APPENDIX C

Terminating an Access Ring on Two N-PEs

This appendix describes how to terminate an access ring on two N-PEs for redundancy in case an access link goes down. It contains the following sections:

Overview

Prime Provisioning supports device-level redundancy in the service topology. This allows the service to remain active in case one access link should drop. This is accomplished through support for provisioning termination of access links against two different N-PEs. This is implemented by allowing an access ring to terminate on two different N-PEs. This may also be described as a "dual-homed access ring." The N-PEs are connected by a logical link using loopback interfaces on the N-PEs. The redundant link starts from a U-PE device and may, optionally, include PE-AGG devices. One attachment link is primary and one is secondary. The selection is made when the Named Physical Circuit (NPC) is created. The terminating device on the NPC acts as the primary N-PE, while the other N-PE on the same ring acts as the secondary N-PE.

For backward compatibility, Prime Provisioning continues to support provisioning services without redundant links, as in previous releases.

N-PE redundancy is supported for FlexUNI/EVC and MPLS services. As many of the basic concepts are shared for both services, both are covered in this appendix.

Figure C-1 and Figure C-2 show two network topologies which illustrate redundancy, starting from a U-PE access node. Both topologies provide open segments for each uplink, starting from the U-PE and terminating on the N-PE devices. The N-PEs are logically connected via loopback interfaces. Services are configured on both of these Ethernet access links starting from the U-PE to two different N-PEs.

Figure C-1 N-PE Redundancy, Starting at the U-PE

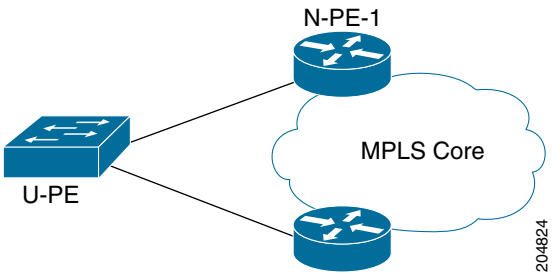
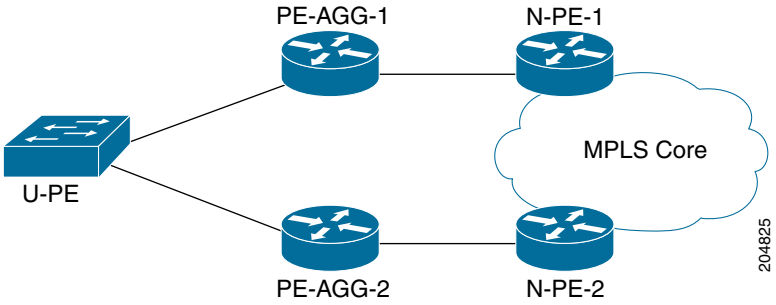


Figure C-2 N-PE and PE-AGG Redundancy, Starting at the U-PE



The first topology (N-PE redundancy starting at the U-PE, as shown in Figure C-1) provides the model of fault recovery for the N-PE device. As shown in the diagram, there are two different outgoing interfaces starting from the U-PE device. Each terminates at a different N-PE.

The second topology (N-PE and PE-AGG redundancy starting at the U-PE, as shown in Figure C-2) provides fault recovery for both the PE-AGG and N-PE devices. The service switches over from the primary to the secondary link when either the PE-AGG or the N-PE of the primary link fails.

For other network scenarios illustrating more complex topologies, see Additional Network Configurations and Sample Configlets, page C-5.

