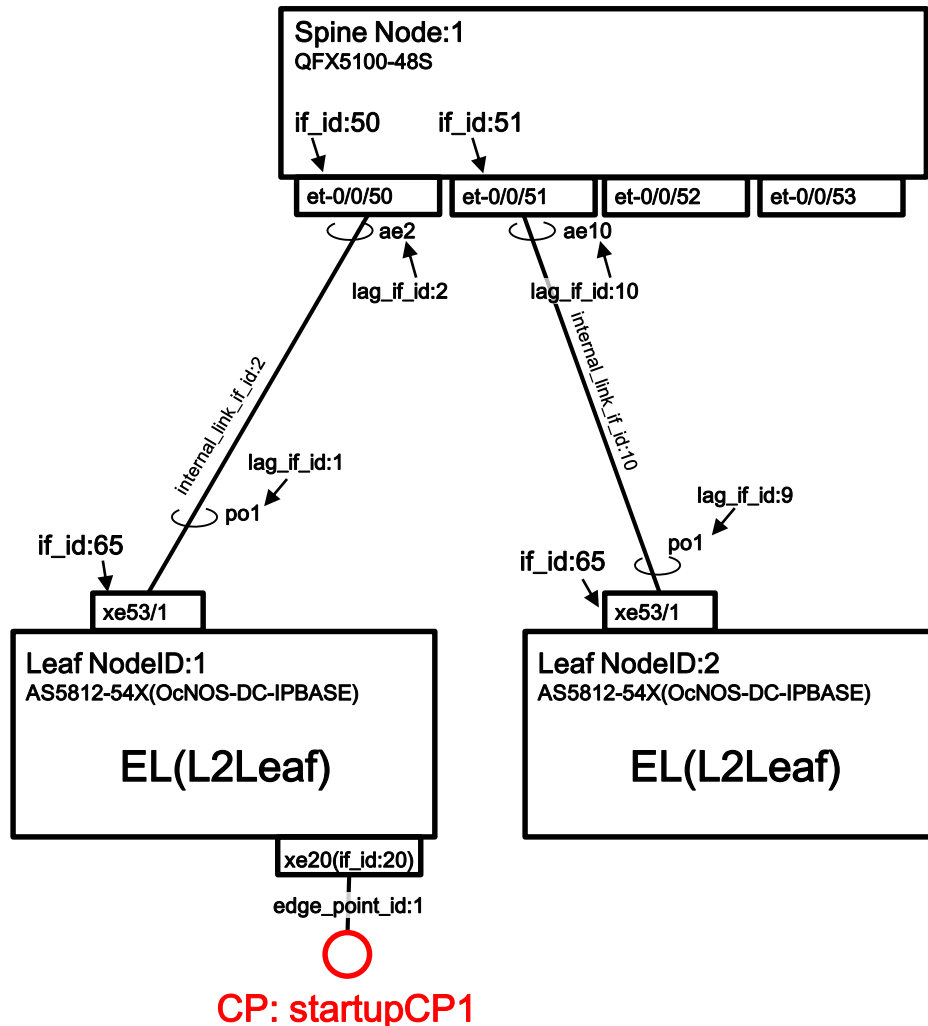

Start-up MSF

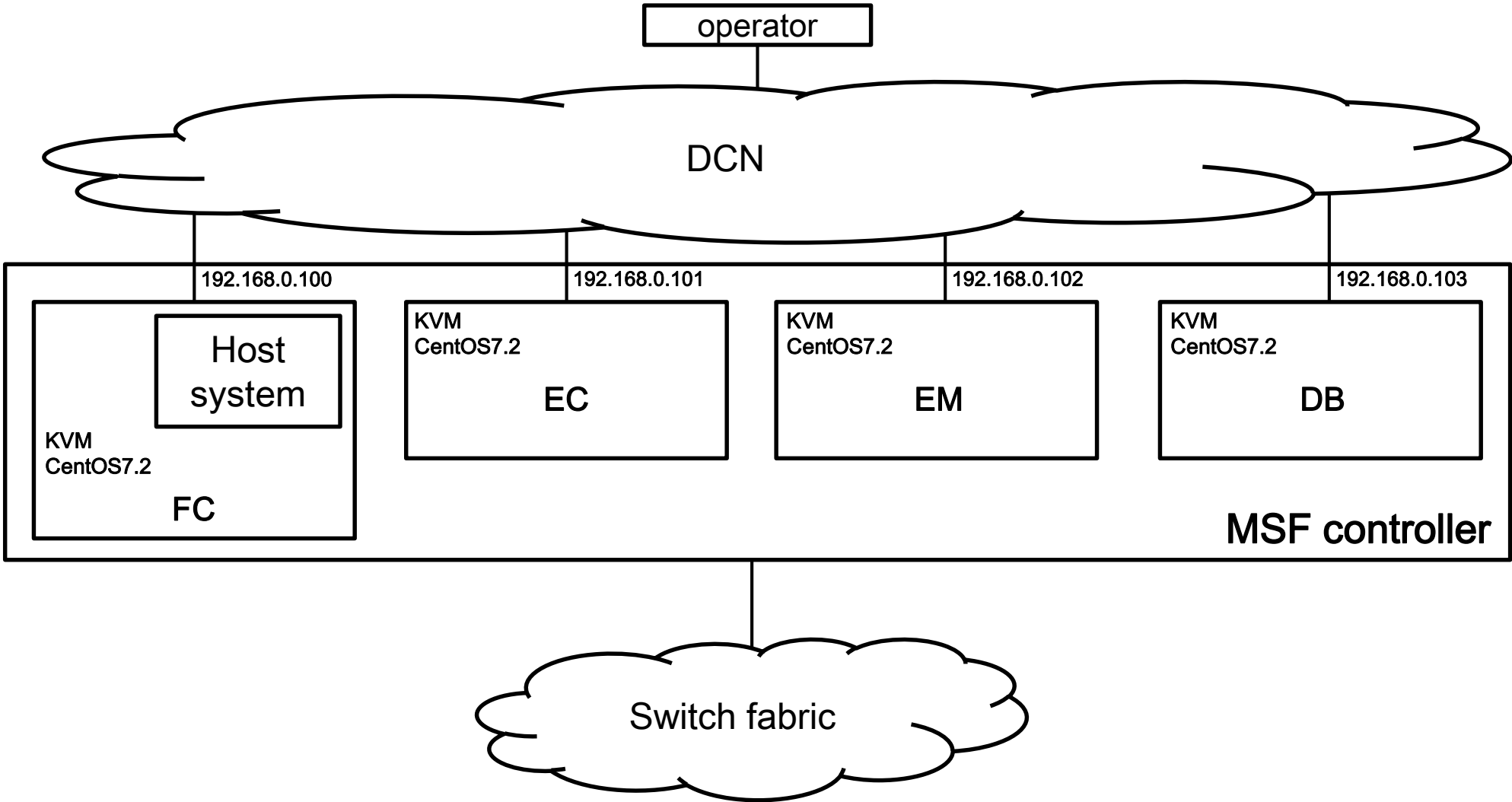
Dec. 2018

The goal of start-up

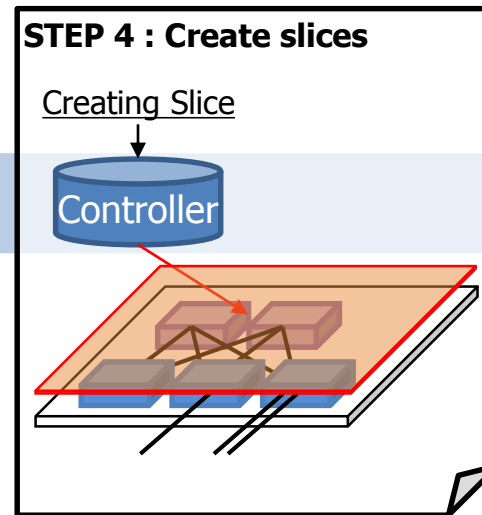
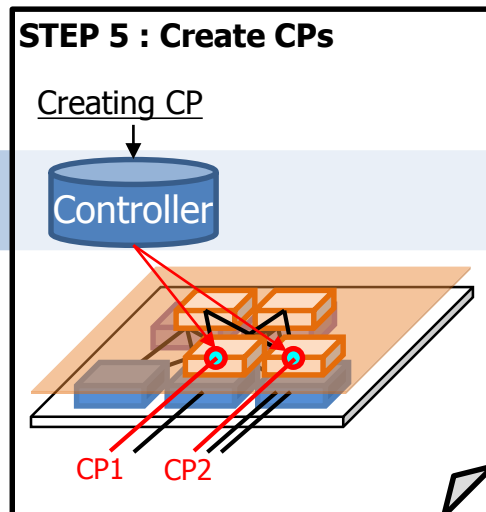
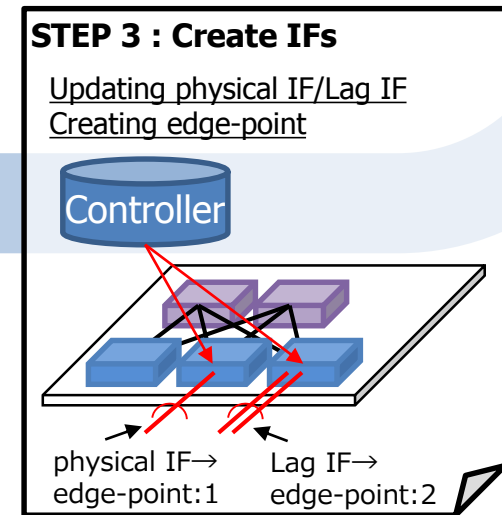
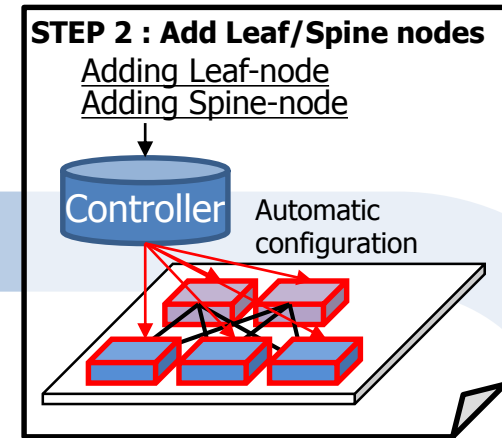
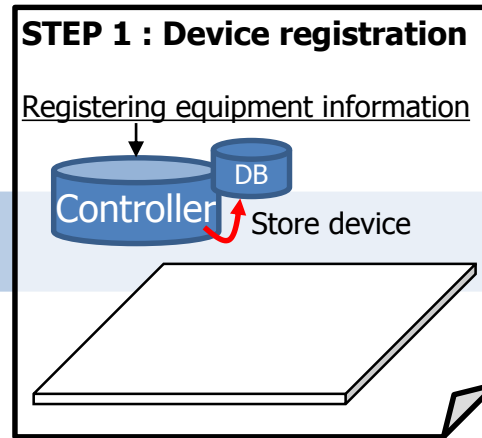
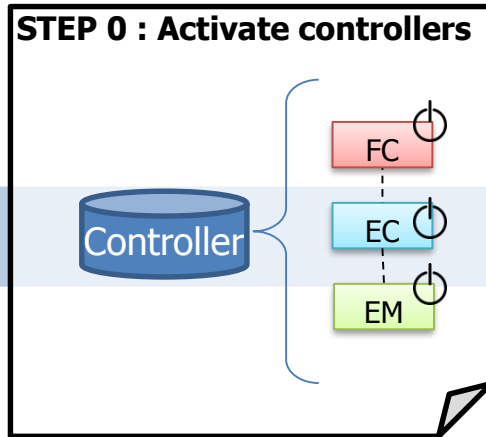
The CP (startupCP1) is created by MSF controllers.



Network configuration

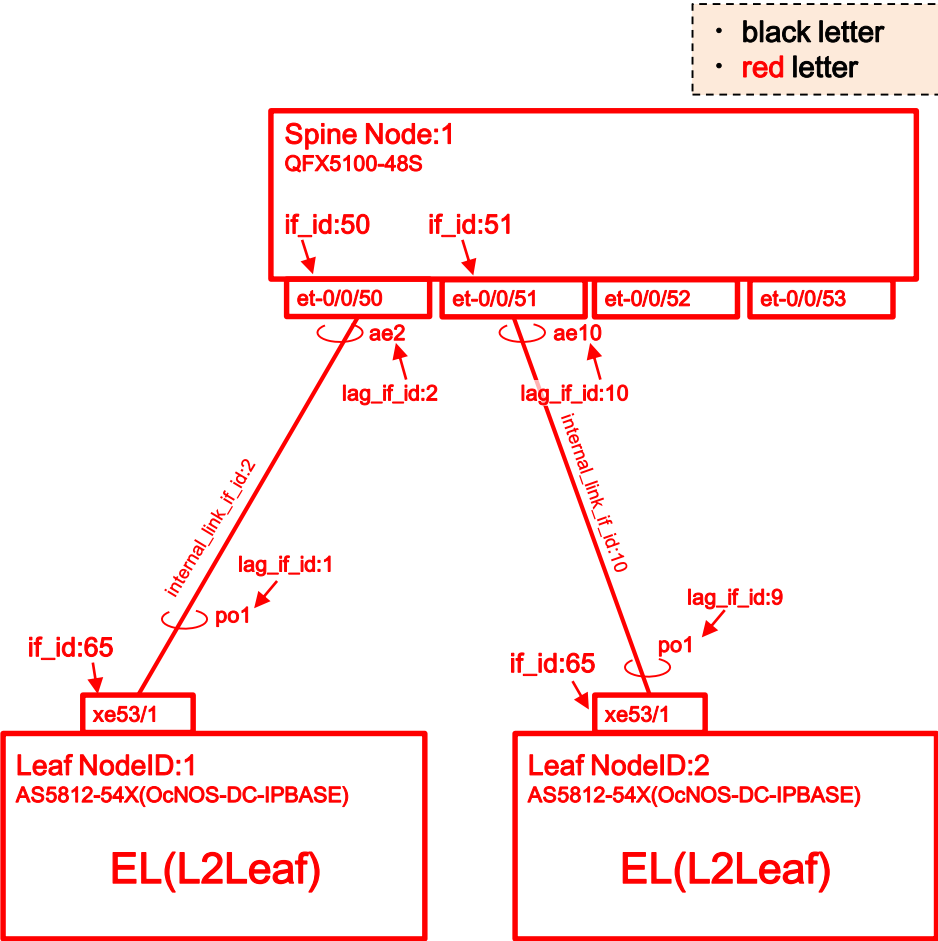


Operation flow



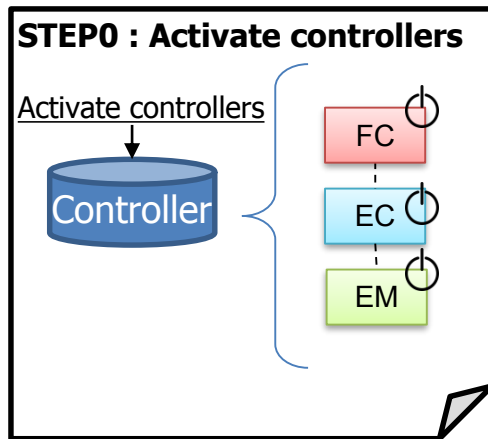
Step 0 to 2 : Fabric building

In step 0 to 2, **fabric** is created as shown below.



Step0 : Activate controllers

- Execute start shell and check status of each controllers.
 - Interface name : Controller status confirmation
 - URI : /v1/internal/MSFcontroller/status (FC), /v1/internal/ec_ctrl/statusget (EC/EM)



Step0 : Activate controllers

■Fabric Controller(FC) start

```
[root@MSF-FC bin]# /home/msf-controller/bin/fc_ctl.sh start  
[root@MSF-FC bin]#
```

■Element Controller(EC) start

```
