

Configuring Virtual Routers

10

The ERX system allows you to create multiple logical or *virtual* routers in a single system. Each virtual router has its own separate set of IP interfaces, forwarding table, and instances of routing protocols.

Topic	Page
Overview	10-1
References	10-3
Configuring Virtual Routers	10-3
Monitoring Virtual Routers	10-8

Overview

Multiple distinct routers are supported within a single system, which allows service providers to configure multiple, separate, secure routers within a single chassis. These routers are identified as *virtual routers* (VRs). Applications for this function include the creation of individual routers dedicated to wholesale customers, corporate virtual private network (VPN) users, or a specific traffic type.

Default Virtual Router

When you first boot your system, it creates a *default* virtual router. The only difference between the default VR and any other router is that you cannot create or delete the default VR. Just like any other router, the default VR gets its IP addresses when you add interfaces to it.

Virtual Router Instances

Your system can support up to 1,000 forwarding tables; that is, up to a total of 1,000 VRs and VPN routing and forwarding (VRF) instances. Each VRF has a forwarding table. A network device attaching to a system sees a router interface. The attaching device has no notion of the *virtual* router behind the interface.

For example, a physical ATM link may have circuits that are connected to different VRs. The physical and data link layers are not aware that there are multiple router instances. See Figure 10-1.

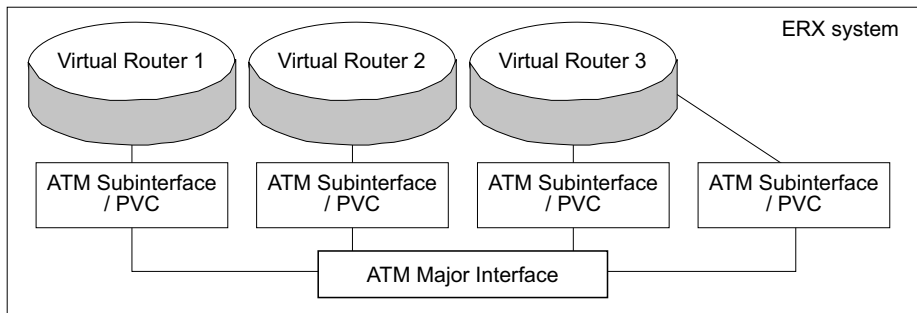


Figure 10-1 Virtual routers

VRs and VRFs are tools for implementing VPNs.

Routing Protocols

Your system implements the VRs by maintaining a separate instance of each data structure for each VR and allowing each protocol (for example, TCP/UDP, RIP, OSPF, and IS-IS) to be enabled on a case-by-case basis. A table of router interfaces associates user connections (for example, PPP or ATM) with one or more IP interfaces within a VR.

VPNs and VRFs

Your system supports VPNs and VRFs. For information on VPNs and VRFs, see *Configuring BGP VPN Services* and *Monitoring BGP/MPLS VPNs* in *ERX Routing Protocols Configuration Guide, Vol. 2, Chapter 3, Configuring BGP/MPLS VPNs*.

VPNs

A VPN is a set of sites attached to a common network, but whose data is handled separately from that common network.

VPNs enable private IP traffic to travel over a public TCP/IP network by tunneling that traffic between VPN member sites. Different levels of security are available depending on the security of the tunnel used between sites.

Your system supports VPNs consisting of VRs or VRFs. See RFC 2547 – *BGP/MPLS VPNs*. Additionally, your system supports tunnels built from GRE, IPSec, L2TP, MPLS, and tunnels built from layer 2 circuits, such as Frame Relay and ATM.

VRFs

A VRF is a virtual routing and forwarding instance that exists within the context of a VR. The VRF provides forwarding information to your system. The system looks up a packet's destination in the VRF associated with the interface on which the packet is received. In general, any application that can be enabled in a VR can be enabled in a VRF. VRFs are generally associated with the VPN behavior described in RFC 2547.

When a VRF receives an update message, it needs to know whether it should add the route to its routing table. Similarly, when a VRF sends update messages, it needs to identify the VPNs that it wants to receive the updates. See *ERX Routing Protocols Configuration Guide, Vol. 2, Chapter 3, Configuring BGP/MPLS VPNs*.

References

For more information about virtual routers, VPNs, or VRFs, consult the following resources:

- *ERX Release Notes, Appendix A, System Maximums* – refer to the Release Notes corresponding to your software release for information on maximum values.
- *ERX Routing Protocols Configuration Guide, Vol. 2, Chapter 3, Configuring BGP/MPLS VPNs*
- RFC 2547 – BGP/MPLS VPNs (March 1999)
- RFC 2917 – A Core MPLS IP Architecture (September 2000)

Configuring Virtual Routers

This section provides examples of some of the more common virtual router tasks.

There are different uses of the **virtual-router** command. You can create or access VRs and VRFs in Global Configuration mode or map a VR to a domain map in Domain Map Configuration mode. Once you create a VR, you can continue to work in different command modes and configure the same user interface parameters as before the virtual router was created.



