Service Configuration Procedures

This chapter is meant to be used in conjunction with the previous chapters that provide examples for configuring the system to support Simple IP services, Mobile IP services, or both. It provides procedures for configuring the various elements to support these services.

It is recommended that you first select the configuration example that best meets your service model, and then use the procedures in this chapter to configure the required elements for that model.

Important This manual is valid for configuring PDSN on multiple platforms. Consequently not all sections, descriptions, features and commands are supported on all platforms. Others are activated by license only.

Important For hardware supporting them, at least onepacket processing card must be made active prior to service configuration. Information and instructions for configuring active cards can be found in the "Configuring System Settings" chapter of the *System Administration Guide*.

The following topics are included:

Configuring PDSN Closed R-P Services

PDSN Closed R-P services are configured within contexts and allow the system to function as a PDSN in 3G wireless data networks that implement the proprietary Closed R-P protocol from Nortel Networks®.

Important A valid feature license must be installed prior to configuring PDSN Closed R-P services. This section provides the minimum instruction set for configuring a Closed R-P service that allows the system to process data sessions. Commands that configure additional Closed R-P service properties are provided in the *Command Line Interface Reference*.

Use this example to configure PDSN Closed R-P services:

```
configure
 context <name>
 pdsnclosedrp-service <name>
peer-pcf <ipv4/ipv6_address> | <ipv4_address/ipv6_address/mask>
 bind address <interface_address>
 authentication
 nai-construction domain <domain_name>
 end
```

- Optionally configure the accounting start/stop RRQ generation during intra-PDSN handoff, by entering the **accounting peer-pcf handoff** command. If disabled, accounting start/stop RRQs are not generated for the Closed R-P to Closed R-P handoff between two PCFs, which are within the same PCF address range as configured in **peer-pcf** configuration. By default it is enabled.
- Repeat this example as needed to create and bind additional PDSN Closed R-P services to any other interfaces.
- Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command **save configuration**. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

Following are a few items to be aware of:

- Since a Closed-RP service uses the R-P interface as an ingress (arriving) interface to receive incoming sessions and session requests from PCFs, this service type must reside in a source context.
- Up to 10000 peer PCFs can be configured for each Closed R-P Service.

Important The Closed R-P service must not be bound to the same ip address of an interface as other L2TP LAC or LNS services.

Verifying your Configuration

Use the following command to verify the Closed R-P configuration:

show pdsnclosedrp-service name service_name

The output of this command should be similar to the output shown in the following example.

```
Service name: closedrp-1
  Context:                      pdsn
  Bind:                         Done
  Local IP Address:             192.168.2.20
  First Retransmission Timeout: 1 (secs)
  Max Retransmission Timeout:   8 (secs)
  Max Retransmissions:          5
  Max Sessions:                 500000        Max Tunnels: 10000
  Max Sessions Per Tunnel:      65535
  Keep-alive interval:          60            Control receive window: 16
  Data Sequence Numbers:        Disabled
  PPP Authentication:           CHAP 1 PAP 2
  Allow Noauthentication:       Disabled      MSID Authentication:    Disabled
  No Default Subscriber defined
```

```
PPP Tunnel Type:                None
PPP Tunnel Context:             <NONE DEFINED>
MIP FA Context:                 <NONE DEFINED>
Service Status:                 Started
Newcall Policy:                 None
```

Creating and Configuring PDSN Services

PDSN services are configured within contexts and allow the system to function as a PDSN in the 3G wireless data network.

Important This section provides the minimum instruction set for configuring a PDSN service that allows the system to process data sessions. Commands that configure additional PDSN service properties are provided in the *Command Line Interface Reference*.