The following list provides additional details about the implementation:

- Using one U-PE and two N-PEs consumes one access link (AL).
- When creating a service on a U-PE, the user specifies an NPC to be used. If the topology includes an access ring with two N-PEs, then the service is configured on both N-PEs.
- For Ethernet over MPLS (EoMPLS) pseudowire (PW) services, if there is N-PE redundancy on both sides of the service provider network, two pseudowires are created. One N-PE is defined as primary and the other as secondary, in order to determine the how the pseudowires connect. If the user enables the PW Redundancy option, the primary and secondary on either end are also connected with pseudowire redundancy.
- For point-to-point (P2P) configurations, the two N-PEs use two separate pseudowires.
- Prime Provisioning supports the case in which the service is configured identically (except for the access interface) on both N-PEs. This saves the user from having to enter data twice because the link attributes in the service request workflow are common for both N-PEs that are part of the attachment circuit.
- This feature is supported for both Cisco 7600 and Cisco ASR 9000 platforms. However, a single service cannot include both 7600 and ASR 9000 platforms.
- For the Cisco ASR 9000 platform, IOS XR version 3.7.3 and 3.9.0 are supported.

Note Check the on-line version of *Cisco Prime Provisioning 6.3 Release Notes,* for the most current information on device and platform support, in case updates have occurred since the publication of this guide.

The implementation of this feature is covered in more detail in the following sections.

Setting Up an NPC Access Ring with Two N-PEs

Terminating an NPC access ring on two N-PEs is achieved by using the standard method of setting up an NPC ring in Prime Provisioning. The basic steps for doing this are described Setting Up Logical Inventory, page 2-53. Additional information is provided in this guide in the section Creating Named Physical Circuits, page 3-12.

In normal cases, a ring would be closed by connecting the devices via physical interfaces. When terminating an access ring on two different N-PEs, there is no need for a physical connection between the N-PEs. However, Prime Provisioning requires that a virtual link must be created between the N-PEs, in order to close the ring. The virtual link is set up through the use of loopback interfaces.

In order to use loopback interfaces in a ring in this manner, you must enable the DCPL property allowLoopbackIntfInNPC, which is accessed in the Host Configuration window under the folder /repository/mlshare. When this DCPL property is set to true, Prime Provisioning allows the use of loopback interfaces in a ring.

Note Note that Prime Provisioning does not generate any configlets onto the loopback interfaces during deployment of the service request.

Using N-PE Redundancy in FlexUNI/EVC Service Requests

Using a dual-homed access ring in a FlexUNI/EVC service request does not require any change in the usual workflow in the Prime Provisioning GUI. During creation of the FlexUNI/EVC service request, you select the NPC which is associated with an NPC access ring terminating on two N-PEs.

Usage notes:

- The service is configured on both N-PEs of the access ring.
- Though there are two different N-PEs, only one access link is consumed.
- You can modify the configuration redundant N-PEs before or after deploying the service request. Modified configlets will be generated according to the changes made in service request.
- The destined N-PE device on the NPC used in the service request is treated as the primary N-PE. The other N-PE on the same ring is treated as the secondary N-PE. To change the primary and secondary N-PE, you must modify the attachment circuits in the service request.
- Configlets are generated according to the configuration specified in the service request. Prime Provisioning generates identical configlets on both of the N-PEs in the attachment circuit (AC). The Link Attributes sections are common for both N-PEs.
- For FlexUNI/EVC services, N-PE redundancy is supported for PSEUDOWIRE and VPLS core connectivity types.

- In case of VPLS core connectivity, all N-PEs in NPC rings are configured to have Layer 2 Virtual Forwarding Interface (VFI), and all N-PEs on the same VPLS VPN participate in the VPLS service at the same time.
- In the case of PSEUDOWIRE core connectivity, the following notes apply:
  - If there is N-PE redundancy on both sides, a point-to-point pseudo wire (PW) will be configured between the N-PEs that were specified as the terminating N-PE devices during the NPC creation (between primary N-PEs). One more point-to-point PW will be configured between the N-PEs that were not specified as the terminating N-PE devices during NPC creation. The VC IDs of these pseudowires will be common.
  - If there is N-PE redundancy on only one side, then the Pseudowire Redundancy option must be checked in the GUI (in the Service Request Details section of the of the FlexUNI(EVC) Service Editor window). The primary PW will connect the primary N-PE of the dual-homed ring with the N-PE of the single-homed ring, and the secondary PW will connect the secondary N-PE of the dual-homed ring with the N-PE of the single-homed ring. Prime Provisioning will issue a warning message if you try to save the service request without enabling the Pseudowire Redundancy option.