[root@MSF-EC bin]# /usr/ec_main/bin/ec_ctl.sh start  
<error> Notify start changeover Failed.  
<error> Notify start changeover Failed.  
<error> Notify start changeover Failed.  
[root@MSF-EC bin]#
```

Step0 : Activate controllers

■Element Manager(EM) start

```
[root@MSF-EM bin]# /opt/em/bin/em_ctl.sh start
EM START: PRECHECK OPERATION RUNNING...
EM STATUS: CHECKING EXISTENCE OF EM PROCESS...
EM STATUS: [ SUCCESS ] NO RUNNING PROCESS
EM START: EM MAIN MODULE STARTING...
<error> Notify start changeover Failed.
<error> Notify start changeover Failed.
EM START: AFTER START CHECK RUNNING...
EM STATUS: CHECKING EXISTENCE OF EM PROCESS...
EM STATUS: PROCESS CONFIRMED.
EM STATUS: START MONITORING MODULE...
EM STATUS: [ FAILURE ] ERROR OCCURED IN MONITORING MODULE 1
EM START: CHECKING MAIN MODULE STATUS...
EM STATUS: CHECKING EXISTENCE OF EM PROCESS...
EM STATUS: PROCESS CONFIRMED.
EM STATUS: START MONITORING MODULE...
EM STATUS: [ SUCCESS ] MONITOR SUCCESSFULLY OPERATED
EM START: [ SUCCESS ] MAIN MODULE SUCCESSFULLY STARTED.
[root@MSF-EM bin]#
```


Step0 : Activate controllers (status check)