Use this example to configure PDSN services:

```
configure
 context <name>
 pdsn-service <name>
 ip local-port <port>
 authentication allow-noauth
 authentication chap 1 mschap 2 pap 3 allow-noauth
 nai-construct domain <alias>
 spi remote-address <pcf_ipv4_address/pcf_ipv6_address/mask> spi-number <number> {
secret <secret> }
 lifetime <time>
 gre protocol-type { any | byte-stream | ppp }
 bind address address
 exit
 ppp lcp-start-delay <seconds>
 no ppp renegotiation retain-ip-address
 end
```

Notes:

- Optional: If you are implementing Mobile IP data services, configure the name of the context in which the FA service is configured by entering the **mobile-ip foreign-agent context fa_context_name [fa-service <name>]** command.
- Optionally configure the PDSN service to monitor all PCFs that it is associated with, enter the **pcf-monitor** command.
- Optionally configure the PDSN behavior for A11 RRQ related parameters. **airlink bad-sequence-number deny** can be used to deny A11 RRQ messages that have an unsupported Vendor Id or invalid Airlink Sequence number (less than or equal to a previously received sequence number). Keywords and options that configure additional PDSN service behavior for A11 RRQs with this command are provided in the *Command Line Interface Reference*.
- Optionally use the **no dormant-transition initial-session-setup** command to configure the PDSN behavior to terminate A10 session, when the PDSN receives the A11-RRQ (Type 4) before the session for the original MN is established completely.

- Optionally use the **no pcf-session-id-change restart-ppp** command to configure the PDSN behavior to disable the ppp renegotiation, when the PDSN receives the A11 RRQ (Type 4) with a change in GRE key or PCF session Id, from current PCF and no change in PCF/PANID/CANID.
- Optionally use the **setup-timeout**<*seconds*> command to change the maximum amount of time, in seconds, allowed to set up a session. The default setting is 60 seconds.
- Optionally configure a delay before starting LCP to avoid the first LCP Configuration Request being lost because the RP link may not be ready even if it has indicated it is active. Losing an LCP Config Request increases the total session setup time.
- Optional: You can configure the system whether to retain the currently allocated IP address for the session or to release the current IP address, and a new IP address is to allocate after PPP renegotiation.
- To retain the allocated IP during PPP renegotiation use the **[default] ppp renegotiation retain-ip-address** command

Important By default it will use the same IP address, allocated during renegotiation, after renegotiation also. Detailed informations are provided in the *Command Line Interface Reference*.

- Optionally configure the MSID length to reject the A11-RRQs with illegal IMSI value by entering the **[default] msid length [min** *min_length* **] max** *max_length* command:

  By default it will use the default MSID length as per standard. Detailed informations are provided in the *Command Line Interface Reference*.
- The nai-construct domain command should only be used if the PDSN service is configured to allow no authentication using the authentication allow-noauth command.
- Multiple SPIs can be configured within the PDSN service in order to accommodate a single PDSN interface communicating with multiple PCFs.
- An infinite lifetime can be configured using the no lifetime command.
- Multiple addresses on the same IP interface can be bound to different PDSN services. However, each address can be bound to only one PDSN service.
- The hardware configuration and features installed can affect the maximum subscriber sessions that can be supported.
- Repeat this configuration as needed to create and bind additional PDSN services to any other interfaces.
- Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command **save configuration**. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

Verifying the PDSN Services

Step 1 Use the following command to verify that the PDSN service was created and configured properly:

```
show pdsn-service { name service_name | all }
```

