



CHAPTER 6

Cisco Network Registrar Login Event Generator Functional Specification

Published: December 21, 2012

Introduction

This chapter describes the Cisco Network Registrar (CNR) Login Event Generator (LEG) design, logging, tracing, and operations performed by the CNR LEG.

The purpose of this appendix is to provide insight into the CNR LEG operation and integration with CNR.

- [CNR LEG High-Level Design, page 6-2](#)
- [Logging and Tracing, page 6-3](#)
- [Extensions Point Operation, page 6-3](#)

CNR LEG High-Level Design

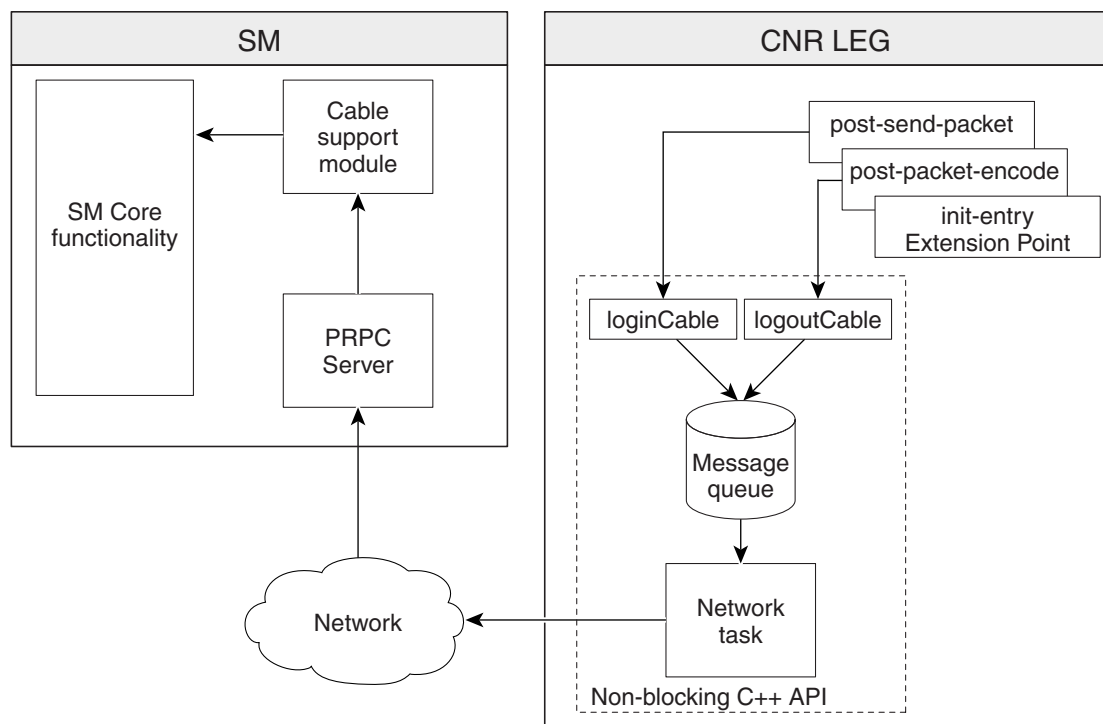
The CNR LEG uses extension points:

- init-entry
- post-packet-decode
- post-send-packet

When an extension point hook is called, the following sequence of events takes place (Figure 6-1):

1. The extension point hook performs the minimal computation necessary to extract all the required data and calls a Nonblocking C++ API operation.
2. The Nonblocking operation encodes a message and places it in a queue.
3. The Nonblocking C++ API network task reads messages from the message buffer and sends them over the network to the PRPC Server on the SM.
4. The PRPC Server decodes the message and passes it to the cable support module, which sets up the subscribers in the SM database using the SM core functionality.

Figure 6-1 CNR LEG High-Level Design



The only operations performed in the context of the CNR extension dispatching thread are message creation and placement in a message queue. A separate thread performs the heavy network operations. Note that if for some reason the message queue is full, the message is dropped to avoid the risk of creating a delay, which would damage CNR performance.

Logging and Tracing

By default, the CNR LEG logs its messages to the CNR log. The LEG supports a debug mode and several trace levels. The LEG configuration file controls logging and tracing.



Note

Changes made to the LEG configuration file become effective only when the LEG is restarted.

Extensions Point Operation

This section briefly describes the operations performed by the CNR LEG at each extension point.

- [init-entry](#), page 6-3
- [post-send-packet](#), page 6-3
- [post-packet-decode](#), page 6-4

init-entry

The extension point **init-entry** initializes and terminates the CNR LEG.

During initialization, the CNR LEG performs the following operations:

- Reading the configuration file
- Initializing the LEG logging and tracing
- Creating a Nonblocking C++ API instance and connecting it to the SM
- Starting the C++ API network-task thread

During termination, the CNR LEG performs the following operations:

- Stopping and freeing the Nonblocking C++ API instance
- Stopping the C++ API network-task thread

post-send-packet

The extension point **post-send-packet** sends the following **cableLogin** operations to the SM:

- Verifying that the request-dictionary is for DHCP REQUEST and the response dictionary is for DHCP ACK.
- Extracting CM-MAC, CPE-MAC, and CMTS-IP from the request dictionary.
- Extracting the assigned CPE-IP and lease time from the response dictionary.
- In CM as Subscriber mode CM requests are ignored.
- Calling the Nonblocking C++ API **cableLogin** or **login** operation with the parameters extracted
- If no CM-MAC (option 82) is found, an attempt to extend the lease based solely on the IP address is performed.

post-packet-decode

The extension point **post-packet-decode** sends the following **cableLogout** or **logout** operations to the Cisco Service Control Subscriber Manager:

- Verifying that the request dictionary is for either DHCP RELEASE or DHCP DECLINE.
- Extracting CM-MAC, CPE-MAC, CPE-IP, and CMTS-IP from the request dictionary.
- Calling the Nonblocking C++ API `cableLogout/logout` operation with the parameters extracted.