Virtual Route Forwarding Design Guide Secure Softphone Connectivity

This document provides overview, technology description, design considerations, sample configuration and other information for connecting softphone securely by using Virtual Private Network Routing and Forwarding (VRF) with your network. Cisco Unified CM

Contents

Introduction

Virtualization is a technique for hiding the physical characteristics of computing resources from the way in which other systems, applications, or end users interact with those resources. This includes making a single physical resource (such as a server, an operating system, an application, storage device, or network) appear to function as multiple logical resources; or it can include making multiple physical resources (such as storage devices or servers) appear as a single logical resource.

Virtual networks is a generic term that uses many different technologies to provide *virtualization.* Fundamentally, virtual networks all provide a mechanism to deploy what looks and operates like multiple networks, and are actually all using the same hardware and physical connectivity.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

A distinction needs to be made among the types of virtualization and which layer this network virtualization occurs at:

- Physical (Layer 1)—A Time Division Multiplexer (TDM) provides a way to make a single physical connection look like many physical connections, while still maintaining separation.
- Datalink (Layer 2)—Frame Relay, Asynchronous Transfer Mode (ATM), and Ethernet switches are all examples of how a single physical link may provide multiple logical or virtual connections per physical connection.
- Network (Layer 3)—Routers are examples of how multiple sessions can be carried over a single connection using IP addresses as the identifier. Routers use IP addresses to direct the data to the correct destination. When an IP packet is received the destination address is looked up in a route table to determine the next hop to send the packet to. Normally all packets within a physical router use the same route table or global table.

Virtualization of Networks

If we needed IP networks that are isolated as they were used by different companies, departments, or organizations, we would normally deploy multiple IP networks made up of separate physical routers that were not connected to each other. They may still be using a shared Layer 2 or Layer 1 infrastructure, however at Layer 3 they are not connected and do not form a network.

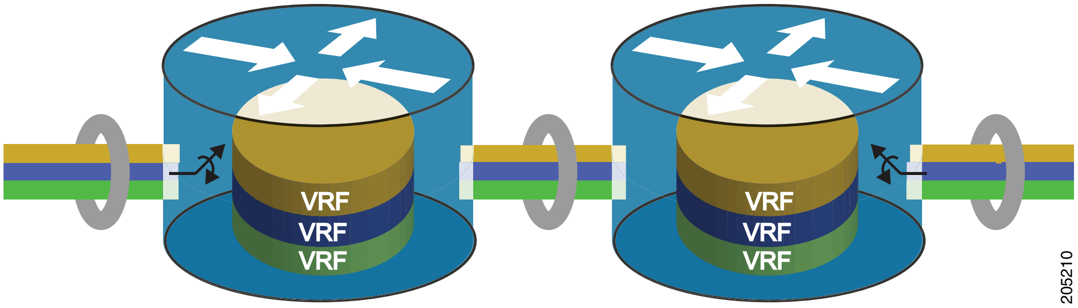
Network virtualization allows a single physical router to have multiple route tables. The global table contains all IP interfaces that are not part of a specific virtual network and route tables are for each unique virtual network assigned to an IP interface.

In its basic form this allows a FastEthernet 0/0 IP interface to be in virtual network 10 and FastEthernet 0/1 IP interface to be in virtual network 20. Packets arriving on FastEthernet 0/0 are only forwarded to other interfaces in virtual network 10 and do not use FastEthernet 0/1, because it is not in its virtual network: virtual network 10 has no routing knowledge of other virtual networks.

Additional virtualization can be provided by allowing multiple virtual networks per physical connection. This is enabled by using Layer 2 logical connections. For a FastEthernet physical port, the use of multiple virtual LANs (VLANs) allows each VLAN to use a different virtual network.

Virtual Route Forward (VRF) is a technique which creates multiple virtual networks within a single network entity (Figure 1). In a single network component, multiple VRF resources create the isolation between virtual networks.

Figure 1 Virtual Route Forwarding



VRF implementations in Cisco Unified Communications Manager Express (Cisco Unified CME) include:

- Single voice network and multiple data networks, which consolidate voice communication into one logically partitioned network to separate voice and data communication on a converged multimedia network.
- Enable Cisco Unified CME on an MPLS provider edge router.
- Enable Cisco Unified CME on multiple CE (VRF Lite) routers.
- Multiple voice networks and multiple data networks, which share a Cisco Unified CME by multiple closed users group with different requirements. Check the feature restrictions for details; VRF does not support identical IP addresses or shared-lines.

Topology

Cisco Unified CM supports Cisco IP phones and soft phones at remote sites attached to Cisco Integrated Service Routers (ISR) across the WAN. When you deploy voice, we recommends that you enable two LANs at the access layer: a native VLAN for data traffic and a voice VLAN under Cisco IOS or Auxiliary VLAN under CatOS for voice traffic.

