Configuring Ethernet over GRE

Ethernet over GRE (EoGRE), is a tunneling protocol that enables tunneling of Layer 2 packets encapsulated in GRE header over IP core networks. Generic Routing Encapsulation (GRE) is a tunneling protocol that encapsulates a wide variety of network layer protocols inside virtual point-to-point links over a Layer 3 IPv4 or Layer 3 IPv6 access network.

Prerequisites

The following are the prerequisites for configuring EoGRE:

- IP routing must be enabled. The following command enables IP routing:

  ip routing

- IP CEF must be enabled. The following command enables IP CEF:

  ip cef

- Sub-interfaces for VLANs must be created to tunnel Ethernet frames with the VLAN tag. The following commands create sub interfaces for VLANs:

  interface Dot11Radio *interface number.sub-interface number*

  encapsulation dot1Q *vlan id*

  bridge-group *bridge id*

  interface GigabitEthernet0.*sub-interface number*

  encapsulation dot1Q *vlan id*

  bridge-group *bridge id*

Note The bridge ID on interfaces with the same VLAN ID, must be the same.

The following are not supported:

- SNMP, and GUI through ACS configurations
- Tunnel establishment using IPv6 address

Configuring EoGRE

Configuring a tunnel profile defines configurable parameters to create a tunnel. The following parameters are to be configured under the dot11 tunnel:

- Tunnel address mode
- Source address
- Destination address
- Maximum segment size (MSS)
- Maximum transmission unit (MTU)
- Type of service (ToS) or Differentiated Services Code Point (DSCP)

Beginning in privileged EXEC mode, follow these steps to configure a tunnel profile under the dot11 tunnel.

Command	Purpose
mode [ipv4 | ipv6]	Set tunnel address mode to IPv4 or IPv6
source *address*	Source address, default is AP's BVI address
destination *address*	Tunnel destination address
mss *size*	Set TCP MSS value for incoming and outgoing TCP syn and syn/ack packets. Default size is 1360.
mtu *size*	Incoming IP packets will fragmented if the size of IP packet is larger than this value and then an ICMP Need Fragmentation error message is sent to the client. Default size is 1400.
tos *value*	To set a ToS or DSCP value in the transport IP address. Default value is zero (0).

Examples

```
ap(config)# dot11 tunnel sample
ap(config-dot11-tunnel)# mode ipv4
ap(config-dot11-tunnel)# destination 1.1.1.1
ap(config-dot11-tunnel)# mss 1360
ap(config-dot11-tunnel)# mtu 1400
ap(config-dot11-tunnel)# tos 5
ap(config-dot11-tunnel)# end
```

Mapping SSID to Tunnel

Mapping the tunnel to the WLAN is done by using the command **tunnel** *tunnel_profile* under the SSID configuration.

Beginning in privileged EXEC mode, follow these steps to map the SSID to the tunnel.

	Command	Purpose
Step 1	**dot11 ssid** *ssid*	Specifies the SSID
Step 2	**vlan** *vlan id*	Specifies the VLAN ID
Step 3	**tunnel** *tunnel profile*	Specifies the tunnel profile to be used
Step 4	**authentication {open | eap }**	Specifies the type of authentication

Examples

```
ap(config)# dot11 ssid doc
ap(config-ssid)# tunnel sample
ap(config-ssid)# authentication open
ap(config-ssid)# end
```

Configuring DHCP Snooping for EoGRE clients

DHCP snooping is a security feature that acts like a firewall between untrusted hosts and trusted DHCP servers. By enabling DHCP snooping on the AP, the AP inserts the relay agent information option (DHCP option 82) which contains two sub-options Circuit ID and Remote ID.

Note DHCP Snooping is disabled by default.

Beginning in privileged EXEC mode, follow these steps to enable DHCP snooping for EoGRE clients under dot11 SSID.

