. If this command is configured, it analyses the traffic the OS is going to route to see if it matches the access list. If it does, a copy of the packet is sent to the server and the OS continues processing the packet.

Syntax:

```
VLI Config$[no] application traffic-export access-list <1..9999>
```

2.2.1.6 APPLICATION TRAFFIC-EXPORT BROADCAST

If you configure this command, a copy of the packets destined to the broadcast IPs is sent to the applications server.

Syntax:

```
VLI Config$[no] application traffic-export broadcast
```

2.2.1.7 APPLICATION TRAFFIC-EXPORT MULTICAST-GROUP

Allows you to configure multicast groups that the applications server wants to join so it receives a copy of the received packets addressed to these groups. Additionally, the OS maintains the corresponding IGMP subscription based on the IGMP proxy configuration. This must be enabled in order for it to function. There is an example of this further on.

Syntax:

```
VLI Config$[no] application traffic-export multicast-group <a.b.c.d>
```

2.2.1.8 APPLICATION VRF

Configures the VRF that the VLI module is connected to. This is the main VRF by default.

Syntax:

```
VLI Config$[no] application vrf <word>
```

2.2.2 [NO] BUFFER-DESCRIPTORS

Configures the number of buffer descriptors used by the shared memory virtual driver. The default value is 512. We recommend using the default value.

Syntax:

```
VLI Config$[no] buffer-descriptors <64..4098>
```

2.2.3 [NO] BUFFER-RING-SIZE

Configures the size of the descriptor rings used by the shared memory virtual driver. The default value is 512. We recommend using the default value.

Syntax:

```
VLI Config$[no] buffer-ring-size <64..4098>
```

2.2.4 [NO] RELOAD

Configures whether the device should reboot in cases where the applications server drops. This is enabled by default, meaning the device reboots whenever there is a serious problem in the applications server. Although we do not recommend doing it, this reboot can be disabled so that the OS continues executing if the server drops.

Syntax:

```
VLI Config$[no] reload on application-crash
```

Chapter 3 Monitoring

3.1 Accessing the Monitoring Menu

You can access the VLI monitoring menu from the router's monitoring console. Please enter the following sequence of commands:

```
*monitor
Console Operator

+feature vli
-- VLI monitor --
vli+
```

Once you have accessed the VLI monitoring menu, you can enter the following commands.

3.2 Monitoring Commands

3.2.1 APPLICATION

Allows you to manage various aspects of the applications server.

3.2.1.1 APPLICATION BOOT

Configures the boot mode for the applications server. If there is an applications server installed, the corresponding device will appear. If you select the **none** option, the applications server will not boot. You can find out if the server is booted or not and what the boot configuration is for the next reboot through the **application list** command.

Syntax:

```
vli+application boot device ?
  hdd     250.0 GB - Hard Disk Drive -   WDC WD2500BEVT-00A0RT0
  none    Do not boot application server
```

3.2.1.2 APPLICATION CHECK

Allows you to check for errors on the internal disk that isn't being used as the applications server. To carry this out, you need to boot the device with the applications server disabled through the **application boot device none** command.

Syntax:

```
vli+application boot check ?
  hdd     250.0 GB - Hard Disk Drive -   WDC WD2500BEVT-00A0RT0
vli+application boot check hdd
/dev/sda1: clean, 23674/15220736 files, 1198959/60862208 blocks
Device check done
vli+
```

3.2.1.3 APPLICATION FORMAT

Allows you to format a device on the applications server.

Syntax:

```
vli+application format <device> [yes]
```

- optional parameter **yes** allows the device to run an operation without prompting the user to confirm first. If this parameter is set to **yes**, no such confirmation is required. If not, the device prompts the user for confirmation.

Command history:

Release	Modification
11.01.06	The "[yes]" option was added as of version 11.01.06.

3.2.1.4 APPLICATION INSTALL

Allows you to install the applications server in an internal storage device using an .iso installation file. To execute this installation, the applications server must not be booted (see the **application boot** command).