■ Fabric Controller(FC) status check

```

      ↙ FC REST address
$ curl http://192.168.0.100:18080/v1/internal/MSFcontroller/status
{
  "service_status": "running",      ...Startup complete
  "blockade_status": "none"         ...Operation accepted
}
```

Step0 : Activate controllers (status check)

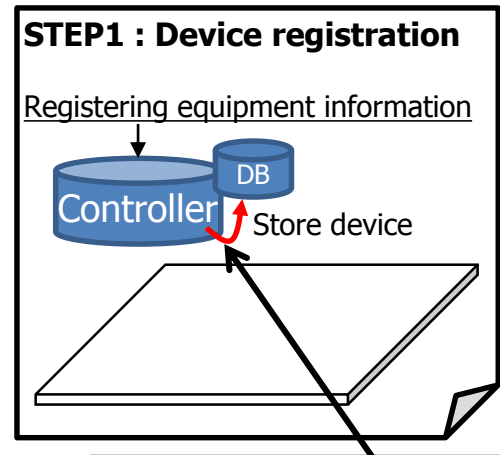
■Element Controller(EC), Element Manager(EM) status check

```

      ↙ EC REST address
$ curl http://192.168.0.101:18080/v1/internal/ec_ctrl/statusget
{"ec_status":{"status":"inservice","busy":"inservice"},"em_status":{"status":"inservice"},"informations":.....
EC Startup complete                               EM Startup complete
```

Step1 : Device registration

- Register device information to be used in the cluster.
 - Interface name : Registering equipment information
 - URI : /v1/equipment-types



Device A

slot1

slot2

slot3

slot4

☐ capability : L2orL3

☐ dhcp, snmp

☐ IF

1g:ge-

10g:xe-

lag:ae-

☐ slots

slot:1→physical port ID:1, ge- or xe-

Typical parameters

body	overview	remarks
platform	platform	
firmware	firmware version	
capability	I2/I3 VPN compatibility	
dhcp/snmp	DHCP, SNMP	
if_definitions	IF information (port, speed, prefix, ..)	
slots	slot information	Mapping of physical port ID and slot.

Step1 : Device registration

“Registering equipment information” registers the model information of the equipment to be used on the MSF network in the controller.

■Registering equipment information REST body sample

```
"equipment_type": {  
  "platform": "Sample",           ...platform name (ex. QFX5100)  
  "os": "sample",                 ...OS name (ex. Junos)  
  "firmware": "sample",           ...OS version (ex. 15.xxx)  
  "router_type": "normal",        ...Available only for "normal"  
  "physical_if_name_syntax": null, ...Available only for "null"  
  "breakout_if_name_syntax": "<PORTPREFIX><IFSOLTNNAME><BREAKOUTIFSUFFIX>", ...Breakout-IF name syntax  
  "breakout_if_name_suffix_list": "0:1:2:3", ...breakout_if_name_suffix_list  
  "capability": {                 ...Capability information  
    "vpn": {                       ...VPN capability information  
      "l2": true,                  ...L2VPN support  
      "l3": false                 ...L3VPN support  
    },  
    "qos": {                       ...QoS Capability  
      "remark": false,  
      "remark_capability": null,  
      "remark_default": null,  
      "shaping": false,  
      "egress_queue_capability": null,  
      "egress_queue_default": null  
    }  
  },  
  "dhcp": {  
    "dhcp_template": "/root/setup/dhcp_template/dhcpd.conf.qfx5100", ...File path of "dhcpd.conf"  
    "config_template": "/initial-config/juniper/ztp.conf.qfx5100-48s_EL", ...File path of initial config template  
    "initial_config": "/initial-config/juniper/ztp.conf.qfx5100-48s_EL" ...File path of initial config  
  },  
  "snmp": {  
    "if_name_oid": "1.3.6.1.2.1.31.1.1.1.1", ...MIB information of IF name  
    "snmptrap_if_name_oid": "1.3.6.1.2.1.31.1.1.1.1", ...MIB information of IF name in the SNMP trap  
    "max_repetitions": 100, ...Maximum number to get with GET Bulk  
  },  
  "boot_complete_msg": "UI_COMMIT_COMPLETED: commit complete", ...Syslog message for confirming the startup  
  "boot_error_msgs": null, ...Syslog message for confirming the failure  
}
```

```
"if_definitions": {  
  "ports": [ ...Available port information  
    {  
      "speed": "40g", ...Port speed  
      "port_prefix": "et-", ...Port name prefix  
    },  
    {  
      "speed": "10g",  
      "port_prefix": "xe-"  
    },  
    {  
      "speed": "1g",  
      "port_prefix": "ge-"  
    }  
  ],  
  "lag_prefix": "ae", ...LAG IF name prefix  
  "unit_connector": ".", ...Unit IF connector  
  "slots": [ ...Physical slot information  
    {  
      "if_id": "0", ...Physical port ID  
      "if_slot": "0/0/0", ...IF slot name  
      "speed_capabilities": [ ...Available port speed  
        "4g",  
        "10g"  
      ]  
    },  
    {  
      "if_id": "1",  
      "if_slot": "0/0/1",  
      "speed_capabilities": [  
        "4g",  
        "10g"  
      ]  
    }  
  ],  
  ..... (slot information follows)
```

Step1 : Device registration

■Registering equipment information →→Adding sample equipment-type (ID ... 4)

```
$ curl -X POST ¥ http://192.168.0.100:18080/v1/equipment-types ¥ -H 'content-type: application/json'
¥ -d '{  "equipment_type": {      "platform": "Sample",      "os": "sample",      "firmware": "sample",
"router_type": "normal", .....
curl: (6) Could not resolve host:
{
  "equipment_type_id": "4"
}
```

...Complete

■Getting equipment list in switch cluster →→Check the adding equipment-type

```
$ curl http://192.168.0.100:18080/v1/equipment-types
{
  "equipment_type_ids": [
    "1",
    "2",
    "3",
    "4"
  ]
}
```

...added equipment-type

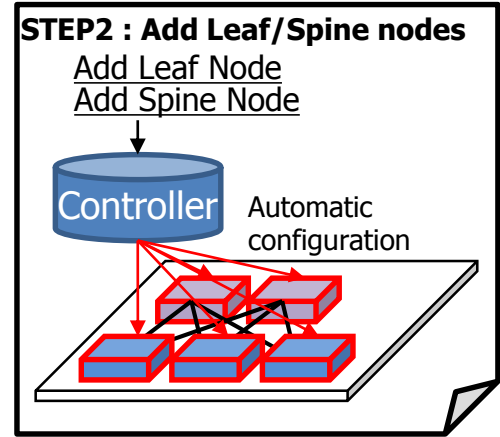
Step1 : Device registration

■Getting equipment information →→Confirm the detailed parameters

```
$ curl http://192.168.0.100:18080/v1/equipment-types/4
{
  "equipment_type": {
    "equipment_type_id": "4",
    "platform": "Sample",
    "os": "sample",
    "firmware": "sample",
    "router_type": "normal",
    "breakout_if_name_syntax":
      .....
```

Step2 : Add Leaf/Spine nodes

- Add leaf and Spine nodes.
 - Interface name : Adding Leaf-node, Adding Spine-node
 - URI : /v1/clusters/{cluster_id}/nodes/leafs, /v1/clusters/{cluster_id}/nodes/spines



- When Leaf is added, the controller also sets the appropriate configuration for the connected Spine.
- If you add the device that has been already configured (you don't use ZTP), you set the "provisioning" body is "false", and set the same conditions for other parameters.
- You need to add Leaf/Spine node one by one.

Typical parameters (Adding Leaf-node)

body	overview	remarks
node_id	Serial number for device	Created by FC
equipment_type_id	model ID	
provisioning	Device setting necessity flag	True: Built-in device not set False: Embed setting device
VPN_type	I2/I3 VPN type	One of "I2" and "I3"
plane	Belonging side	Set "1".
internal_links	Internal link information	

Typical parameters (Adding Spine-node)

body	overview	remarks
node_id	Serial number for device	Created by FC
equipment_type_id	model ID	
provisioning	Device setting necessity flag	True: Built-in device not set False: Embed setting device
internal_links	Internal link information	

Step2 : Add Leaf/Spine nodes

■Adding Spine-node1

```
[root@MSF-FC logs]# curl -H 'Content-type: application/json' ¥
> -X POST ¥
> "http://192.168.0.100:18080/v1/clusters/1/nodes/spines?notification_address=192.168.0.100&notification_port=19090" ¥
> -d '{
>   "node_id":"1",
>   "equipment_type_id":"1",
>   "host_name":"Startup-Spine01",
>   "mac_address":"d8:18:d3:67:ba:41",
>   "username":"startup",
>   "password":"Startup3000",
>   "provisioning":false,           ...Initialized equipment(true) or Configured equipment(false)
>   "snmp_community":"ntt-msf",
>   "ntp_server_address":"192.168.0.254",
>   "breakout":null,
>   "internal_links":null,
>   "management_if_address":"192.168.0.244",
>   "management_if_prefix":24
> }'
{
  "operation_id": "0011538453749816"   ...Return operation ID
}[root@MSF-FC logs]#
```