Note: For information on the many VR tasks you can configure, see the related chapter; for example, *Configuring IP* or *Configuring BGP*.

- Create and name a VR in Configuration mode.

```
host1(config)#virtual-router western
host1:western(config)#
```

- Create a VRF to provide forwarding information to your system. In this example, the VRF created is in context with the VR created above.

```
host1:western(config)#ip vrf eastern
Proceed with new VRF creation?[confirm]
host1:western(config-vrf)#virtual-router:eastern
host1:western:eastern(config)#
```

- Access a VRF from the context of a different VR.

```
host1(config)#virtual-router western:eastern
host1:western:eastern(config)#
```

- View your configuration choices from a VR or VRF context.

```
host1:western:eastern(config)#?
aaa                Configure authentication, authorization,
                  and accounting characteristics
access-list        Configure an access list entry
arp                Configure a static ARP entry
bandwidth          Configure slot-group bandwidth control
banner             Define a banner line
baseline           Configure baseline operations
boot               Configure boot time behavior
bulkstats          Configure bulkstats parameters
cbf                Configure connection-based forwarding
classifier-list    Configure a classifier list entry
clns               Configure CLNS characteristics
clock              Set the system's clock
controller         Configure controller parameters
crypto             Configure cryptographic parameters
disable-autosync   Disable automatic synchronization of
                  redundant system controller file system
disable-switch-on-error Disable automatic switch to redundant system
                  controller upon software/hardware error
```

enable	Configure security related options
end	Exit Global Configuration mode
exception	Configure core dump
exclude-subsystem	Exclude copying a subsystem from the release
exit	Exit from the current command mode
ftp-server	Configure FTP Server characteristics
help	Describe the interactive help system
host	Add/modify an entry to the host table
hostname	Set the host (system) name
interface	Enter Interface Configuration mode
ip	Configure IP characteristics
l2f	Configure L2F parameters
l2tp	Configure L2TP parameters
license	Configure licenses
line	Enter Line Configuration mode
log	Configure logging settings
macro	Run a CLI macro
map-list	Create an NBMA static map
memory	Configure and administer memory operations
mpls	Configure MPLS global parameters
no	Negate a command or set its default(s)
ntp	Configure the Network Time Protocol
policy-list	Enter Policy Configuration mode
pppoe	Configure PPPoE
profile	Specify a profile
radius	Configure RADIUS server
rate-limit-profile	Enter rate limit profile configuration mode
redundancy	Perform a redundancy configuration
route-map	Configure a route map
router	Configure a routing protocol
rtr	Configure rtr parameters
service	Configure system-level services
set	Configure
sleep	Make the Command Interface pause for a specified duration
slot	Configure and administer slot operation
snmp-server	Configure SNMP parameters
sscc	The SSC Client
telnet	telnet daemon configuration
timing	Configure network timing
traffic-shape-profile	Enter traffic shape profile configuration mode
virtual-router	Specify a virtual router

host1:western:western(config)#

- View the VRF configuration choices from VRF Configuration mode.

host1:western(config-vrf)#?

```

exit          Exit from the current command mode
export       Specify VRF export characteristics
help        Describe the interactive help system
import      Specify VRF import characteristics
log         Configure logging settings
macro       Run a CLI macro
no          Negate a command or set its default(s)
rd          Specify route distinguisher
route-target Specify VPN extended community Target
sleep       Make the Command Interface pause for a
            specified duration
host1:western(config-vrf)#

```

- Access a VR to configure it with an interior gateway protocol (IGP) or exterior gateway protocol (EGP) to learn routes from a customer edge device (CE). See the related routing protocol chapters for detailed information.

Example 1
VR with an
IGP

```

host1(config)#virtual-router miami
host1:miami(config)#router ospf 5
host1:miami(config-router)#

```

Example 2
VR with an
EGP

```

host1(config)#virtual-router western
host1:western(config)#router bgp 359
host1:western(config-router)#

```

- Configure a Telnet daemon to listen in VRs other than the default VR.

```

host1(config)#virtual-router boston
host1(config)#telnet listen port 23

```

- List all VRs and VRFs on the system.

```

host1#show virtual-router
Virtual Router : default
Virtual Router : thursday
Virtual Router : western
                  VRF : eastern
Virtual Router : boston
Virtual Router : miami
Virtual Router : northern
                  VRF : southern
host1#

```

- Map a VR to a user domain name in Domain Map Configuration mode. The VR must already exist.

```
host1(config)#aaa domain-map jacksonville
host1(config-domain-map)#virtual-router western
host1(config-domain-map)#
```

aaa domain-map

- Use to map a user domain name to a virtual router.
- Examples


```
host1-0-1-90(config)#aaa domain-map juniper.net vrouter_1
host1-0-1-90(config)#aaa domain-map none
vrouter__all_purpose
host1-0-1-90(config)#aaa domain-map DEFAULT
vrouter_all_purpose
```
- Use the **no** version of the command to delete the domain map.