The installer file can be found in a storage device connected to the external USB connector or downloaded from an HTTP server through the network.

If you use an external USB, you need to specify the .iso file to use in order to install. If you don't specify a file, the first one found will be installed.

When downloading from the HTTP url installer, you need to configure the **application address** command from the router's configuration menu (see chapter 2 of this manual) with a valid IP that can connect with the server. If not, the download will fail. You also need to enter the IP of the HTTP server in the URL, as DNS resolution is not supported.

Syntax:

```
vli+application install device ?
   hdd     250.0 GB - Hard Disk Drive -   WDC WD2500BEVT-00A0RT0
vli+application install device hdd from ?
   ext_usb    External USB
      file    ISO file to install
      <cr>
   url        URL to download
      <1..100 chars>    Text
```

Example:

```
vli+application install device hdd from url http://192.168.212.1/debian/images/
 debian-frodo-powerpcspe-NETINST-1.iso
Are you sure to overwrite frodo 8.0  on hdd(Yes/No)? y
Install:   begin/new_installation                                    [OK]
Install:   image/downloading_iso_image                               [OK]
Install:   prepare/runnig_init                                       [OK]
Install:   hw-detect/detect_progress_step                            [OK]
Install:   partman/progress/init/fallback                            [OK]
Install:   partman-auto/progress/info                                [OK]
Install:   partman-basicfilesystems/progress_formatting_mountable    [OK]
Install:   base-installer/section/cleanup                            [OK]
Install:   apt-setup/progress/volatile                               [OK]
Install:   debian-installer/pkgsel/title                             [OK]
Install:   pkgsel/progress/init                                      [OK]
Install:   pkgsel/progress/kdesudo                                   [OK]
Install:   pkgsel/progress/fallback                                  [OK]
Install:   pkgsel/progress/laptop-detect                             [OK]
Install:   pkgsel/progress/fallback                                  [OK]
Install:   pkgsel/progress/install-hwpackages                        [OK]
Install:   pkgsel/progress/fallback                                  [OK]
Install:   pkgsel/progress/save-logs                                 [OK]
Install:   pkgsel/progress/fallback                                  [OK]
Install:   pkgsel/progress/popcon                                    [OK]
Install:   pkgsel/progress/fallback                                  [OK]
Install:   pkgsel/progress/tasksel                                   [OK]
Install:   pkgsel/progress/cleanup                                   [OK]
Install:   finish-install/progress/umount                            [OK]

Install done!
vli+
```

3.2.1.5 APPLICATION LIST

Displays the following information on the applications server:

- If the applications server is running.
- The boot configuration for the next startup.
- If the automatic restart is enabled after installing the applications server.
- If an installable ISO in the external USB connector has been detected.
- Lists the storage devices detected in the device.

Syntax:

```
vli+application list
```

Example:

```
vli+application list

Application frodo 8.0  running on hdd
Next boot configuration: Run hdd application
Automatic reload after installation: OFF.
Application installer debian-frodo-powerpcspe-NETINST-1.iso  found on ext_usb
Device  App Installed   Description
hdd     frodo 8.0       250.0 GB - Hard Disk Drive -   WDC WD2500BEVT-00A0RT0
ext_usb none            4001 MB - External USB    - KingstonDataTraveler 410
```

3.2.1.6 APPLICATION RELOAD

Allows you to automatically reload the configuration right after the installation process.

Syntax:

```
vli+application reload automatic [yes]
```

- optional parameter **yes** allows the device to run an operation without prompting the user to confirm first. If this parameter is set to **yes**, no such confirmation is required. If not, the device prompts the user for confirmation.

Command history:

Release	Modification
11.01.06	The "[yes]" option was added as of version 11.01.06.