The output is a concise listing of PDSN service parameter settings as shown in the sample output below. In this example, a PDSN service called pdsn1 was configured.

```
Service name: pdsn1
  Context: test1
  Bind:                 Not Done
  Local IP Address:     0.0.0.0          Local IP Port:          699
  Lifetime:             00h30m00s        Retransmission Timeout: 3 (secs)
  Max Retransmissions:  5                Setup Timeout         : 60 (secs)
  No MIP FA Context defined
  No NAI construct domain defined
  GRE Sequence Numbers: Enabled          GRE Protocol Type:      Any
  GRE Reorder Timeout:  100 msec         GRE Sequence Mode:      None
  GRE Checksum: Disabled                 GRE Checksum Verification: Disabled
  Enable Data Available Indicator:  Yes   Inter-PDSN handoffs have MEI:  No
  Reg discard on bad extension:     No    Reg discard on GRE key change: No
  Reg ack deny terminates session:  No    Reg update wait timeout:       No
  Deny newcall if no rev. tunnel:   No
  Terminate session on R-P errors:  No    Max retried replies on reg deny: 3
  Deny using zero GRE key:          No    Deny if session already closed:  No
  Deny if session already dormant:  No    Deny if session already active:  No
  Deny if CoA & src addr mismatch:  No
  Deny newcall if no conn setup:    No   (Deny code: Reason Unspecified)
  RRQ with bad airlink seq num:     Accept(Deny code: Poorly Formed Request)
  Deny if CRP to RP H/O in progress:No
  Handoff with no conn setup:       Accept
  Accept H/O if sess being disc:    No
  PPP Authentication:  CHAP 1 PAP 2
  Allow Noauthentication:  Disabled       MSID Authentication:    Disabled
  Fragment PPP Data:        Enabled
  GRE Flow Control:        Disabled
  GRE Flow Control Timeout: 10000 msec
  GRE Flow Control Timeout Action: disconnect-session
  Max sessions:            500000
  Alt-PPP:            Disabled
  PPP Tunnel Type:         None           No PPP Tunnel Context defined
  No Default Subscriber defined
  IP SRC-Violation Reneg Limit: 5         IP SRC-Violation Drop Limit:  10
  IP SRC-Violation Clear-on-ValidPDU: No  IP SRC-Violation Period:      120 secs
  Always-On-Indication: Disabled          SDB Indication for Echo Req: Disabled
  SPI(s):
  Service Status:          Not started
  Overload Policy:         Reject (Reject code: Admin Prohibited)
  Newcall Policy:          None
  Service Option Policy:   Enforce
  Service Options:         7,15,22,23,24,25,33,59
  PCF Monitor Config:      Disabled
```

Step 2 Verify configuration for errors by entering the following command:

`show configuration errors section pdsn-service verbose | more`

Creating and Configuring FA Services

FA services are configured within contexts and allow the system to function as an FA in the 3G wireless data network.

Important This section provides the minimum instruction set for configuring an FA service that allows the system to process data sessions. Commands that configure additional FA service properties are provided in the *Command Line Interface Reference*. Additionally, when configuring Mobile IP take into account the MIP timing considerations discussed in the Appendix *MIP Timer Considerations*.

Use this example to create and/or configure FA services:

```
configure
 context <name>
 fa-service <name>
 ip local-port <port>
 fa-ha-spi remote-address <ipv4_address/ipv6_address>|<ipv4/ipv6_address/mask
spi-number number
{ encrypted secret secret | secret secret }
 advertise adv-lifetime <time>
 advertise num-adv-sent <number>
 advertise reg-lifetime <reg_time>
 multiple-reg <number>
 authentication mn-aaa { always | ignore-after-handoff | init-reg |
init-reg-except-handoff | renew-and-dereg-noauth | renew-reg-noauth }
 reg-timeout time
 bind address ipv4_address max-subscribers max
 end
```

Following are a few things to be aware of:

- The **ip local-port** command configures the User Datagram Protocol (UDP) port for the Pi interfaces\' IP socket.
- A maximum of 2048 FA-HA SPIs can be configured for a single FA service.
- The agent advertisement lifetime is the amount of time that an FA agent advertisement remains valid in the absence of further advertisements.
- An infinite registration lifetime can be configured using the **no advertise reg-lifetime** command.
- The system only supports multiple Mobile IP sessions per subscriber if the subscriber\'s mobile node has a static IP address. The system only allows a single Mobile IP session for mobile nodes that receive a dynamically assigned home IP address.The hardware configuration and features installed can affect the maximum subscriber sessions that can be supported.
- Optionally configure the FA service for controlling the negotiation and sending of the I-bit in revocation messages by adding the **revocation negotiate-i-bit** comand. By default, it will not send I-bit in revocation message.
- Repeat the configuration as needed to create and bind additional FA services to any other interfaces.