Separate voice and data VLANs are recommended for the following reasons:

- Address space conservation and voice device protection from external networks—Private addressing of phones on the voice or auxiliary VLAN ensures address conservation and ensures that phones are not accessible directly through public networks. PCs and servers are typically addressed with publicly routed subnet addresses; however, voice endpoints should be addressed using RFC 1918 private subnet addresses.
- QoS trust boundary extension to voice devices—QoS trust boundaries can be extended to voice devices without extending these trust boundaries and, in turn, QoS features to PCs and other data devices.
- Protection from malicious network attacks—VLAN access control, 802.1Q, and 802.1p tagging can provide protection for voice devices from malicious internal and external network attacks such as worms, denial of service (DoS) attacks, and attempts by data devices to gain access to priority queues through packet tagging.
- Ease of management and configuration—Separate VLANs for voice and data devices at the access layer provide ease of management and simplified QoS configuration.

You can use VLAN access control lists (ACLs) to control data that flows on a network. Cisco switches have the capability of controlling Layers 2 to 4 within a VLAN ACL. Depending on the types of switches in a network, VLAN ACLs can be used to block traffic into and out of a particular VLAN. They can also be used to block intra-VLAN traffic to control what happens inside the VLAN between devices.

If you plan to deploy a VLAN ACL you should verify which ports are needed to allow the phones to function with each application used in your IP telephony network. Normally, any VLAN ACL is applied to the VLAN that the phones use. This allows control at the access port, as close as possible to the devices that are plugged into that access port.

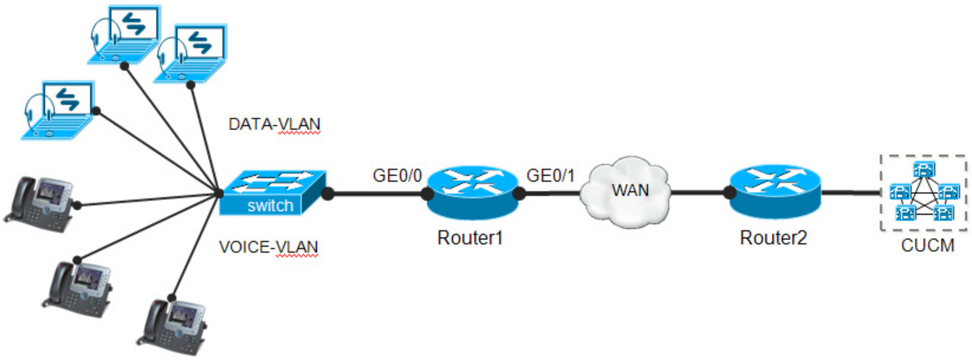
But, VLAN ACLs are very difficult to deploy and manage at an access-port level that is highly mobile. Because of these management issues, take care when deploying VLAN ACLs at the access port in the network.

As with VLAN ACLs, routers have the ability to process both inbound and outbound ACLs by port. The first Layer 3 device is the demarcation point between voice data and other types of data when using voice and data VLANs, where the two types of data are allowed to send traffic to each other. Unlike the VLAN ACLs, router ACLs are not deployed in every access device in your network. Instead, they are applied at the edge router, where all data is prepared for routing across the network. This is the perfect location to apply a Layer-3 ACL to control which areas the devices in each of the VLANs have the ability to access within a network. Layer-3 ACLs can be deployed across your entire network to protect devices from each other at points where the traffic converges.

But, as the ACLs become more granular and detailed, any changes in port usage in a network could break not only voice but also other applications in the network. If there are software phones in the network, if web access to the phone is allowed, or if you use the attendant console or other applications that need access to the voice VLAN subnets, the ACLs are much more difficult to deploy and control. By deploying VRFs with Trusted Relay Point (TRP) functionality into the network, we can solve this problem very easily and securely.

Figure 2 shows a branch office with several Cisco IP phones and soft phones connected to a Cisco router (Router1) via a switch that supports both DATA and VOICE VLAN. The router provides connections to a WAN link. The Cisco IP phones and soft-phones connect to their primary Cisco Unified CM at the central office via this WAN link.

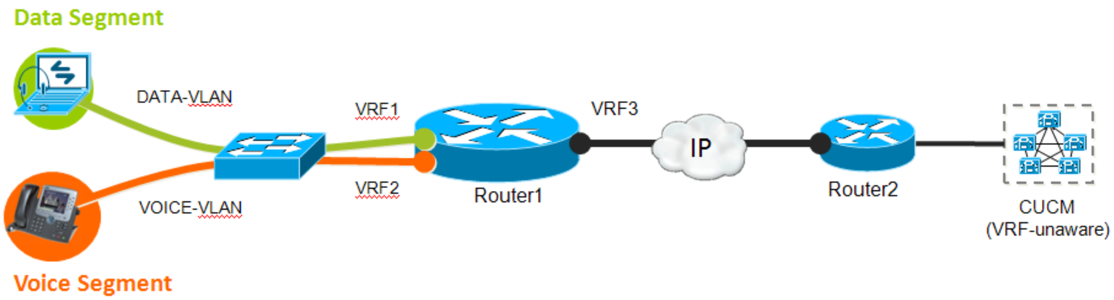
Figure 2 Branch Office Cisco IP Phones and Soft Phones are connected to Central Cisco Unified CM



To connect softphone securely within the enterprise by using VRF technology, we need to understand how VRFs work in this very specific case. Let's follow the steps to understand is better, before we jump into the configuration.