Using N-PE Redundancy in MPLS Service Requests

Access ring termination on two N-PEs is supported for MPLS/L3 services for the Regular PE-CE policy type. The process of creating the NPC rings and associating them into the MPLS service is similar to that covered in Using N-PE Redundancy in FlexUNI/EVC Service Requests, page C-3. There are not any changes to the standard MPLS service request workflow.

Usage notes:

- The service is configured on both N-PEs of the access ring in the PE_NO_PE case. However, in the PE_CE case, the service request is configured on the primary N-PE of the access ring.
- Though there are two different N-PEs, only one access link is consumed.
- You can modify the configuration-redundant N-PEs before or after deploying the service request. Modified configlets will be generated according to the changes made in the service request.
- The destined N-PE device on the NPC used in the service request is treated as the primary N-PE. The other N-PE on the same ring is treated as the secondary N-PE.
- To change the primary N-PE, delete and recreate the NPC, provided the NPC is not associated with any service requests. To change the secondary N-PE, you have to modify the secondary N-PE at the ring level.
- During MPLS service request creation using the PE_NO_CE policy, the secondary NPE device can be configured through the second link. Separate link attributes such as VLAN ID, PE Interface
- Address/Mask, VPN and RD and others can be configured separately for both primary and secondary N-PEs. This way you can manually add a different IP address on primary and secondary N-PEs. UNI device information will be available only in the link of the primary N-PE.
- During MPLS service request creation using the PE-CE policy, only one MPLS VPN link would be created even though the selected NPC has two N-PEs. Service can be associated only to the primary N-PE, no additional link will be provided for the secondary N-PE. Configlets will be generated and pushed to all the devices in the ring except the secondary N-PE.
- VPNs and VRF objects are supported for MPLS service requests using access ring termination on two N-PEs.

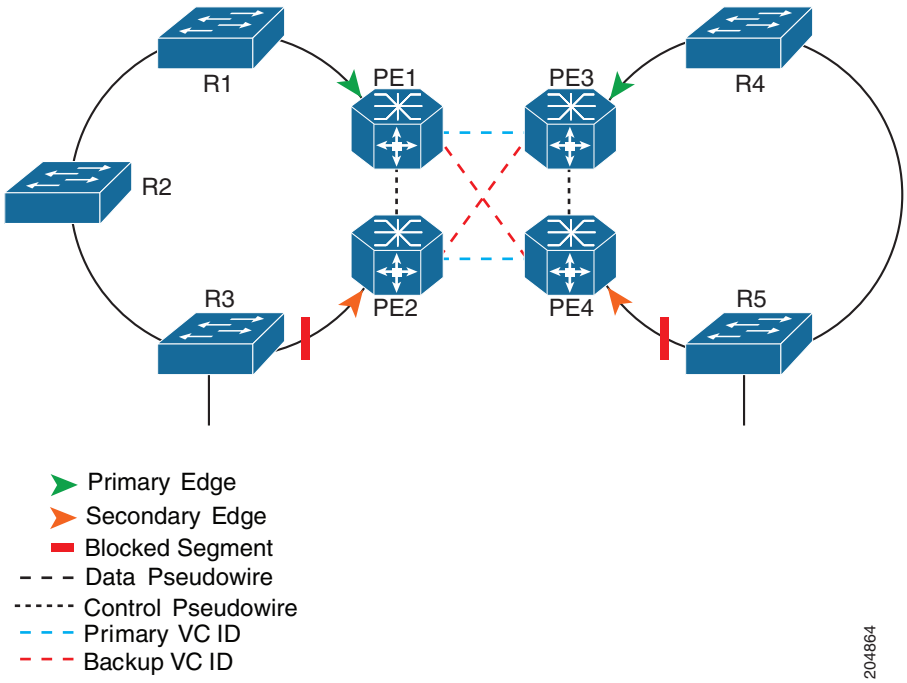
Additional Network Configurations and Sample Configlets

This section provides additional network scenarios for reference, along with sample configlets for associated network devices.