Step2 : Add Leaf/Spine nodes

■Adding Leaf-node1

```
[root@MSF-FC logs]# curl -H 'Content-type: application/json' ¥
> -X POST ¥
> "http://192.168.0.100:18080/v1/clusters/1/nodes/leafs?notification_address=192.168.0.100&notification_port=19090" ¥
> -d '{
>   "node_id" : "1",
>   "equipment_type_id" : "2",
>   "leaf_type" : "EL",    ...EL or IL
>   "host_name" : "Startup-L2-Leaf01",
>   "mac_address" : "a8:2b:b5:0d:c0:e0",
>   "username" : "admin",
>   "password" : "Startup3000",
>   "provisioning" : false, ...Initialized equipment(true) or Configured equipment(false)
>   "vpn_type" : "l2", "plane" : 1,
>   "snmp_community" : "ntt-msf",
>   "ntp_server_address" : "192.168.0.254",
>   "breakout" : null,
>   "internal_links" : { "lag_links" : [{
>     "opposite_node_id" : "1", "local_traffic_threshold" : null, "opposite_traffic_threshold" : null, "member_ifs" : [{
>       "local" : { "physical_if" : { "physical_if_id" : "65", "physical_if_speed" : "40g" }, "breakout_if" : null },
>       "opposite" : { "physical_if" : { "physical_if_id" : "50", "physical_if_speed" : "40g" }, "breakout_if" : null } } ] },
>   "management_if_address" : "192.168.0.248",
>   "management_if_prefix" : 24>   }'
{
  "operation_id": "0011538455845737"    ...Return operation ID
}[root@MSF-FC logs]#
```

Step2 : Add Leaf/Spine nodes

■Adding Leaf-node2

```
[root@MSF-FC logs]# curl -H 'Content-type: application/json' ¥
> -X POST ¥
> "http://192.168.0.100:18080/v1/clusters/1/nodes/leafs?notification_address=192.168.0.100&notification_port=19090" ¥
> -d '{
>   "node_id" : "2",
>   "equipment_type_id" : "2",
>   "leaf_type" : "EL",    ...EL or IL
>   "host_name" : "Startup-L2-Leaf02",
>   "mac_address" : "a8:2b:b5:0d:c4:0e",
>   "username" : "admin",
>   "password" : "Startup3000",
>   "provisioning" : false, , ...Initialized equipment(true) or Configured equipment(false)
>   "vpn_type" : "l2", "plane" : 1,
>   "snmp_community" : "ntt-msf",
>   "ntp_server_address" : "192.168.0.254",
>   "breakout" : null,
>   "internal_links" : { "lag_links" : [{ "opposite_node_id" : "1", "local_traffic_threshold" : null,
>     "opposite_traffic_threshold" : null, "member_ifs" : [{
>       "local" : { "physical_if" : { "physical_if_id" : "65", "physical_if_speed" : "40g" }, "breakout_if" : null },
>       "opposite" : { "physical_if" : { "physical_if_id" : "51", "physical_if_speed" : "40g" }, "breakout_if" : null } } ] },
>   "management_if_address" : "192.168.0.249",
>   "management_if_prefix" : 24
> }'
{
  "operation_id": "0011538456809941"    ...Return operation ID
}[root@MSF-FC logs]#
```

Step2 : Add Leaf/Spine nodes (Get)

■Get Spine-node

```
[root@MSF-FC logs]# curl http://192.168.0.100:18080/v1/clusters/1/nodes/spines?user-type=operator
{
  "spine_node_ids": [
    "1"          ...added Spine-node
  ]
}[root@MSF-FC logs]#
```

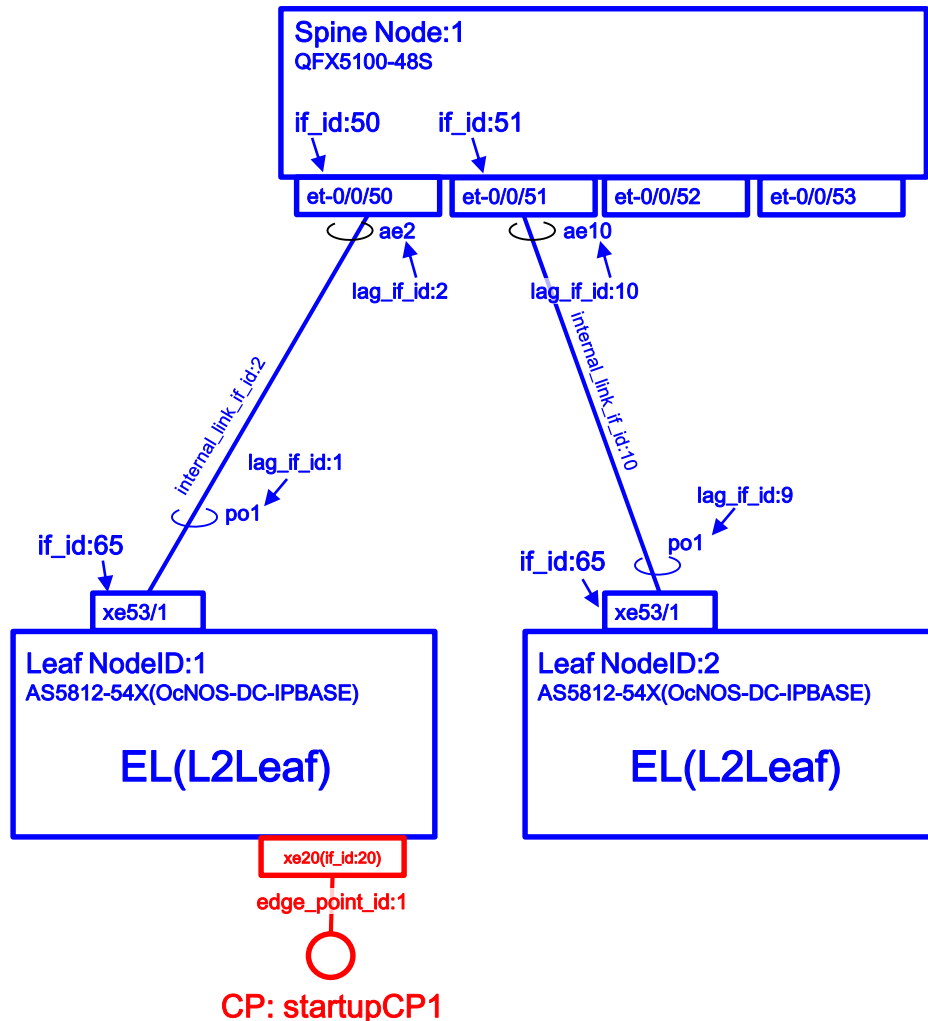
■Get Leaf-node

```
[root@MSF-FC logs]# curl http://192.168.0.100:18080/v1/clusters/1/nodes/leafs
{
  "leaf_node_ids": [
    "1",          ...added Leaf-node
    "2"          ...added Leaf-node
  ]
}[root@MSF-FC logs]#
```

Network configuration after Step 2

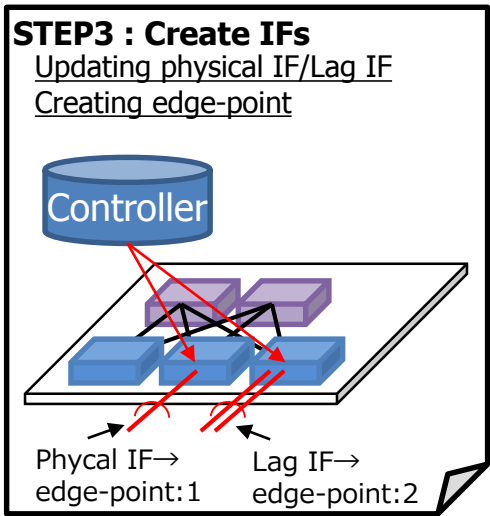
Current network configurations are shown below.
In step 3 to 5, **Slice**, **IFs** and **CP** are created.