In Figure 3, we are showing the conceptual separation of networks - voice and data networks have been separated at layer-3 (Router1) level. In the switch, we have created two different VLANs i.e., DATA-VLAN and VOICE-VLAN. These two VLANs are connected to the Router1 through two different sub-interfaces.

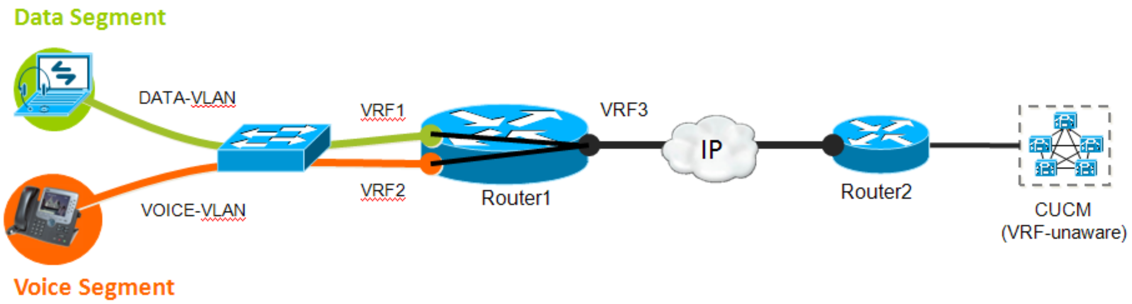
Figure 3 Network is segmented into two networks - Voice and Data



Now, we are going to create VRFs for each of the network segments. For this example, we have defined 3 VRFs i.e., VRF1, VRF2 and VRF3. As you know, VRF has no significance outside the router, and VRFs are tied with specific interfaces. Single VRF can be assigned to multiple interfaces; but multiple VRFs cannot be assigned to a single interface. In this specific example, we could put Gig0/1 under VRF1 as part of data network. But, to maintain the security, simplicity and clarity, we have put Gig0/1 under VRF3.

In Figure 4, all the interfaces (networks) are segmented. Under this definition, no interface can see other's traffic. Soft-phones under VRF1 cannot reach to Cisco Unified CM which is under VRF3. Same thing applies for hard phones under VRF2. We have just created three different segments of the network, where no traffic can flow among those segments. No phone can register to Cisco Unified CM as packet cannot flow between VRF1 and VRF3; and between VRF2 and VRF3.

Figure 4 ***Connecting Segments***



Now, it is time to connect those segmented network so that they can communicate securely. First of all, we want to create bridge between VRF1 and VRF3 so that soft phones can communicate with Cisco Unified CM. And, similarly we are going to connect a bridge between VRF2 and VRF3 so that hard phones also can communicate with Cisco Unified CM. In Figure 5, the connection has been shown by black lines (in Router1) between VRFs. So, under this situation, VRF1 and VRF2 cannot communicate each others, although VRF1 and VRF2 can communicate with VRF3 individually.

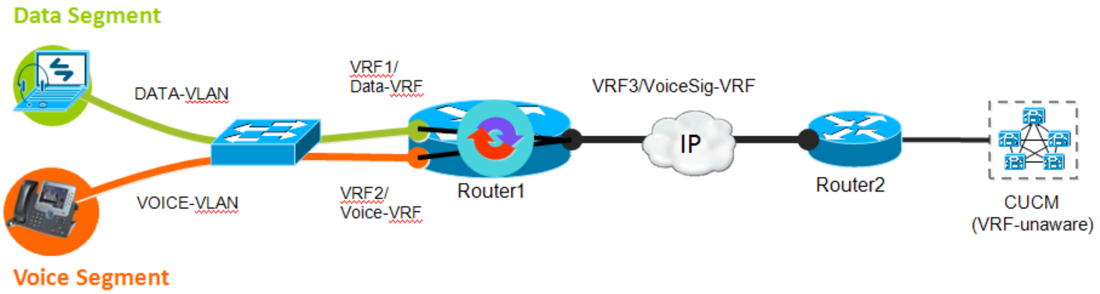
By using import/export command (importing/exporting routing tables), bridge can be created between 2 (two) VRFs. In this example, VRF3 will export only it's own routing table, and import both VRF1 & VRF2. Here is the logical representation of this exchange of routing tables -

```
!
VRF1
  export VRF1
  import VRF3
!
VRF2
  export VRF2
  import VRF3
!
VRF3
  export VRF3
  import VRF1
  import VRF2
!
```

Again, these are not the actual IOS CLI. We are just showing the logic behind creating the bridge. In our example, we will show the actual IOS commands those will enable the VRFs accordingly.

By importing/exporting VRFs, we define the rules to import and export routing tables among the VRFs. But we need some routing protocols to exchange the routes among those VRFs. BGP, OSPF or EIGRP can be used to inject the routes from one VRF to another VRF. This routing protocol may not be used to connect any other routers. It can be used locally to exchange routing tables among VRFs based on the rules set by export/import.