Verifying the FA Service

Step 1 Verify that your FA services were created and configured properly by entering the following command:

show fa-service { name service_name | all }

The output is a concise listing of FA service parameter settings similar the sample displayed below. In this example, a FA service called fa1 was configured.

```
Service name:       fa1
  Context:          xxx
    Bind:             Done               Max Subscribers:      500000
  Local IP Address: 195.20.20.3         Local IP Port:        434
  Lifetime:         00h10m00s           Registration Timeout: 45 (secs)
  Advt Lifetime:    02h30m00s           Advt Interval:        5000 (msecs)
  Num Advt:         5
  Advt Prefix Length Extn: NO
  Reverse Tunnel:   Enabled             GRE Encapsulation:    Enabled
  Optimize Tunnel Reassembly: Disabled  Allow Priv Addr w/o Rev Tunnel: Disabled
  Dynamic MIP Key Update:     Enabled   Ignore Dynamic MIP Key: Disabled
  Remove MN-AAA/MN-FAC extns: Disabled
  Proxy MIP:                  Enabled   Proxy MIP Max Retransmissions: 5
  Proxy MIP Retrans Timeout:  3 (secs)  Proxy MIP Renew Percent Time:  75
  SPI(s):
    FAHA: Remote Addr: 195.30.30.3/32
    Hash Algorithm:    HMAC_MD5         SPI Num:  1000
    Replay Protection: Timestamp        Timestamp Tolerance: 60
  FAHA: Remote Addr: 195.30.30.2/32
    Hash Algorithm:    HMAC_MD5         SPI Num:  1000
    Replay Protection: Timestamp        Timestamp Tolerance: 60
  FAHA: Remote Addr: 195.30.30.1/32
    Hash Algorithm:    HMAC_MD5         SPI Num:  1000
    Replay Protection: Timestamp        Timestamp Tolerance: 60
  FAHA: Remote Addr: 195.20.20.4/32
    Hash Algorithm:    HMAC_MD5         SPI Num:  1000
    Replay Protection: Timestamp        Timestamp Tolerance: 60
  IPSEC Crypto Map(s):
    Peer HA Addr:       195.30.30.2
      Crypto Map:         test
  GRE Sequence Numbers:       Disabled  GRE Sequence Mode:    None
  GRE Reorder Timeout:        100 msec
  GRE Checksum:               Disabled  GRE Checksum Verification: Disabled
    Registration Revocation:    Enabled   Reg-Revocation I bit:   Enabled
  Reg-Revocation Max Retries: 3         Reg-Revocation Timeout: 3 (secs)
  Reg-Rev on InternalFailure: Enabled
  Default Subscriber:     None
  Max sessions:           500000
  Max challenge len:      16
  Challenge Window:       2
  Service Status:         Started
  MN-AAA Auth Policy:     Always
  MN-HA Auth Policy:      Always
  Newcall Policy:         None
  Idle Timeout Mode:      Normal
  Ignore Stale Challenge: Disabled
```

Step 2 Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command **save configuration**. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

Creating and Configuring HA Services

HA services are configured within contexts and allow the system to function as an HA in the 3G wireless data network.

Note This section provides the minimum instruction set for configuring an HA service that allows the system to process data sessions. Commands that configure additional HA service properties are provided in the *Command Line Interface Reference*. Additionally, when configuring Mobile IP take into account the MIP timing considerations discussed in *MIP Timer Considerations*.

Use this example to create and/or configure HA services:

```
configure
 context <name>
          ha-service <name>
 ip local-port <port>
 authentication mn-aaa { allow-noauth | always | noauth | renew-reg-noauth
 }
 fa-ha-spi remote-address <ipv4/ipv6_address > | <ipv4/ipv6_address/mask> spi-number
 <number> { [encrypted] secret <secret> }
 mn-ha-spi spi-number <number> { encrypted secret <secret> | secret
<secret> }
 reg-lifetime <time>
 simultaneous-bindings <number>
 bind address <ipv4_address> max-subscribers <max>
 end
```

Following are a few things to be aware of:

- The **ip local-port** command configures the User Datagram Protocol (UDP) port for the Pi interfaces\' IP socket.
- A maximum of 2048 FA-HA SPIs can be configured for each HA service.
- An infinite registration lifetime can be configured using the **no reg-lifetime** command.
- The hardware configuration and features installed can affect the maximum subscriber sessions that can be supported.
- Optionally configure the HA service for controlling the negotiation and sending of the I-bit in revocation messages by adding the **revocation negotiate-i-bit** comand. By default it will not send I-bit in recocation message.
- Optionally change the maximum amount of time, in seconds, allowed to set up a session. The default setting is 60 seconds. To change this value add the **setup-timeout** *seconds* command.
- Repeat the configuration as needed to create and bind additional HA services to any other interfaces.

