Network Resource Aware Broker (NARB) Users' Manual

1. Starting NARB service

To start the NARB service, use the command:

./narb [–d [–f narb.conf]

-d: run in daemon mode
-f: config file (default config file paths: (1)./narb.conf, (2)/usr/local/etc/narb.conf)

2. NARB configuration file

When starting, NARB loads the configuration file in the path specified by the –f option. Without this option, it searches for the default configuration file named *narb.conf* under the current directory and then under /usr/local/etc. NARB aborts upon failing to load the configuration file.

Lines starting with ! are comment lines and will be ignored. The configuration file consists of multiple configuration blocks starting with a block id, followed by a block body in {}. A configuration block may contain sub-level blocks. The top-level blocks are divided into four parts as described below.

(1) Domain ID configuration

domain-id { (ip *IP*) | (id *NUM*) }

Domain ID is used to associate each advertised intER-domain router or link with a unique domain. A domain ID can be expressed as an IP address or an unsigned integer number.

(2) OSPFd configuration

inter-domain-ospfd {address *HOST* port *NUM* originate-interface *IP* area *IP*}
intra-domain-ospfd {address *HOST* port *NUM* originate-interface *IP* area *IP*}

These two configuration blocks describe the location and configuration information of the intER- and intrRA-domain OSPF daemons.

(2) Abstract topology configuration

We have two options as described below to generate the abstract topology.

Option A. Complete manual configuration:

In an abstract topology, we use a *router* configuration block to describe each router.

```
router {id IP
    link { id IP type NUM
        local_if IP remote_if IP
        ……
        Optional TE parameters
        vlan_tags {NUM, NUM …}
    }
    link {id IP type NUM
        local_if IP remote_if IP
        ……
```

Optional TE parameters
vlan_tags {*NUM, NUM …*}
}
}

A *router* block has a *router id* and one or more sub-level *link* blocks. Each *link* block represents a TE link originating from this router. The *router id* will be its advertising router. Four parameters, *link id, link type,* and *local* and *remote interface addresses,* are mandatory. In CSPF calculation, *router id, link id* and *local* and *remote interface addresses* are used together to generate the network graph. Optional link TE parameters include *metric*, *max_bw*, *max_rsv_bw*, *unrsv_bw0*.. *unrsv_bw7*, *enc_type*, *sw_type* and *vlan_tags*. When the parameter *metric* does not show up, NARB sets a default metric value 1. The parameters *sw_type* and *enc_type* etc. are needed for constraining CSPF routing computation. By default, *sw_type, enc_type* and those bandwidth parameters are set to 0 by NARB. Only if proper TE parameters show up, can a CSPF route be found through the link. Note that *vlan_tags* is the DRAGON proprietary parameter, which represents a set of VLAN tags associated with a TE link.

Option B. Automatic virtual TE link probing:

We define types of services in the service blocks.

service {id *NUM* sw_type *NUM* enc_type *NUM* max_bw *FLOAT*}

We define the router block a bit different than Option A.

router {id *IP_ADDR* type *NUM*

svc_probes {*NUM NUM NUM …* }

}

The type of a router could be 0 (default), 1 (host) or 2 (border router). Each *NUM* in the svc_probes block corresponds to the id *NUM* in a service block, representing the service type we want to use to constrain the virtual TE link probing. NARB will probe virtual TE links by sending CSPF requests to the intRA-domain OSPFd for each pair of host and border router and for each pair of border routers. The service parameters of the corresponding svc_probes *NUM*'s will be used as CSPF routing constraints. Each svc_probes *NUM* produces one probing. Each successful probing will result in a virtual TE link with the specified service parameters in the corresponding service block.

*Check with the authors to make sure your NARB version does support Option B.

(3) Inter-domain TE link configuration

inter-domain-te-link {id *IP* narb-peer *HOST* port *NUM*}

This kind of configuration blocks defines the inter-domain TE links. We can have one or more *inter-domain-te-link* blocks. The *id* here must be to one of the *link id* defined in the above abstract topology, which corresponds to a *router id* in a neighboring domain. *narb-peer* and *port* describe the location information of a peering NARB server in a neighboring domain, which is associated with this TE link.

(4) CLI configuration

cli {host *IF_ADDR* password *STRING* }

host specifies an interface the NARB CLI server listens on. By default, the CLI server listens on all interfaces. With logging on, CLI users will be asked for *password*.

3. NARB CLI commands

After logging in to the NARB CLI, type "*show module*" to check the availability of collaborating services; "*configure*" to enter general configuration mode; "*exit*" or "*quit*" to log out.

(1) *show module*

Show addresses, ports and status of the intER- and intRA-domain instances of OSPFd, resource computation engineer (RCE) and peering NARB's in neighboring domains. Status of OSPFd is two-fold: (a) whether the OSPFd is online; (b) whether the API client connection to OSPFd is alive. Status of a peering NARB also shows a list of next-domain interface addresses associated with the NARB.