Example 1: Pseudowire Connectivity (A)

Figure C-3 illustrates a network configuration with pseudowire connectivity with dual-homed N-PEs on both sides of the network and with pseudowire redundancy.

Figure C-3 Pseudowire Connectivity, Dual-Homed N-PEs on Both Sides of the Network, with Pseudowire Redundancy



Sample configlets for the devices are provided below.

PE1

```
vlan <S-Vlan>
!
interface <UNI-to-R1>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE3 loopback> <PrimaryVcId> encapsulation mpls
    backup peer <PE4 loopback> <BackupVcId>
```

PE2

```
vlan <S-Vlan>
!
interface <UNI-to-R3>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE4 loopback> <PrimaryVcId> encapsulation mpls
    backup peer <PE3 loopback> <BackupVcId>
```

PE3

```
vlan <S-Vlan>
!
interface <UNI-to-R4>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE1 loopback> <PrimaryVcId> encapsulation mpls
    backup peer <PE2 loopback> <BackupVcId>
```

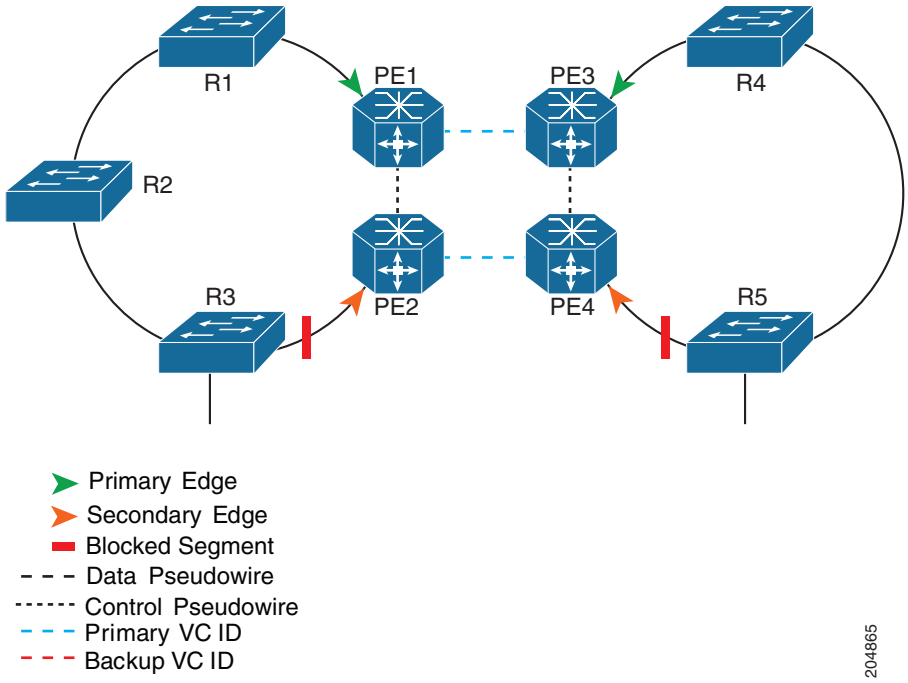
PE4

```
vlan <S-Vlan>
!
interface <UNI-to-R5>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE2 loopback> <PrimaryVcId> encapsulation mpls
    backup peer <PE1 loopback> <BackupVcId>
```

Example 2: Pseudowire Connectivity (B)

Figure C-4 illustrates a network configuration using pseudowire connectivity, with dual-homed N-PEs on both sides of the network without pseudowire redundancy.