Verifying the HA Service

Step 1 Verify that your HA services were created and configured properly by entering the following command:

show ha-service { name service_name | all }

The output is a concise listing of HA service parameter settings. In this example, a HA service called ha1 was configured.

```
Service name:  ha1
  Context: ha
  Bind:                Done              Max Subscribers: 500000
  Local IP Address:    192.168.4.10      Local IP Port:   434
  Lifetime:            00h10m00s         Simul Bindings:  3
  Reverse Tunnel:      Enabled           GRE Encapsulation:   Enabled
  Optimize Tunnel Reassembly: Enabled    Setup Timeout:   60 sec
SPI(s):
  MNHA: Remote Addr: 0.0.0.0
    Hash Algorithm:    MD5               SPI Num:  1000
    Replay Protection: Timestamp         Timestamp Tolerance: 60
    Permit Any Hash Algorithm:    Disabled
  FAHA: Remote Addr: 195.20.20.6/32
    Hash Algorithm:    HMAC_MD5          SPI Num:  1000
    Replay Protection: Timestamp         Timestamp Tolerance: 60
  FAHA: Remote Addr: 195.20.20.5/32
    Hash Algorithm:    HMAC_MD5          SPI Num:  1000
    Replay Protection: Timestamp         Timestamp Tolerance: 60
  FAHA: Remote Addr: 195.20.20.3/32
    Hash Algorithm:    HMAC_MD5          SPI Num:  1000
    Replay Protection: Timestamp         Timestamp Tolerance: 60
  FAHA: Remote Addr: 195.20.20.2/32
    Hash Algorithm:    HMAC_MD5          SPI Num:  1000
    Replay Protection: Timestamp         Timestamp Tolerance: 60
IPSEC Crypto Map(s):
    Peer FA Addr:       192.168.4.1
      Crypto Map:         test
      'S' Key expires at: No Valid S-Key
'S' Lifetime Skew:   00h00m10s
  IPSEC AAA Context:                xxx
GRE Sequence Numbers: Disabled          GRE Sequence Mode: None
  GRE Reorder Timeout:  100 msec
  GRE Checksum:         Disabled          GRE Checksum Verification: Disabled
  Registration Revocation:    Enabled     Reg-Revocation I bit:   Enabled
  Reg-Revocation Max Retries: 3           Reg-Revocation Timeout:    3 (secs)
  Reg-Rev Handoff old-FA:     Enabled     Reg-Rev Idle-Timeout:      Enabled
Default Subscriber:      None
  Max Sessions:            500000
  Service Status:          Started
  MN-AAA Auth Policy:      Always
  MN-HA Auth Policy:       Always
  IMSI Auth:               Disabled
  AAA accounting:          Enabled
  Idle Timeout Mode:       Aggressive
  Newcall Policy:          None
  Overload Policy:         Reject    (Reject code: Admin Prohibited)
  NW-Reachability Policy:  Reject    (Reject code: Admin Prohibited)
```

Step 2 Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command **save configuration**. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

Configuring IP Address Pools on the System

One of the steps in establishing a PPP session between the mobile and the PDSN service running on the system is that upon successful authentication, the subscriber's mobile node is assigned an IP address. The IP address could be dynamically assigned from a pool that is configured on the system or on the AAA server. It may also be an address that is statically configured in the user profile or even one that is requested by the subscriber.

IP addresses can be dynamically assigned from a single pool/a group of IP pools/a group of IP pool groups. The addresses/IP pools/ IP pool groups are placed into a queue in each pool or pool group. An address is assigned from the head of the queue and, when released, returned to the end. This method is known as least recently used (LRU).