- black letter ...Actual setting of the devices
- blue letter ...set by MSF controller in step 0 to 2
- red letter ...set by MSF controller in step 3 to 5



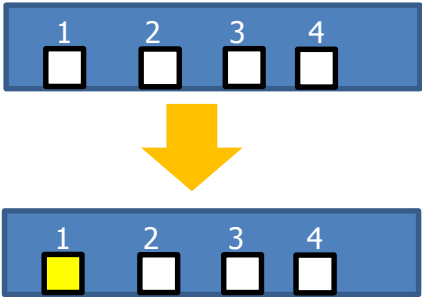
Step3 : Create IFs - physical interface -

- Determine the speed of the physical interface. (not confirmed at device registration)
- The selectable speed is the value defined at device registration.
 - Interface name : Updating information of physical interface
 - URI : /v1/clusters/{cluster_id}/nodes/{fabric_type}/{node_id}/interfaces/physical-ifs/{if_id}



Parameters

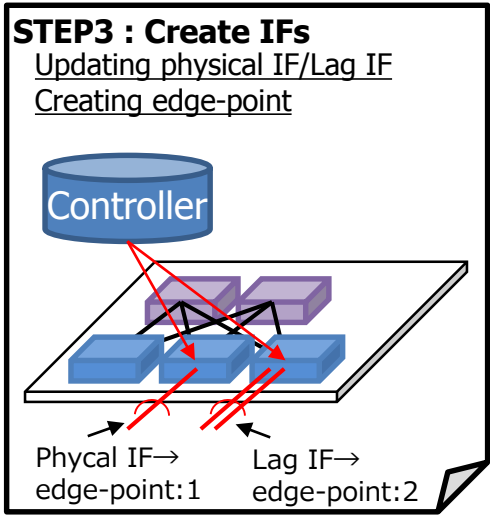
body	overview	remarks
cluster_id	Switch cluster ID	Identify the target physical IF
fabric_type	Device type	
node_id	Device ID	
if_id	Physical IF ID	
action	Control type	
speed	IF speed	



set speed

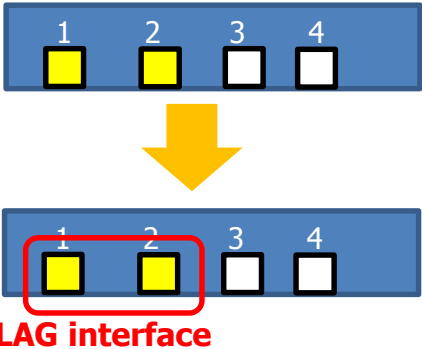
Step3 : Create IFs - LAG interface -

- Create the LAG-IF from several physical interfaces set speed.
 - Interface name : Creating Link-aggregation interface
 - URI : /v1/clusters/{cluster_id}/nodes/{fabric_type}/{node_id}/interfaces/lag-ifs



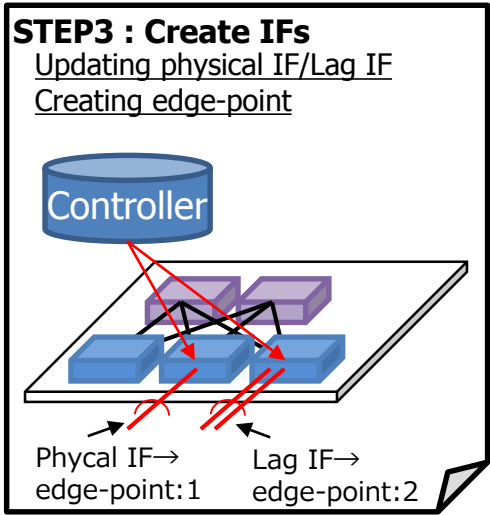
Parameters

body	overview	remarks
cluster_id	Switch cluster ID	Identify the target physical IF
fabric_type	Device type	
node_id	Device ID	
physical_if_ids	List of Physical IF ID	



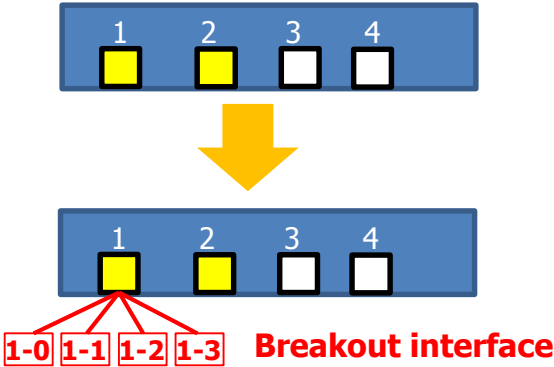
Step3 : Create IFs - Breakout interface -

- Create the Breakout-IF.
 - Interface name : Creating or deleting breakout interface
 - URI : /v1/clusters/{cluster_id}/nodes/{fabric_type}/{node_id}/interfaces/breakout-ifs



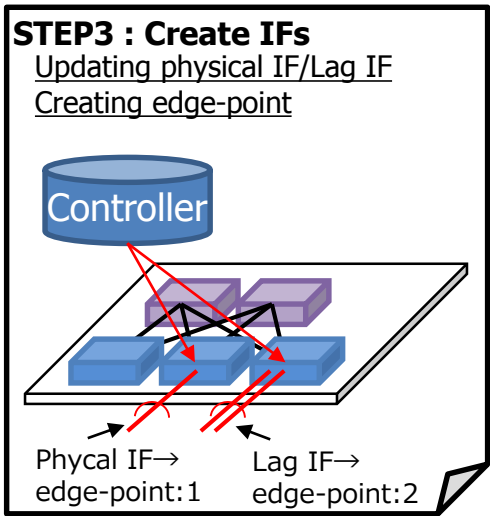
Parameters

body	overview	remarks
op	control type	create: "add" delete: "remove"
path	Breakout IF ID	"/" + "breakout IF ID"
physical_if_id	target physical IF ID	
division_number	number of divisions of physical IF	
breakout_if_speed	Speed of each breakout IF	



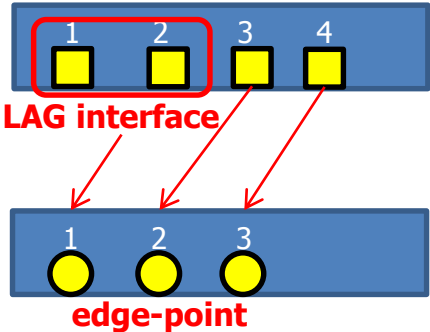
Step3 : Create IFs - edge-point -

- Create edge-point so that the upper systems do not identify the interface type. The CP is registered on the edge-point.
- You can not register another edge-point in the IF where the edge-point is already registered.
 - Interface name : Creating edge-point
 - URI : /v1/clusters/{cluster_id}/points/edge-points



Parameters

body	overview	remarks
cluster_id	Switch cluster ID	
leaf_node_id	Leaf device ID	
laag_if_id	LAG IF ID	Specify either LAGIF ID or Physical IF ID
physical_if_ids	Physical IF ID	



Step3 : Create IFs - physical interface -

■Updating physical interfaces -> Set 1G speed to the physical IF:20

```
[root@MSF-FC logs]# curl -v -H 'Content-type: application/json' ¥
> -X PUT ¥
> http://192.168.0.100:18080/v1/clusters/1/nodes/leafs/1/interfaces/physical-ifs/20 ¥
> -d '{
>     "action":"speed_set",
>     "speed":"1g"
> }'
* About to connect() to 192.168.0.100 port 18080 (#0)
* Trying 192.168.0.100...
* Connected to 192.168.0.100 (192.168.0.100) port 18080 (#0)
> PUT /v1/clusters/1/nodes/leafs/1/interfaces/physical-ifs/20 HTTP/1.1
> User-Agent: curl/7.29.0
> Host: 192.168.0.100:18080
> Accept: */*
> Content-type: application/json
> Content-Length: 65
>
* upload completely sent off: 65 out of 65 bytes
< HTTP/1.1 200 OK
< Date: Tue, 02 Oct 2018 05:17:39 GMT
< Access-Control-Allow-Origin: *
< Access-Control-Allow-Methods: GET, POST, DELETE, PUT, PATCH
< Access-Control-Allow-Headers: Content-Type
< Content-Length: 0
< Server: Jetty(9.3.11.v20160721)
<
* Connection #0 to host 192.168.0.100 left intact
[root@MSF-FC logs]#
```

Step3 : Create IFs (Get) - physical interface -

■Get physical interfaces -> physical IF:20

```
[root@MSF-FC logs]# curl http://192.168.0.100:18080/v1/clusters/1/nodes/leafs/1/interfaces/physical-ifs/20
{
  "physical_if": {
    "physical_if_id": "20",
    "speed": "1g",
    "if_name": "xe20"
  }
}
}[root@MSF-FC logs]#
```