Commands in general configuration mode:

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

(Re)Connect to an OSPF daemon. Upon success of connection, domain summary is (re)originated.

(3) *delete ospfd (interdomain | intradomain)*

Disconnect to an OSPF daemon. Threads for LSA origination/refreshment, NARB-LSDB synchronization and asynchronous read are cancelled.

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

Reconfigure intER- or intRA-domain OSPFd parameters.

(5) *set peer-narb (HOST) port (NUM) via (IF_ADDR)*

Reconfigure parameters for a peering NARB. A new interface address *IF_ADDR* associated with this NARB may be added.

(6) *set routing-mode (all-loose-allowed | all-strict-only | mixed-allowed | mixed-preferred)*

Configure the routing mode that determines how routing request is processed.

- *all-strict-only* mode: All ERO hops must be strict hops.
- *mixed-preferred* mode: Only the hops in the first domain are strict hops. The hops in next domains must be loose hop.
- *mixed-allowed* mode: Return as many as possible strict hops. Loose hops are allowed when they cannot be expanded into strict hops.
- *all-loose-allowed* mode: All ERO hops may be loose hops.

(7) *exit*

Exit from the NARB configuration mode.

Commands in topology configuration mode:

(1) *show topology (local | global)*

‘show topology local’ displays all the router-id and te-link LSA’s originated from this NARB as well as the IDs of intER-domain TE links.

(2) *delete topology*

Remove all LSA’s representing the manually configured topology of this domain.

(3) *set topology (FILE)*

(Re)Originate the LSA’s from the topology configuration FILE.

(4) *set refresh-interval SECONDS*

Set the value interval between refreshments, which re-originate LSA’s that represent the abstract topology.

(5) *update link local_if_addr LCL_IF_ADDR rmt_if_addr RMT_IF_ADDR*

Enter the link update mode. The link is identified by the local and remote interface addresses.

(6) *add link local_if_addr LCL_IF_ADDR rmt_if_addr RMT_IF_ADDR*

Add a new link and enter the link update mode.

(7) *add link local_if_addr LCL_IF_ADDR rmt_if_addr RMT_IF_ADDR*

Add a new link and enter the link update mode.

(8) *set / delete / show rce*

Set, remove or show RCE configuration.

(9) *exit*

Exit from the topology configuration mode.

Commands in ink update mode (entered from the general configuration mode):

(1) *show link (original | updated)*

‘show original’ displays the original link ID and TE parameters obtained from the configuration file. ‘show updated’ displays the link ID and TE parameters changed under the update mode.

(2) *set (link_id | adv_router | metric | lcl_if | rmt_if) VALUE*

Set link ID, advertising router and TE parameters to VALUE.

(3) *set sw_capability (lsc | tdm | psc1 | psc2 | psc3 | psc4) encoding (ethernet | lambda | packet | sdh)*

Set the switching capability related TE parameters.

(4) *commit*

Originate and updated TE link LSA and send it to the intER-domain OSPFd.

(5) *cancel*

Cancel and quit the link update mode.

4. Running NARB in an experimental environment

Step 1. Check the path of the NARB configuration file. Make sure that (1) the configuration file uses a default name and is under default directories; or (2) a correct path is specified by the –f option in the command line. Note that we may put the command in a shell script called *run*.

Step 2. Start the zebra daemon and the intER- and intRA-domain OSPF daemons. Make sure the OSPFd information is set correctly in NARB configuration file. Note that the setting can be changed in NARB CLI if it were misconfigured.

cd *zebra-path*
./zebra –d
cd *ospfd-path*
./run

Step 3. Log into intER-domain OSPFd CLI. Check whether the LSA's originated by NARB represent a desired abstract topology.

telnet *localhost* 2614 /* for intER-domain OSPFd */

Step 4. Start the NARB daemon. Enter general configuration mode in NARB CLI. Use 'show module' to check if OSPFd's and peering NARB's have the correct status. For OSPFd, the status should be *online* and *connected.* For peering NARB, the status should be *online* and each next-domain interface address associated with the peering NARB should appear.

cd *narb-path*
./run
telnet *localhost* 2626
> show module

Appendix I. Trouble shooting

When odds happen with NARB service, try some of the solutions below.

(1) If some LSA's were not originated as expected, enter configuration mode in NARB CLI. Try '*delete topology*' and '*set topology FILE*.'

(2) If connections to OSPFd go wrong, enter configuration mode in NARB CLI. Try '*disconnect ospfd (interdomain | intradomain)*' and then '*connect ospfd (interdomain | intradomain).*' You may also want to change the OSPFd configuration by using '*set ospfd (interdomain | intradomain) …*'

(3) If peering NARB's become *offline*, use the command '*set peer-narb …*' to reconfigure peering-NARB information. Check if the NARB in the peering domain has been started. Then '*show module*' will show it comes back online.

(4) If individual links originated by this NARB are misconfigued, use '*update link*' or '*test delete link*' in the topology configuration mode.

(5) Restart OSPFd and/or RCE and/or NARB servers.