Figure C-4 Pseudowire Connectivity, Dual-Homed N-PEs on Both Sides of the Network, with No Pseudowire Redundancy



Sample configlets for the devices are provided below.

PE1

```
vlan <S-Vlan>
!
interface <UNI-to-R1>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE3 loopback> <PrimaryVcId> encapsulation mpls
```

PE2

```
vlan <S-Vlan>
!
interface <UNI-to-R3>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE4 loopback> <PrimaryVcId> encapsulation mpls
```

PE3

```
vlan <S-Vlan>
!
interface <UNI-to-R4>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE1 loopback> <PrimaryVcId> encapsulation mpls
```

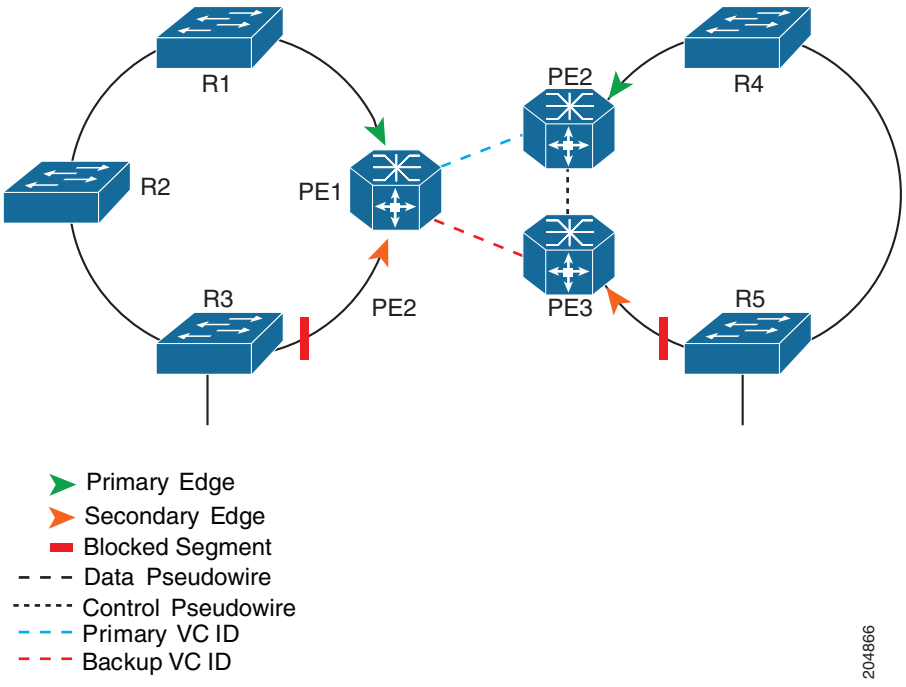
PE4

```
vlan <S-Vlan>
!
interface <UNI-to-R5>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE2 loopback> <PrimaryVcId> encapsulation mpls
```

Example 3: Pseudowire Connectivity (C)

Figure C-5 illustrates a network configuration using pseudowire connectivity with dual-homed N-PEs at one side of the network and with pseudowire redundancy.

Figure C-5 Pseudowire Connectivity, Dual-Homed N-PEs on One Side of the Network, with Pseudowire Redundancy



Sample configlets for the devices are provided below.

PE1

```
vlan <S-Vlan>
!
interface <UNI-to-R1>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE2 loopback> <PrimaryVcId> encapsulation mpls
    backup peer <PE3 loopback> <BackupVcId>
```

PE2

```
vlan <S-Vlan>
!
interface <UNI-to-R4>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE1 loopback> <PrimaryVcId> encapsulation mpls
```