Physical IF ID

Step3 : Create IFs - edge-point -

■Create edge-piont

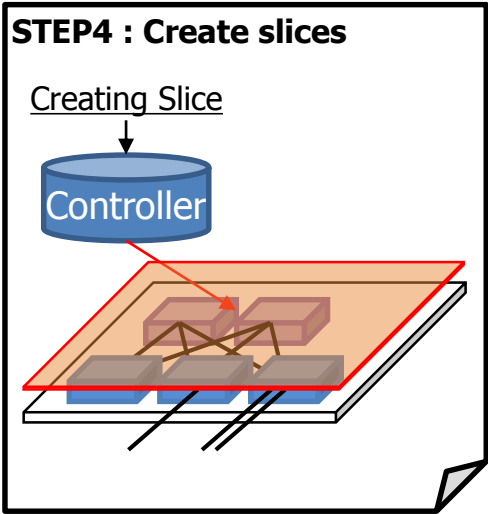
```
[root@MSF-FC logs]# curl -H 'Content-type: application/json' ¥  
> -X POST ¥  
> http://192.168.0.100:18080/v1/clusters/1/points/edge-points ¥  
> -d '{  
>   "leaf_node_id" : "1",  
>   "physical_if_id" : "20"  
> }'  
{  
  "edge_point_id": "1"  
}  
}[root@MSF-FC logs]#
```

■Get edge-piont

```
[root@MSF-FC logs]# curl -X GET http://192.168.0.100:18080/v1/clusters/1/points/edge-points?format=detail-list  
{  
  "edge_points": [  
    {  
      "edge_point_id": "1",  
      "support_protocols": {  
        "L2": true,  
        "L3": false,  
        "L3_protocols": []  
      }  
    }  
  ]  
}  
}[root@MSF-FC logs]#
```

Step4 : Create slices

- Create network slice.
 - Interface name : Creating Slice
 - URI : /v1/slices/{slice_type}



Parameters

body	overview	remarks
slice_type	Slice type	"l2vpn" : L2 slice "l3vpn" : L3 slice
slice_id	Slice ID	If it is not specified, FC creates ID.

Step4 : Create slices

■Create L2Slice

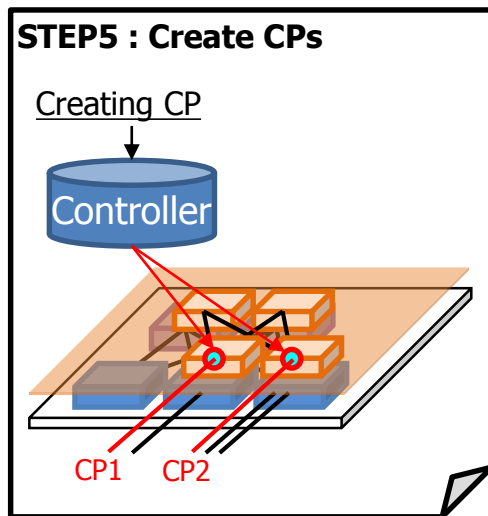
```
[root@MSF-FC logs]# curl -H 'Content-type: application/json' ¥
> -X POST ¥
> http://192.168.0.100:18080/v1/slices/l2vpn ¥
> -d '{
>     "slice_id": "startup",
>     "remark_menu": null
> }'
{
  "slice_id": "startup"
}[root@MSF-FC logs]#
```

■Get L2Slice

```
[root@MSF-FC logs]# curl -X GET http://192.168.0.100:18080/v1/slices/l2vpn?format=detail-list
{
  "l2_slices": [
    {
      "slice_id": "startup",
      "l2_cp_ids": []
    }
  ]
}[root@MSF-FC logs]#
```

Step5 : Create CPs

- CP is set above the edge-point.
- L3CP needs to specify the protocol to be used. (BGP, OSPF, static, VRRP)
 - Interface name : Creating CP
 - URI : /v1/slices/{slice_type}/{slice_id}/cps



Parameters (slice type -> L2 slice)

body	overview	remarks
slice_type	Slice ID	"l2vpn"
slice_id	Slice ID	
cluster_id	Switch cluster ID	
edge_point_id	Edge-point ID to be created for CP	
vlan_id	VLAN ID	VLAN ID of CP
cp_id	Create CP ID	
port_mode	Port mode of VLAN	"access" or "trunk"

Typical parameters (slice type -> L3 slice)

body	overview	remarks
slice_type	Slice ID	"l2vpn"
slice_id	Slice ID	
ipv4_addr	Housing equipment IF address	
bgp	Information for BGP	specified when setting BGP
static_routes	Static Route information list	specified when setting static
vrrp	information for VRRP	specified when setting VRRP

Step5 : Create CPs

■Creating CP -> Create the L2CP “startupCP1”

```
[root@MSF-FC logs]# curl -v -H "Accept: application/json" -H "Content-type: application/json" -X POST ¥
-d '{
  "cluster_id" : "1",
  "edge_point_id" : "1",
  "vlan_id" : 4001,
  "cp_id" : "startupCP",
  "pair_cp_id" : null,
  "qos" : {"ingress_shaping_rate" : null, "egress_shaping_rate" : null, "egress_queue_menu" : null},
  "port_mode" : "access"
}' http://192.168.0.100:18080/v1/slices/l2vpn/startup/cps?notification_address=192.168.0.100¥&notification_port=19090
{
  "operation_id": "0011538459176544"
}
* Connection #0 to host 192.168.0.100 left intact
}[root@MSF-FC logs]#
```

Slice ID

Step5 : Create CPs

■Get CP -> Create the L2CP "startupCP1"

```
[root@MSF-FC logs]# curl http://192.168.0.100:18080/v1/slices/l2vpn/startup/cps
{
  "l2_cp_ids": [
    "startupCP1"
  ]
}[root@MSF-FC logs]#
```


Step5 : Create CPs

Check the device configuration.

```
Startup-L2-Leaf01>show running-config
```

```
...
```

```
vlan database  
vlan 4001 bridge 1 state enable
```

...VLAN configuration

```
...
```

```
interface xe20  
switchport
```

...IF setting

```
...
```

```
nvo vxlan access-if port xe20  
no shutdown  
map vnid 2
```