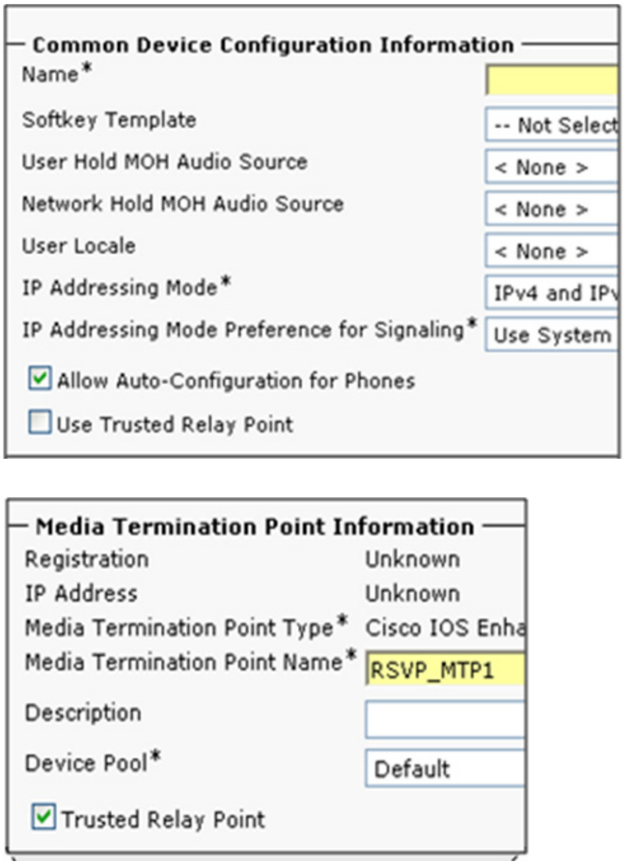
Figure 5 Cisco Unified CM in Segmented Network, connecting segments via TRP



Up to this level, both soft phones and hard phones can register to Cisco Unified CM and make calls within their own segments - soft phone can call to another soft phone and hard phone can call to another hard phone. But, they cannot make call between soft phones and hard phones. Now, we need to create some bridge so that traffic can flow between VRF1 and VRF2. This is the place where TRP comes into the picture. In Figure 6, TRP is used to create a bridge between VRF1 and VRF2.

TRP can be used to enable the VRF traversal functionality in the *Router1*. TRP can be configured in Cisco Unified CM as shown below.

Figure 6 Configuring TRP in Cisco Unified CM



Common Device Configuration Information

Name*	
Softkey Template	-- Not Select
User Hold MOH Audio Source	< None >
Network Hold MOH Audio Source	< None >
User Locale	< None >
IP Addressing Mode*	IPv4 and IPv
IP Addressing Mode Preference for Signaling*	Use System

☑ Allow Auto-Configuration for Phones

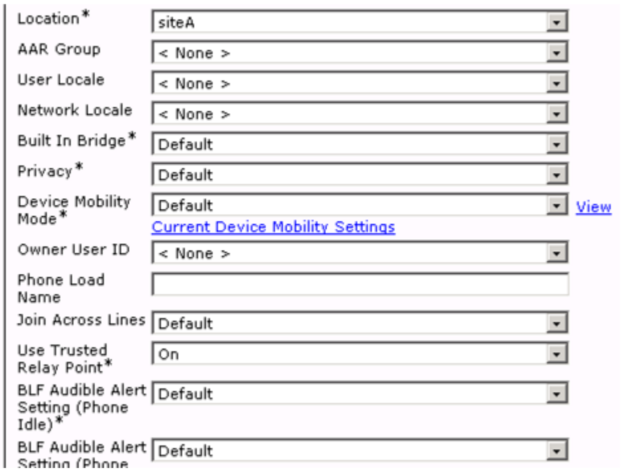
□ Use Trusted Relay Point

Media Termination Point Information

Registration	Unknown
IP Address	Unknown
Media Termination Point Type*	Cisco IOS Enha
Media Termination Point Name*	RSVP_MTP1
Description	
Device Pool*	Default

☑ Trusted Relay Point

Figure 7 Assigning TRP for phones in Cisco Unified CM

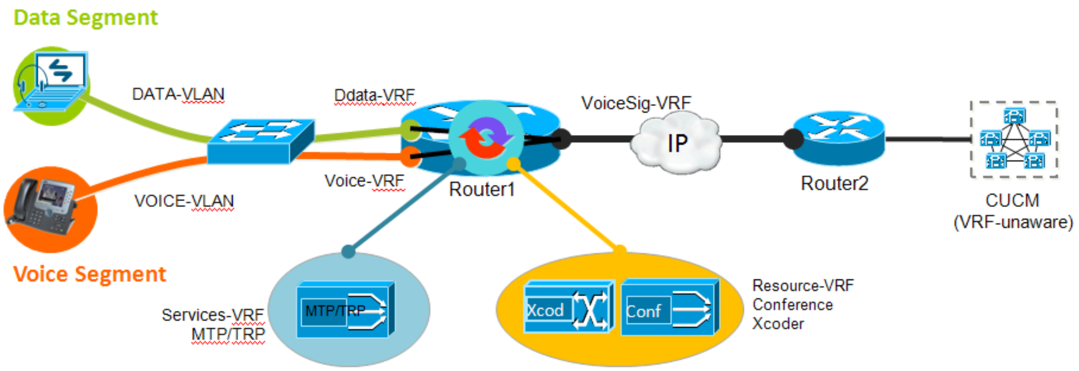


Location*	siteA
AAR Group	< None >
User Locale	< None >
Network Locale	< None >
Built In Bridge*	Default
Privacy*	Default
Device Mobility Mode*	Default View Current Device Mobility Settings
Owner User ID	< None >
Phone Load Name	
Join Across Lines	Default
Use Trusted Relay Point*	On
BLF Audible Alert Setting (Phone Idle)*	Default
BLF Audible Alert Setting (Phone	Default

Note To enable TRP in IOS Router, we don't need to configure anything extra, but configuring the MTP services.