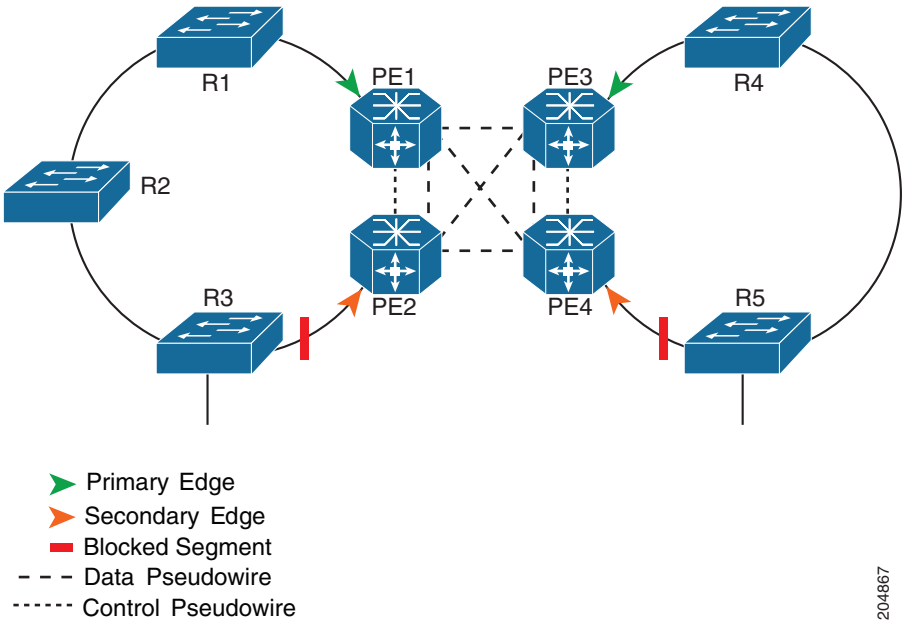
PE3

```
vlan <S-Vlan>
!
interface <UNI-to-R5>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE1 loopback> <BackupVcId> encapsulation mpls
```

Example 4: VPLS Connectivity

Figure C-6 illustrates a network configuration using VPLS connectivity with dual-homed N-PEs on both sides of the network.

Figure C-6 VPLS Connectivity, Dual-Homed N-PEs on Both Sides of the Network



Sample configlets for the devices are provided below.

PE1

```
vlan <S-Vlan>
!
l2 vfi <VFI-ID> manual
    vpn id <S-Vlan>
    neighbor <PE2> encapsulation mpls
    neighbor <PE3> encapsulation mpls
    neighbor <PE4> encapsulation mpls
!
interface vlan <S-Vlan>
    xconnect vfi <VFI-ID>
!
interface <NNI-to-R1>
    switchport trunk allowed vlan add <S-Vlan>
```

PE2

```
vlan <S-Vlan>
!
l2 vfi <VFI-ID> manual
    vpn id <S-Vlan>
    neighbor <PE1> encapsulation mpls
    neighbor <PE3> encapsulation mpls
    neighbor <PE4> encapsulation mpls
!
interface vlan <S-Vlan>
    xconnect vfi <VFI-ID>
!
interface <NNI-to-R3>
    switchport trunk allowed vlan add <S-Vlan>
```

PE3

```
vlan <S-Vlan>
!
l2 vfi <VFI-ID> manual
    vpn id <S-Vlan>
    neighbor <PE1> encapsulation mpls
    neighbor <PE2> encapsulation mpls
    neighbor <PE4> encapsulation mpls
!
interface vlan <S-Vlan>
    xconnect vfi <VFI-ID>
!
interface <NNI-to-R5>
    switchport trunk allowed vlan add <S-Vlan>
```

PE4

```
vlan <S-Vlan>
!
l2 vfi <VFI-ID> manual
    vpn id <S-Vlan>
    neighbor <PE1> encapsulation mpls
    neighbor <PE2> encapsulation mpls
    neighbor <PE3> encapsulation mpls
!
interface vlan <S-Vlan>
    xconnect vfi <VFI-ID>
!
interface <NNI-to-R4>
    switchport trunk allowed vlan add <S-Vlan>
```