	Command	Purpose
Step 1	**dhcp-snoop enable**	Enables DHCP snooping. By default, DHCP snooping is disabled.
Step 2	**dhcp-snoop circuit_id format {ap-mac | client-mac | eth-mac | name | ssid | type | vlan | raw** *word_string***}**	Specify the format of the string sequence to used as the Circuit ID. To know the format to be specified, see Circuit ID and Remote ID Format and Strings, page 22-4. The Circuit ID gets inserted into the DHCP packets
Step 3	**dhcp-snoop circuit_id** *circuit-id-string_sequence*	Specify the string sequence to used as the Circuit ID, in the format you have set. Each string is separated from others using a character delimiter, the default being ';'

	Command	Purpose
Step 4	**dhcp-snoop remote_id format {ap-mac | client-mac | eth-mac | name | ssid | type | vlan | raw** *word_string***}**	You need to specify the format of the string sequence to used as the Remote ID. To know the values to be specified, see Circuit ID and Remote ID Format and Strings, page 22-4.
Step 5	**dhcp-snoop remote_id** *remote-id-string_sequence*	You need to specify the string sequence to used as the Remote ID, in the format you have set. Each string is separated from others using a character delimiter, the default being ';'

Examples

```
ap(config)# dot11 ssi
ap(config)# dot11 ssid doc
ap(config-ssid)# dhcp-snoop enable
ap(config-ssid)# dhcp-snoop circuit_id format ap-mac ssid type
ap(config-ssid)# dhcp-snoop circuit_id 00:10:A4:23:B6:C0;xfinityWiFi;s
ap(config-ssid)# dhcp-snoop remote_id format client-mac
ap(config-ssid)# dhcp-snoop remote_id 00:50:24:23:B7:D0
ap(config-ssid)# end
```

Additional Commands

The default DHCP Snooping encoding is in binary. You can set it to ASCII using the following command:

ap(config-ssid)# **dhcp-snoop encoding ascii**

The default DHCP Snooping string sequence delimiter is the single character ';'. To change this, use the following command:

ap(config-ssid)# **dhcp-snoop delimiter** *single_character_or_string*

The *single_character_or_string* can be up to 127 characters long.

Circuit ID and Remote ID Format and Strings

For both the Circuit ID and the Remote ID, you need to specify the format of the string sequence for each, before you assign the string for each.

The format and strings can be a combination of up to five out of eight values shown in the following table. When specifying the string sequence, the strings are separated by the delimiter character, the default being ';'.

Format	Nature of corresponding string
ap-mac	AP radio MAC address
client-mac	Client MAC address
eth-mac	AP Ethernet MAC address
name	AP name
raw *word_string*	Any string. If raw is specified in the format command, then the string to be entered is also specified alongside.

Format	Nature of corresponding string
ssid	Service Set Identifier (SSID)
type	Type of SSID. it is ‘o’ for Open SSID and ‘s’ for Secure SSID
vlan	VLAN name

Configuring Redundancy for Tunnel Gateway Address

Configuring a redundancy for the tunnel helps you to switchover from primary to secondary when the working gateway address fails or becomes unreachable.

The following parameters are to be configured under dot11 tunnel to configure redundancy:

- Backup destination
- Backup timeout
- Keep alive parameters

Beginning in privileged EXEC mode, follow these steps to configure redundancy address for the tunnel:

	Command	Purpose
Step 1	**Backup destination** *address*	Specifies the backup destination address
Step 2	**Backup timeout** *seconds*	Specifies the number of seconds after which the tunnel switches from backup to primary
Step 3	**Keepalive** *count interval dead-count timeout*	The *count* is the number of ping packets sent every *interval* seconds. After the *dead-count* pings fail, a tunnel endpoint is assumed to be dead. The *timeout* is the number of seconds the AP waits for ping replies after sending a ping. Default values for *count*, *interval*, *dead-count*, and *timeout* is 3, 60, 3, and 1 respectively.

Note During the switchover from primary to secondary, or vice versa, all associated clients will be deauthenticated and will reassociate after the switchover.
When both the primary and secondary are down, the SSIDs that are attached to the tunnel will also be down. Once either of the primary or secondary address can be reached by the AP, the SSID will come up and start serving clients.

Examples

```
ap(config)# dot11 tunnel sample
ap(config-dot11-tunnel)# backup destination 2.2.2.2
ap(config-dot11-tunnel)# backup timeout 60
ap(config-dot11-tunnel)# keepalive 3 60 3 3
ap(config-dot11-tunnel)# end
```