We can use the same technique to have VRF traversal in multiple VRFs. It can be used to traverse multiple voice and data VRFs. In this example above, we have demonstrated how to connect softphone securely by using three VRFs and TRP functionality. But, it can be extended to more complex scenarios. We can put conferencing and transcoding services into different VRFs, so that we can control those devices too. In Figure 8, we have shown five different VRFs to segment the network more complex way.

Figure 8 MTP, Transcoding and Conferencing services under VRFs



In the configuration example, we have used this network topology with five VRFs to show you the actual working configuration.

Configuration

To configure a voice VRF, you must shut down voice services on the gateway, assign a previously defined VPN VRF to the VoIP SPI, and then restart voice services.

This section describes the tasks required to configure VRF-aware voice gateways.

Note If a voice VRF is not configured, signaling and media packets are sent using the default routing table.

Prerequisites

Be sure to check the following prerequisites before configuring a voice VRF:

- ?oTo ensure there are no active calls on the voice gateway during a VRF change, you must shut down the voice gateway before you configure or make changes to a voice VRF.
- ?oIf your configuration uses address binding, use the h323-gateway voip bind srcaddr ip-address command to bind the gateway to an interface that belongs to the voice VRF.
- ?oIf the voice gateway configuration has H.323 RAS enabled, use the h323-gateway voip interface command to configure RAS on the interface that belongs to the voice VRF.

Restrictions

Restrictions for configuring VRF-aware H.323 and SIP are as follows:

- If the voice gateway configuration has H.323 RAS enabled, the gatekeeper must be accessible to the gateway in the configured voice VRF.
- When voice VRF is configured, the H.323 gateway and gatekeeper cannot communicate with each other if they are running on same router.

Voice VRF supports only the following call types:

- A single VRF for SIP-to-SIP calls
- A single VRF for H323-to-SIP calls
- A single VRF for H323-to-H323 calls
- A single VRF in IP-to-IP gateway call with a gatekeeper involved, but the gatekeeper is not on the same router.
- A SIP SRST call
- A SCCP SRST call
- A SCCP CME call
- A SIP CME call

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrfname*
4. **rd** *route-distinguisher*
5. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
6. **exit**
7. **voice service voip**
8. **shutdown**
9. **exit**
10. **voice vrf** *vrfname*
11. **voice service voip**
12. **no shutdown**
13. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`ip vrf`** *`vrfname`*
	Example:
	`Router(config)# ip vrf vrf1`
		Defines a Virtual Private Network (VPN) routing/forwarding (VRF) instance and enters VRF configuration mode.
		• *vrfname*—Identifier for the VRF.

Step 4	**`rd`** *`route-distinguisher`*
	Example:
	`Router(config-vrf)# rd 1:1`
		Creates a routing and forwarding table for a VPN VRF.
		• *route-distinguisher*—Adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix.

Step 5	**`route-target {import | export | both} route-target-ext-community`**
	Example:
	`Router(config-vrf)# route-target export 1:2`
		Creates a list of import or export route target communities for the specified VRF.
		• **import**—Imports routing information from the target VPN extended community.
		• **export**—Exports routing information to the target VPN extended community.
		• **both**—Imports both import and export routing information to the target VPN extended community.
		• *route-target-ext-community*—Adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities.

Step 6	**`exit`**
	Example:
	`Router(config-vrf)# exit`
		Exits VRF configuration mode.

Step 7	**`voice service voip`**
	Example:
	`Router(config)# voice service voip`
		Enters voice-service configuration mode.

Step 8	**`shutdown`**
	Example:
	`Router(config-voi-serv)# shutdown`
		Shuts down voice services.

Step 9	**exit** **Example:** Router(config-voi-serv)# exit	Exits voice-service configuration mode.
Step 10	**voice vrf name** **Example:** Router(config)# voice vrf vrf1	Assigns a predefined VRF to voice services. • *vrfname*—Identifier for the VRF.
Step 11	**voice service voip** **Example:** Router(config)# voice service voip	Enters voice-service configuration mode.
Step 12	**no shutdown** **Example:** Router(config-voi-serv)# no shutdown	Restarts voice services.
Step 13	**end** **Example:** Router(config-voi-serv)# end	Returns to privileged EXEC mode.

Examples

```
!
ip vrf vrf1
 rd 100:1
 route-target export 100:1
 route-target import 100:2
!
voice vrf vrf1
!
voice service voip
!
```

Cisco IOS Global Configuration

The following example is part of a TRP configuration. There are two VRFs, VPN801 and VPN802, defined for two groups of phones. Cisco Unified Communication Manager (Cisco Unified CM) belongs to the third VRF and is named ccm.