When a group of pools have the same priority, an algorithm is used to determine a probability for each pool based on the number of available addresses, then a pool is chosen based on the probability. This method, over time, allocates addresses evenly from the group of pools.

Important Note that setting different priorities on each individual pool can cause addresses in some pools to be used more frequently.

To configure the IP pool:

- Create the IP pool for IPv4 addresses in system context by applying the example configuration.
- Optional. Configure the IP pool for IPv6 addresses in system context by applying the example.
- Optional. Configure the overlap-pool addresses to routing by applying the example configuration.
- Verify your IP pool configuration.
- Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command **save configuration**. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

Creating IPv4 Pool

Use the following example to create the IPv4 address pool:

```
configure
 context <dest_ctxt_name>
 ip pool <pool_name> <ipv4/ipv6_address|ipv4/ipv6_address/mask>
 end
```

Following are a few things to be aware of:

- To ensure proper operation, IP pools should be configured within a destination context.
- Each address in the pool requires approximately 24 bytes of memory. Therefore, in order to conserve available memory, the number of pools may need to be limited depending on the number of addresses to be configured and the number of PACs/PSCs installed.
- Setting different priorities on individual pools can cause addresses in some pools to be used more frequently.

- For more information on commands/keywords that configure additional parameters and options, refer ipv6 pool command section in the "Context Configuration Mode Commands" chapter of the *Command Line Interface Reference*.

Creating IPv6 Pool

Use the following example to create the IPv6 address pool:

```
configure
 context <dest_ctxt_name>
 ipv6 pool <pool_name> 6to4 local-endpoint <ipv4/ipv6_address>
 end
```

Following are a few things to be aware of:

- To ensure proper operation, IP pools should be configured within a destination context.
- Each address in the pool requires approximately 24 bytes of memory. Therefore, in order to conserve available memory, the number of pools may need to be limited depending on the number of addresses to be configured and the number of PACs/PSCs installed.
- Setting different priorities on individual pools can cause addresses in some pools to be used more frequently.
- For more information on commands/keywords that configure additional parameters and options, refer ipv6 pool command section in the "Context Configuration Mode Commands" chapter of the *Command Line Interface Reference*.

Adding Overlap-Pool Addresses to Routing

Use the following configuration to advertise overlap-pool addresses in dynamic routing protocols.

```
configure
 context <context_name>
 [ no | default ] ip routing overlap-pool
```

If **ip routing overlap-pool** is configured, then the overlap addresses are added as interface addresses in the routing stack and a route is added in the kernel. The intf-address in the routing stack and the route in the kernel for the overlap address are removed when all the overlap-pools are deleted. The default is **no ip routing overlap-pool**.

Verifying IP Pool Configuration

Step 1 Verify that your IPv4 address pool configured properly by entering the following command in Exec Mode:

```
show ip pool
```

The output from this command should look similar to the sample shown below. In this example all IP pools were configured in the isp1 context.

```
context : isp1:
+-----Type:    (P) - Public    (R) - Private
|              (S) - Static    (E) - Resource
```

```
|
|+----State:   (G) - Good      (D) - Pending Delete   (R)-Resizing
||
||++--Priority: 0..10 (Highest (0) .. Lowest (10))
||||
||||+-Busyout: (B) - Busyout configured
|||||
|||||
vvvvv Pool Name Start Address   Mask/End Address Used     Avail
----- --------- --------------- ---------------  -------- --------
PG00  ipsec             12.12.12.0      255.255.255.0    0        254
RG00  pool3             30.30.0.0       255.255.0.0      0        65534
SG00  pool2             20.20.0.0       255.255.0.0      10       65524
PG00  pool1             10.10.0.0       255.255.0.0      0        65534
SG00  vpnpool            192.168.1.250   192.168.1.254    0        5
Total Pool Count: 5
```

Step 2 Verify that your IPv6 address pools configured properly by entering the following command in Exec Mode:

`show ipv6 pools`

The output from this command should look similar to the sample shown above except IPv6 addresses.