Resource Computation Engine (RCE) Users' Manual

Supplemental to DRAGON Software Suite Release V1.0

1. Introduction

Resource Computation Engine (RCE) is a key control-plane element developed by the Dynamic Resource Allocation via GMPLS Optical Networks (DRAGON) Project for resource information management and path computation in GMPLS networks. As a standalone software, RCE collects, stores and correlates resource information from various intradomain and interdomain entities and provides coordinated resource information management and path computation services in the control plane. The design of RCE helps develop several critical features in the DRAGON network, including a) collaborative interdomain resource management; b) interdomain end-to-end path computation; c) advance LSP scheduling; and d) authentication, authorization and accounting (AAA) policy based provisioning.

Departing from this version, RCE will be designed to provide resource information management and path computation services to GMPLS networks based on the DRAGON architecture.

2. RCE Installation Guide

The system requirements for install and compilation of DRAGON RCE are as follows:

- FreeBSD (4.10 or newer release) or LINUX (2.4.20 kernel or newer)
- gcc version 3.2 or newer

The below steps should allow one to complete the install and compile this software.

1. Download a snapshot of the RCE Software from:

    http://dragon.maxgigapop.net/dragonsoftware/dragon-v1.0-rce.tar.gz

2. Install software package in desired directory:

    gzip -cd dragon-v1.0-rce.tar.gz | tar xvf -

3. Compile dragon-v1.0-rce

```
cd dragon-v1.0-rce
./configure --enable-ext-attr
make
```

Note that the "--enable-ext-attr" option enables the feature for extensible resource attribute support.

3. RCE Configuration Guide

RCE does not need a configuration file. The current version does not support CLI commands either.

When the extensible resource attribute feature is enabled, RCE needs to read a RCE schema file, which defines the resource (link) TE parameters that will be stored into the TEDB in RCE. A sample RCE schema file *schema_combo.rsd* is included in dragon-v1.0-rce.tar.gz. The file name is specified by RCE command line argument *-s*.

RCE can be configured using the command line interface (CLI). The default CLI port is 2688. (*telnet localhost 2688*) Some mostly used CLI commands are described below.

(1) *show module*

Show addresses, ports and status of the intER- and intRA-domain instances of OSPFd as well as the associated NARB. Status of OSPFd is two-fold: (a) whether the OSPFd is online; (b) whether the API client connection to OSPFd is alive.

(2) *show topology (interdomain | intradomain)*

Show brief information of the TE database in intER- or intRA-domain OSPFd.

Use the *load-configure* command to load and execute a batch of predefined commands from a file. *exit* or *quit* to logoff the CLI. Or type *configure* to enter the general configuration mode.

Commands in general configuration mode:

(3) *set ospfd (interdomain | intradomain) (HOST) (LCL_PORT) (RMT_PORT) (ORI_IF) (AREA)*

Reconfigure intER- or intRA-domain OSPFd parameters.

(4) *connect ospfd (interdomain | intradomain)*

(Re)Connect to an OSPF daemon. Upon success of connection, TE database is re-synchronized.

4. Running RCE with NARB

RCE provides an API server for other control plane entities to send resource information and query routing paths. Any third-party software can use the API to communicate with RCE. The below steps describe the configuration needed for NARB to use RCE to perform path computation.

1. Configure and run the DRAGON NARB and zebra interdomain and intradomain OSPF daemons. (Refer to the NARB Users' Manual.)
2. Configure the interdomain and intradomain zebra OSPF daemons' host addresses and ports in RCE in the below code lines in rce_main.cc.

```
ZebraOspfSync ospfSyncIntra("arlg-narb", 2617, DOMAIN_MASK_LOCAL, 30);
ZebraOspfSync ospfSyncInter("sands200", 2607, DOMAIN_MASK_GLOBAL, 30);
```

In this example, arlg-narb/2617 and sands200/2607 represent the interdomain and intradomain OSPF daemons, respectively. Note that RCE needs about 30 seconds to synchronize to the OSPF daemons after started.

3. Recompile RCE.

   make

4. Run RCE

   ./rce –d [–f config-file] [-s schema-file] [-P api-port] [-p cli-port] [-h]

5. Log onto DRAGON NARB (*telnet localhost 2626*) using CLI and issue the following commands.

   >configure

   >set rce service-machine-name servier-port

   >>show module

   resource comp engine milk/2678 online connected

   This line indicates that the RCE API server is running on the machine *milk* at port *2678* and an API connection has been created successfully.

6. Send LSP requests to NARB. Now NARB will redirect all CSPF requests, which were originally sent to the interdomain and intradomain OSPF daemons, to RCE. By using “delete rce” in NARB CLI, this configuration can be reset.