The VPN801 and VPN802 imports the routes from the VRF ccm. VPN801 and VPN802 export routes to the VRF ccm. VPN801 and VPN802 can access VRF ccm, but are independent of each other's routes.

Default voice VRF is ccm, where SIP phones must belongs and registered Cisco Unified CME.

Initially, phones belongs to VRF VPN801 and VPN802 register to Cisco Unified CM.

```
!
ip vrf VRFdata
 rd 100:1
 route-target export 100:1
 route-target import 100:3
 route-target import 100:2
!
ip vrf VRFresource
 rd 100:5
 route-target export 100:5
 route-target import 100:3
 route-target import 100:2
!
ip vrf VRFservice
 rd 100:2
 route-target export 100:2
 route-target import 100:1
 route-target import 100:4
 route-target import 100:3
 route-target import 100:5
!
ip vrf VRFvoice
 rd 100:4
 route-target export 100:4
 route-target import 100:3
 route-target import 100:2
!
ip vrf VRFvoicesig
 rd 100:3
 route-target export 100:3
 route-target import 100:5
 route-target import 100:1
 route-target import 100:4
 route-target import 100:2
!
```

VRF Routing Configurations

As the VRFs are all different virtual networks, they cannot communicate among themselves without the exchange of routing information. Even though, they reside in a single box/router, but they are all independent virtual networks; they don't know how to reach IP-address of other networks. We can use VRF-enabled routing protocols to exchange routing tables among themselves.

The following is an example BGP configuration showing how to exchange routing information among the five VRF resources defined earlier.

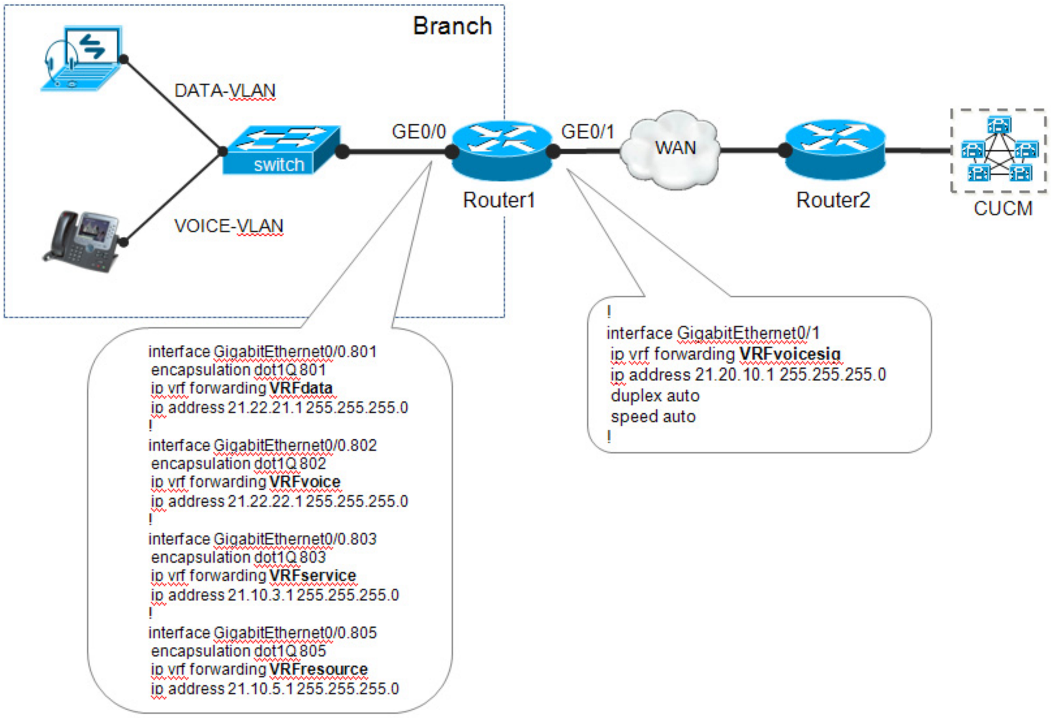
```
!
router bgp 1000
 no synchronization
 bgp router-id 22.22.22.22
 bgp log-neighbor-changes
 no auto-summary
 !
 address-family ipv4 vrf VRFvoicesig
  redistribute connected
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf VRFvoice
  redistribute connected
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf VRFservice
  redistribute connected
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf VRFresource
  redistribute connected
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf VRFdata
  redistribute connected
  no synchronization
 exit-address-family
!
```

Notice that, BGP does not have any neighbor; and this BGP is not used to make Border Routing implementation. BGP is used only to exchange routing tables among the VRFs within the same router via IPv4 VRF extension. Other routing protocols, for example, OSPF and EIGRP, can also be used to achieve the same purpose.

Softphone Connectivity Configuration Example

Figure 9 shows a configuration topology for VRF-Aware SIP call flows on Cisco Unified CME.