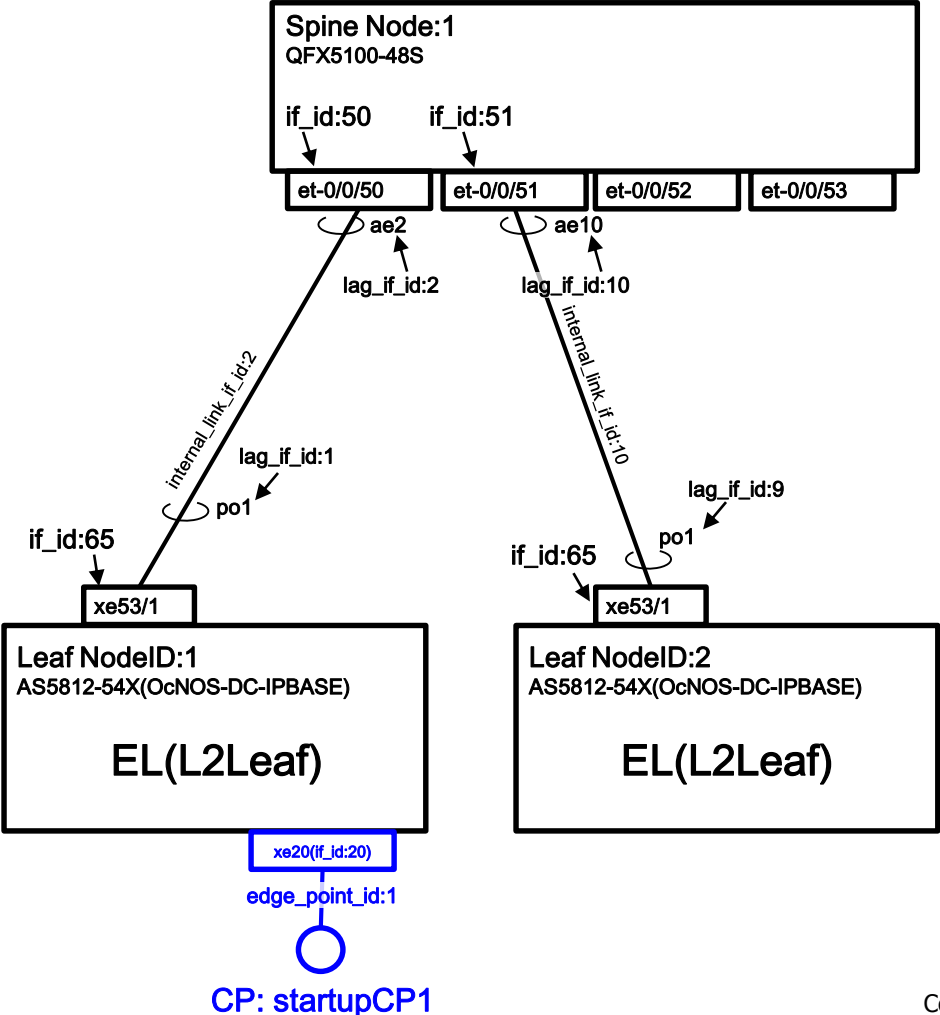
...VNI<->VLAN mapping

```
...
```

Network configuration after Step 5

Current network configuration

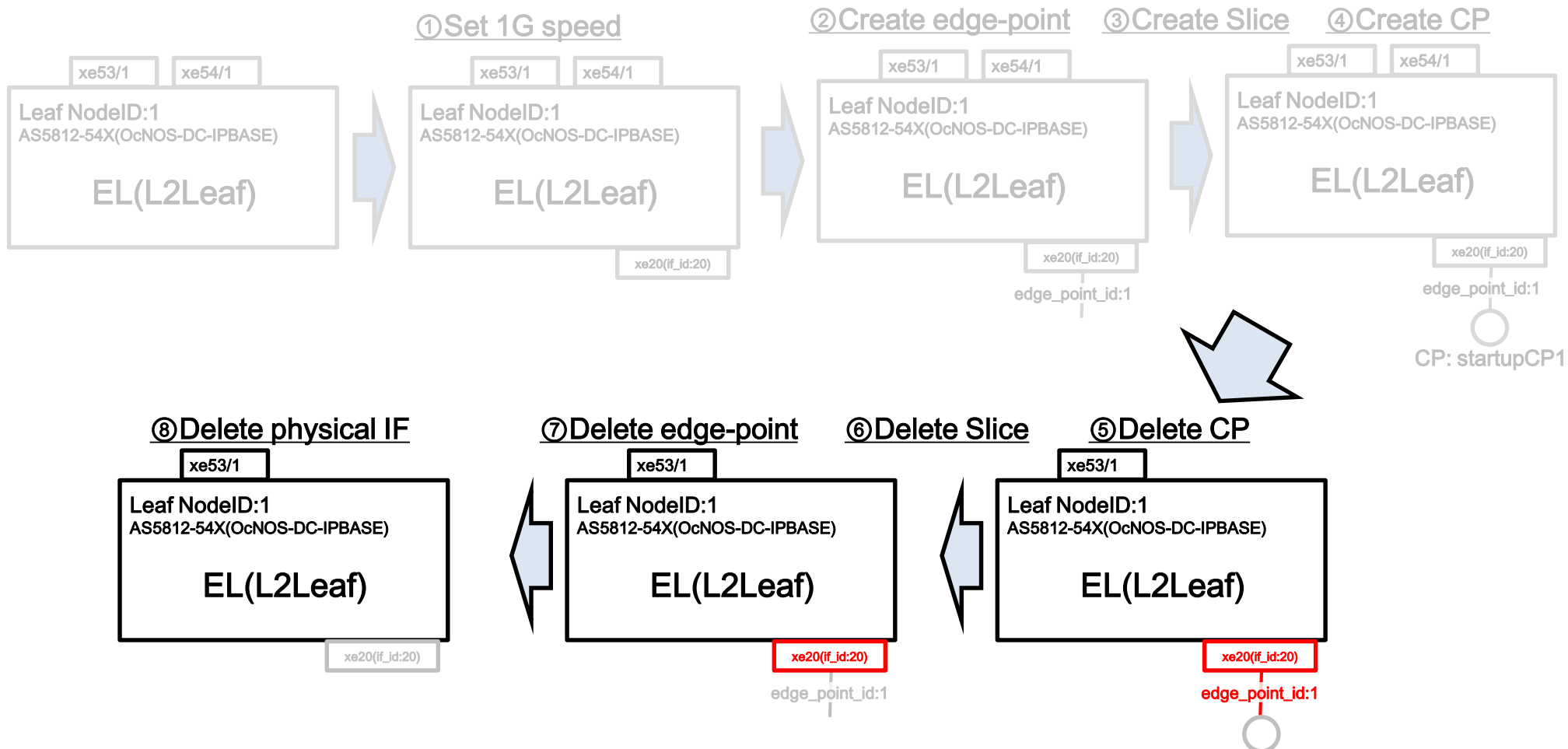
- black letter ...Actual setting of the devices
 - blue letter ...set by MSF controller in step 3 to 5



Step 6 : Delete CP/Slice/IF setting

Be sure to set in the following order.

Execution in an unexpected order may cause an error,
and additionally, the state mismatch of the controller may occur.



Step 6 : Delete CP/Slice/IF setting

■Deleting CP -> Delete the L2CP "startupCP1"

```
[root@MSF-FC logs]# curl -X DELETE
"http://192.168.0.100:18080/v1/slices/l2vpn/startup/cps/startupCP1?notification_address=192.168.0.10
0&notification_port=19090"
                                Slice ID      CP ID
{
  "operation_id": "0011538459433945"
}[root@MSF-FC logs]#
```

■Get CP -> Delete the L2CP "startupCP1"

```
[root@MSF-FC logs]# curl http://192.168.0.100:18080/v1/slices/l2vpn/startup/cps
                                Slice ID
{
  "l2_cp_ids": []
}[root@MSF-FC logs]#
```

Step 6 : Delete CP/Slice/IF setting

■Deleting Slice -> Delete the L2Slice “startup”

```
[root@MSF-FC logs]# curl -v -X DELETE http://192.168.0.100:18080/v1/slices/l2vpn/startup_
* About to connect() to 192.168.0.100 port 18080 (#0)
*   Trying 192.168.0.100...
* Connected to 192.168.0.100 (192.168.0.100) port 18080 (#0)
> DELETE /v1/slices/l2vpn/startup HTTP/1.1
> User-Agent: curl/7.29.0
> Host: 192.168.0.100:18080
> Accept: */*
>
< HTTP/1.1 204 No Content
< Date: Tue, 02 Oct 2018 05:54:24 GMT
< Access-Control-Allow-Origin: *
< Access-Control-Allow-Methods: GET, POST, DELETE, PUT, PATCH
< Access-Control-Allow-Headers: Content-Type
< Server: Jetty(9.3.11.v20160721)
<
* Connection #0 to host 192.168.0.100 left intact
[root@MSF-FC logs]#
```

Slice ID

■Get Slice -> Delete the L2Slice “startup”

```
[root@MSF-FC logs]# curl http://192.168.0.100:18080/v1/slices/l2vpn
{
  "l2_slice_ids": []
}[root@MSF-FC logs]#
```

Step 6 : Delete CP/Slice/IF setting

■Delete edge-point

```
[root@MSF-FC logs]# curl -v -X DELETE http://192.168.0.100:18080/v1/clusters/1/points/edge-points/1_
* About to connect() to 192.168.0.100 port 18080 (#0)
*   Trying 192.168.0.100...
* Connected to 192.168.0.100 (192.168.0.100) port 18080 (#0)
> DELETE /v1/clusters/1/points/edge-points/1 HTTP/1.1
> User-Agent: curl/7.29.0
> Host: 192.168.0.100:18080
> Accept: */*
>
< HTTP/1.1 204 No Content
< Date: Tue, 02 Oct 2018 05:59:37 GMT
< Access-Control-Allow-Origin: *
< Access-Control-Allow-Methods: GET, POST, DELETE, PUT, PATCH
< Access-Control-Allow-Headers: Content-Type
< Server: Jetty(9.3.11.v20160721)
<
* Connection #0 to host 192.168.0.100 left intact
[root@MSF-FC logs]#
```

Edge-point ID

■Get edge-point

```
[root@MSF-FC logs]# curl -X GET http://192.168.0.100:18080/v1/clusters/1/points/edge-points?format=detail-list
{
  "edge_points": []
}[root@MSF-FC logs]#
```