Chapter 4 Configuration Examples

4.1 Basic Configuration

Here we are going to show a basic configuration where the device only uses the ethernet0/0 interface and has only one IP (shared between the OS and the applications server). There is also a default route:

```
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.212.179 255.255.255.0
;
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 192.168.212.1
;
   exit
;
;
   feature vli
; -- VLI configuration --
      application address ethernet0/0
   exit
;
```

If the IP configuration has to be obtained through DHCP, the configuration is as follows:

```
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address dhcp-negotiated
;
   exit
;
   feature vli
; -- VLI configuration --
      application address ethernet0/0
   exit
```

The configuration below shows an applications server with an exclusive IP address (1.1.1.1). To access the applications server, network routing must be configured to reach IP 1.1.1.1 through the device with IP address 192.168.212.179:

```
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.212.179 255.255.255.0
;
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 192.168.212.1
;
   exit
;
;
   feature vli
; -- VLI configuration --
      application address 1.1.1.1
   exit
```

4.2 DLNS Application Configuration

An application example in the applications server can be a DLNA server. Device discovery is executed through SSDP using the multicast IP address 239.255.255.250. As a result, packets must be sent and received from this multicast group so the application functions properly.

Here we have a configuration example. You need to enable the OS's IGMP Proxy feature and configure the **ethernet0/0** interface as upstream. The interface will then act as host. In the VLI configuration, configure the **application traffic-export multicast-group** command to register the 239.255.255.250 multicast group in the IGMP proxy and to export the packets received for this group to the applications server.

```
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.212.179 255.255.255.0
;
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 192.168.212.1
;
      proxy-igmp
; -- IGMP proxy user configuration --
         enable
         upstream ethernet0/0 default
      exit
;
   exit
;
;
   feature vli
; -- VLI configuration --
      application address ethernet0/0
      application traffic-export multicast-group 239.255.255.250
   exit
;
```

4.3 Configuring Intrusion Detection

Another application that can be executed in the applications server is an intrusion detector that analyses traffic by searching for known attack patterns and generates alarms and reports.

In the following example, a device acts as the router between two Ethernet interfaces. Ethernet0/0 acts as the LAN and ethernet0/1 as WAN. You want the applications server to receive a copy of the traffic entering through the LAN in order to analyze it.

Configure a route-map to assign a label to the traffic entering through the eithernet0/0 interface. This label is used in the access list that classifies the traffic exported to the applications server. Once done, configure an exclusive IP address in the applications server. Otherwise, if the server shares an IP with the OS, it will respond to traffic entering through the LAN and addressed to a OS service in the shared IP (since this traffic always ends up being exported to the server).

```
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 label 10
;
      exit
;
   exit
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.212.179 255.255.255.0
;
```

```
      ip policy route-map MAP1
;
   exit
;
   network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip address 192.168.1.2 255.255.255.0
;
   exit
;
   feature route-map
; -- Route maps user configuration --
      route-map "MAP1"
         entry 1 default
         entry 1 permit
         entry 1 set label 10
;
      exit
;
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 192.168.1.1
;
   exit
;
   feature vli
; -- VLI configuration --
      application address 1.1.1.1
      application traffic-export access-list 100
;
   exit
;
```

4.4 Configuring a Transparent Web Proxy

If we want to implement a transparent web proxy in the applications server, the OS must intercept the WEB traffic and redirect it to the applications server. To do this, configure the **application traffic-divert** command.

As in the above example, the device acts as the router between the ethernet0/0 (LAN) and the ethernet0/1, which acts as WAN.

An access list is configured to classify the traffic destined to TCP port 80. This traffic is diverted to the applications server, preventing it from being normally processed in the OS. As a result, the proxy application can download the requested resources and respond to clients (thus allowing, for instance, cache implementation or content filtering).

The configuration would be as follows:

```
feature access-lists
; -- Access Lists user configuration --
      access-list 101
         entry 1 default
         entry 1 permit
         entry 1 destination port-range 80 80
         entry 1 protocol tcp
;
      exit
;
   exit
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.212.179 255.255.255.0
;
   exit
;
```

```
   network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip address 192.168.1.2 255.255.255.0
;
   exit
;
   feature vli
; -- VLI configuration --
      application address ethernet0/1
      application traffic-divert access-list 101
;
   exit
```