Figure 9 Topology for Softphone Connectivity with MTP and Transcoding/Conferencing facility



```
interface GigabitEthernet0/0.801
 encapsulation dot1Q 801
 ip vrf forwarding VRFdata
 ip address 21.22.21.1 255.255.255.0
!
interface GigabitEthernet0/0.802
 encapsulation dot1Q 802
 ip vrf forwarding VRFvoice
 ip address 21.22.22.1 255.255.255.0
!
interface GigabitEthernet0/0.803
 encapsulation dot1Q 803
 ip vrf forwarding VRFservice
 ip address 21.10.3.1 255.255.255.0
!
interface GigabitEthernet0/0.805
 encapsulation dot1Q 805
 ip vrf forwarding VRFresource
 ip address 21.10.5.1 255.255.255.0
```

```
!
interface GigabitEthernet0/1
 ip vrf forwarding VRFvoicesig
 ip address 21.20.10.1 255.255.255.0
 duplex auto
 speed auto
!
```

Router-1 Configuration:

Here is the full configuration of the ROUTER-1 running on ISR platform:

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router1
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
logging buffered 999999
!
no aaa new-model
!
ip source-route
!
!
ip cef
!
!
ip vrf VRFdata
 rd 100:1
 route-target export 100:1
 route-target import 100:3
 route-target import 100:2
!
ip vrf VRFresource
 rd 100:5
 route-target export 100:5
 route-target import 100:3
 route-target import 100:2
!
ip vrf VRFservice
 rd 100:2
 route-target export 100:2
 route-target import 100:1
 route-target import 100:4
 route-target import 100:3
 route-target import 100:5
!
ip vrf VRFvoice
 rd 100:4
 route-target export 100:4
 route-target import 100:3
 route-target import 100:2
!
ip vrf VRFvoicesig
 rd 100:3
 route-target export 100:3
 route-target import 100:5
 route-target import 100:1
 route-target import 100:4
 route-target import 100:2
!
no ipv6 cef
multilink bundle-name authenticated
!
!
voice dsp waitstate 0
!
```

```
voice service voip
 allow-connections h323 to h323
 allow-connections h323 to sip
 allow-connections sip to h323
 allow-connections sip to sip
 shutdown
 supplementary-service h450.12
 h323
  call start slow
 sip
  registrar server expires max 3600 min 3600
!
!
voice-card 0
 no dspfarm
 dsp services dspfarm
!
!
archive
 log config
  hidekeys
!
!
interface GigabitEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface GigabitEthernet0/0.801
 encapsulation dot1Q 801
 ip vrf forwarding VRFdata
 ip address 21.22.21.1 255.255.255.0
 ip helper-address 21.20.10.11
!
interface GigabitEthernet0/0.802
 encapsulation dot1Q 802
 ip vrf forwarding VRFvoice
 ip address 21.22.22.1 255.255.255.0
 ip helper-address 21.20.10.11
!
interface GigabitEthernet0/0.803
 encapsulation dot1Q 803
 ip vrf forwarding VRFservice
 ip address 21.10.3.1 255.255.255.0
!
interface GigabitEthernet0/0.804
 encapsulation dot1Q 804
 ip vrf forwarding VRFvoicesig
 ip address 21.10.4.1 255.255.255.0
!
interface GigabitEthernet0/0.805
 encapsulation dot1Q 805
 ip vrf forwarding VRFresource
 ip address 21.10.5.1 255.255.255.0
!
interface GigabitEthernet0/1
 ip vrf forwarding VRFvoicesig
 ip address 21.20.10.1 255.255.255.0
 duplex auto
 speed auto
!
router bgp 1000
 no synchronization
 bgp router-id 22.22.22.22
 bgp log-neighbor-changes
 no auto-summary
 !
 address-family ipv4 vrf VRFvoicesig
  redistribute connected
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf VRFvoice
  redistribute connected
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf VRFservice
  redistribute connected
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf VRFresource
  redistribute connected
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf VRFdata
  redistribute connected
  no synchronization
 exit-address-family
!
ip forward-protocol nd
!
!
ip http server
!
access-list 10 permit 21.10.10.0 0.0.0.255
access-list 21 permit 21.22.21.0 0.0.0.255
access-list 22 permit 21.22.22.0 0.0.0.255
!
!
!
!
control-plane
!
!
ccm-manager fax protocol cisco
!
mgcp fax t38 ecm
!
sccp local GigabitEthernet0/1
sccp ccm 21.20.10.11 identifier 2 version 6.0
sccp
!
sccp ccm group 2
 bind interface GigabitEthernet0/0.805
 associate ccm 2 priority 1
 associate profile 103 register CFB00175a378101
!
sccp ccm group 3
 bind interface GigabitEthernet0/0.804
 associate ccm 2 priority 1
 associate profile 101 register MTP00175a378101
 associate profile 105 register softmtp-3825
!
dspfarm profile 101 transcode
 codec g711ulaw
 codec g711alaw
 codec g729ar8
 codec g729abr8
 codec g729r8
 maximum sessions 20
 associate application SCCP
!
dspfarm profile 103 conference
 codec g711ulaw
 codec g711alaw
 codec g729ar8
 codec g729abr8
 codec g729r8
 codec g729br8
 maximum sessions 8
 associate application SCCP
!
dspfarm profile 105 mtp
 codec g711ulaw
 rsvp
 maximum sessions hardware 5
 maximum sessions software 250
 associate application SCCP
!
!
!
gateway
 timer receive-rtp 1200
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 exec-timeout 0 0
 login
!
exception data-corruption buffer truncate
scheduler allocate 20000 1000
end