ip vrf

- Use to create a VRF or access VRF Configuration mode to configure a VRF.
- You must specify a route distinguisher after you create a VRF. Otherwise, the VRF will not operate.
- Example


```
host1-00-02-80:boston(config)#ip vrf vpn-A
```
- Use the **no** version to remove a VRF.

telnet listen

- Use to create a Telnet daemon to listen in a virtual router.
- Example


```
host1(config)#virtual-router 3
host1(config)#telnet listen port 3223
```
- Use the **no** version of the command to delete the daemon.

virtual-router

- From Global Configuration mode, use this command to create a virtual router or access the context of a previously created virtual router or a VRF.
- From Domain Map Configuration mode, use this command to map the VR to a user domain name. Use the **no** version in this mode to delete the VR parameter and assign the default VR.
- A VR name consists of up to 15 alphanumeric characters.
- Once you are in the context of a particular VR or VRF (indicated by the change in the prompt), all subsequent commands you enter apply to that context until you exit the context.
- Use the **no** version of the command only to delete the VR and return the system to the default VR. Issuing the command **no virtual-router vrName.vrfName** has no effect.
- Issuing a **no** version of this command (**no virtual-router :vrfName** or **no virtual-router vrName:vrfName**) that specifies an existing VRF only

displays the error message: "Cannot delete a VRF with this command." You must use the **no ip vrf** command to remove a VRF.



Note: See *ERX Command Reference Guide* for additional information.

Monitoring Virtual Routers

Use the **show virtual-router** and **show aaa domain-map** commands to display virtual router and user-domain-to-virtual-router mapping information. Use the **show ip forwarding table** command to display information on memory usage by virtual routers.

show aaa domain-map

- Use to display the mapping between user domains and virtual routers.
- The following keywords have significance when used as user domains:
 - › **none** – all client requests with no user domain name are associated with the virtual router mapped to the *none* entry
 - › **default** – all client requests with a domain present that has no map are associated with the virtual router mapped to the *default* entry
- Example

```
host1#show aaa domain-map
Domain: boston; virtual-router: default

Tunnel Tunnel Tunnel Tunnel Tunnel Tunnel Tunnel Tunnel
Tag Peer Source Type Medium Password Id Hostname
-----
31 <null> <null> l2tp ipv4 <null> <null> <null>

Tunnel
Tunnel Server Tunnel
Tag Name Preference
-----
31 <null> 2000
```

show configuration virtual-router

- Use to display configuration information for the virtual routers configured on your system.
- You can create a configuration script from the output by saving it as a file with the .scr extension.
- You can exclude information about a particular type of interface.
- You can use the output filtering feature of the **show** command to include or exclude lines of output based on a text string you specify. See *Chapter 2, Command Line Interface*, for details.
- Example

```
host1#show configuration virtual-router default
virtual-router default
ip domain-lookup
ip name-server 10.2.0.3
ip domain-name "junipercom.com"
!
host f 10.10.0.129 ftp anonymous null
interface null 0
!
interface fastEthernet 0/0
  ip address 192.168.1.155 255.255.255.0
!
ip route 0.0.0.0 0.0.0.0 192.168.1.1
no ip multicast-routing
!
mpls rsvp profile default
mpls ldp profile default
  cr-ldp
!
rtr 1
  type echo protocol ipIcmpEcho 10.5.0.200 source
    fastEthernet0/0
  frequency 1
  samples-of-history-kept 5
  timeout 10000
!
```

show ip forwarding-table slot

- Use to display the memory used by each VR configured on a line module and free memory available on the line module.
- Field descriptions
 - › Free Memory – amount of memory free on the line module, in kilobytes
 - › Virtual Router – name of the virtual routers configured on the line module
 - › Memory (KB) – amount of memory consumed by the VR, in kilobytes
 - › Load Errors – counts errors made while loading the routing table on the line module

› Status – indicates whether the routing table for the VR is valid

- Example

```
host1#show ip forwarding-table slot 9
```

```
Free Memory = 14,328KB
```

Virtual Router	Memory (KB)	Load Errors	Status
vr1	4128	0	Valid
vr2	3136	0	Valid
vr3	2256	0	Valid
vr4	1512	0	Valid
default	1024	0	Valid

show virtual-router

- Use to display virtual routers configured on your system.
- The display shows the name of the virtual router and the status of the supported protocols.
- You can use the output filtering feature of the **show** command to include or exclude lines of output based on a text string you specify. See *Chapter 2, Command Line Interface*, for details.
- Example

```
host1-0-6-60#show virtual-router
```

```
Virtual Router : default
```

```
Ip:      Present
Bgp:     Present
Isis:    Present
Ospf:    Not Present
Rip:     Present
```

```
Virtual Router : two
```

```
Ip:      Present
Bgp:     Not Present
Isis:    Not Present
Ospf:    Not Present
Rip:     Present
```

```
Virtual Router : three
```

```
Ip:      Present
Bgp:     Not Present
Isis:    Not Present
Ospf:    Not Present
Rip:     Present
```