Router1#
```

Verification

The following commands can be used to verify and debug the VRF configuration:

```
show ip route vrf <vrf-name>
 show ip route vrf VRFvoice
 show ip route vrf VRFdata
show ip vrf interface
 show ip vrf  detail
 ping vrf VRFvoice 1.1.1.1

 debug ip cef packet
 debug ip cef drop
 debug ip dhcp packet
 debug voip rtp sessions
 debug voip rtp packet (hidden and huge impact to performance)
```

Some of the outputs are given in the following section.

The following command shows the different interfaces assigned to different VRF resources.

```
router#show ip vrf interface
Interface     IP-Address      VRF                              Protocol
Gi0/0.801     21.22.21.1      VRFdata                          up
Gi0/0.805     21.10.5.1       VRFresource                      up
Gi0/0.803     21.10.3.1       VRFservice                       up
Gi0/0.802     21.22.22.1      VRFvoice                         up
Gi0/0.804     21.10.4.1       VRFvoicesig                      up
Gi0/1         21.20.10.1      VRFvoicesig                      up
router#
```

The following command shows the VRF routing table for *VRFvoice*.

```
router#sh ip route vrf VRFvoice
Routing Table: VRFvoice
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
     21.0.0.0/24 is subnetted, 4 subnets
B       21.20.10.0 is directly connected, 00:19:36, GigabitEthernet0/1
C       21.22.22.0 is directly connected, GigabitEthernet0/0.802
B       21.10.4.0 is directly connected, 00:19:36, GigabitEthernet0/0.804
B       21.10.3.0 is directly connected, 00:19:36, GigabitEthernet0/0.803
```

The following command shows the VRF routing table for *VRFdata*.

```
router#sh ip route vrf VRFdata
Routing Table: VRFdata
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
     21.0.0.0/24 is subnetted, 5 subnets
B       21.20.10.0 is directly connected, 5d23h, GigabitEthernet0/1
C       21.22.21.0 is directly connected, GigabitEthernet0/0.801
B       21.10.4.0 is directly connected, 1d01h, GigabitEthernet0/0.804
B       21.10.3.0 is directly connected, 1d01h, GigabitEthernet0/0.803
router#
```

The following command shows the routing table for the *VRFvoicesig* VRF, which is connected to Cisco Unified CM.

```
router#sh ip route vrf VRFvoicesig
Routing Table: VRFvoicesig
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
     21.0.0.0/24 is subnetted, 6 subnets
```

```
C       21.20.10.0 is directly connected, GigabitEthernet0/1
B       21.22.22.0 is directly connected, 1d01h, GigabitEthernet0/0.802
B       21.22.21.0 is directly connected, 1d01h, GigabitEthernet0/0.801
B       21.10.5.0 is directly connected, 1d01h, GigabitEthernet0/0.805
C       21.10.4.0 is directly connected, GigabitEthernet0/0.804
B       21.10.3.0 is directly connected, 1d01h, GigabitEthernet0/0.803
router#
```

The following command shows different SCCP resources (conferencing, transcoding, MTP, and so on) that are registered with Cisco Unified CM.

```
router#sh sccp all
SCCP Admin State: UP
Gateway Local Interface: GigabitEthernet0/1
        IPv4 Address: 21.20.10.10
        Port Number: 2000
IP Precedence: 5
User Masked Codec list: None
Call Manager: 21.20.10.12, Port Number: 2000
                Priority: N/A, Version: 5.0.1, Identifier: 1
                Trustpoint: N/A
Transcoding Oper State: ACTIVE - Cause Code: NONE
Active Call Manager: 21.20.10.12, Port Number: 2000
TCP Link Status: CONNECTED, Profile Identifier: 101
………
MTP Oper State: ACTIVE - Cause Code: NONE
Active Call Manager: 21.20.10.12, Port Number: 2000
TCP Link Status: CONNECTED, Profile Identifier: 105
………..
Conferencing Oper State: ACTIVE - Cause Code: NONE
Active Call Manager: 21.20.10.12, Port Number: 2000
TCP Link Status: CONNECTED, Profile Identifier: 103
…….
```

Caveats

These are some known limitations of the system:

- There is only single-VRF support for TDM GWs.
- MGCP is not supported.
- Multi-VRF supports only Cisco Unified CME and Cisco Unified CM DSP resources (conf/xcod/MTP) .
- Other components (Voice GW, CUBE) are single-VRF capable only.
- VRF-configuration per dial-peer is not supported.
- Connecting calls between different VRFs requires Cisco Unified CME flow-through mode, even for local SCCP-SCCP calls.
- There is no video support for VRF.
- Firewall traversal and VRF traversal are mutually exclusive.
- Not supported at the same time on the same platform.
- RSVP GW and RSVP-Agent are not VRF-aware yet.
- GateKeeper is not VRF-aware.

- If GateKeeper is co-resident with VRF-aware Voice GW or CUBE configurations, then they cannot communicate with each other