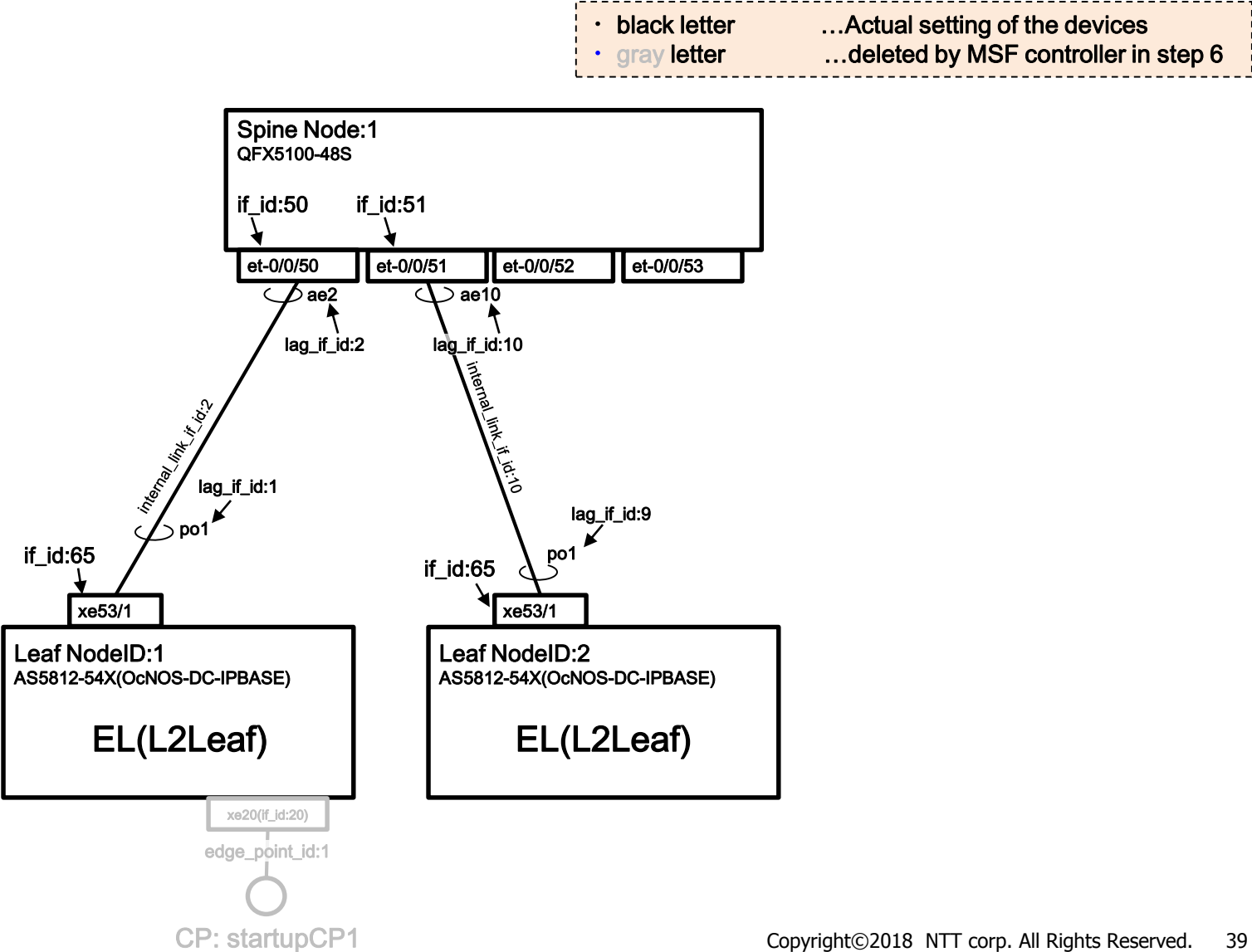
Step 6 : Delete CP/Slice/IF setting

■Updating physical interface -> Delete the speed setting at physical IF:20

```
[root@MSF-FC logs]# curl -v -H 'Content-type: application/json' ¥
> -X PUT ¥
> http://192.168.0.100:18080/v1/clusters/1/nodes/leafs/1/interfaces/physical-ifs/20 ¥
> -d '{
>     "action":"speed_delete"
> }'
Physical IF ID
* About to connect() to 192.168.0.100 port 18080 (#0)
* Trying 192.168.0.100...
* Connected to 192.168.0.100 (192.168.0.100) port 18080 (#0)
> PUT /v1/clusters/1/nodes/leafs/1/interfaces/physical-ifs/20 HTTP/1.1
> User-Agent: curl/7.29.0
> Host: 192.168.0.100:18080
> Accept: */*
> Content-type: application/json
> Content-Length: 45
>
* upload completely sent off: 45 out of 45 bytes
< HTTP/1.1 200 OK
< Date: Tue, 02 Oct 2018 06:01:17 GMT
< Access-Control-Allow-Origin: *
< Access-Control-Allow-Methods: GET, POST, DELETE, PUT, PATCH
< Access-Control-Allow-Headers: Content-Type
< Content-Length: 0
< Server: Jetty(9.3.11.v20160721)
<
* Connection #0 to host 192.168.0.100 left intact
[root@MSF-FC logs]#
```

```
[root@MSF-FC logs]# curl http://192.168.0.100:18080/v1/clusters/1/nodes/leafs/1/interfaces/physical-ifs/20
{
  "physical_if": {
    "physical_if_id": "20"
  }
}
[root@MSF-FC logs]# }
```

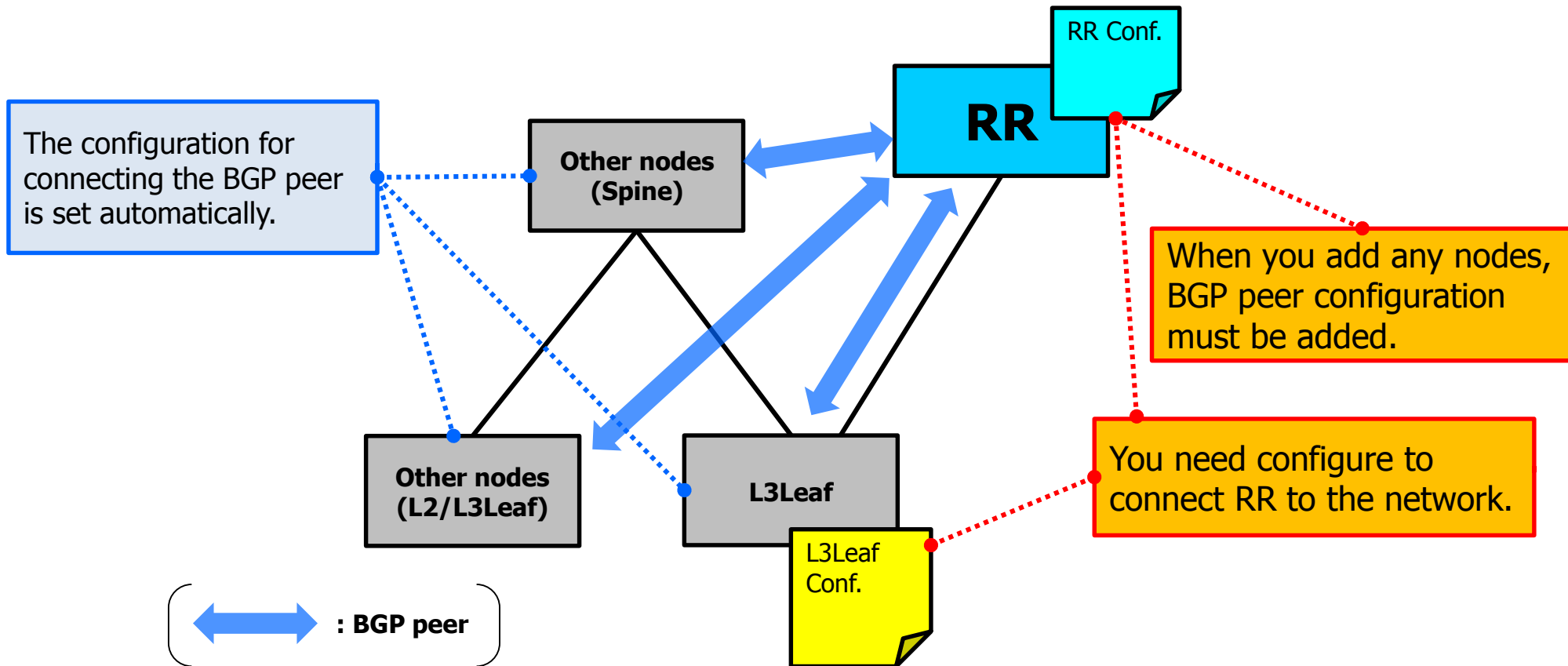
Network configuration after Step 6



<Appendix> Initial config

<Appendix> RR (BGP Route Reflector) setting

- You need to set the RR configuration and network setting yourself, because the current controller does not implement the function of automating the setting of RR.
- The node configuration for peering the neighbor with RR is set automatically when Leaf/Spine is added, by registering the ID and loopback address of RR in the initial configuration of FC. But you need to configure the added node as a neighbor in RR conf. when you add any nodes.
- When both L2Leaf and L3Leaf exist, RR must be connected to L3Leaf.



<Appendix> sample configuration of RR

```
hostname soetsu-RR1
clock timezone JST 9
logging trap alerts
logging buffered 1250000
logging buffered debugging
logging facility local5
logging source-interface Loopback0
service timestamps log datetime msec
service timestamps debug datetime msec
telnet vrf default ipv4 server max-servers 100
domain lookup disable
ntp
server 192.168.134.14
source MgmtEth0/RSP0/CPU0/0
update-calendar
!
interface Loopback0
ipv4 address 10.0.100.1 255.255.255.255
!
interface MgmtEth0/0/CPU0/0
ipv4 address 192.168.2.36 255.255.0.0
!
interface GigabitEthernet0/0/0/0
description To_Leaf4
mtu 4110
ipv4 address 10.121.54.202 255.255.255.252
!
route-policy PASS_ALL
pass
end-policy
!
router ospf v4 MSF OSPF
router-id 10.0.100.1
mpls ldp auto-config
dead-interval 40
hello-interval 10
timers throttle spf 200 200 2000
area 0
interface Loopback0
cost 10
passive enable
!
interface GigabitEthernet0/0/0/0
cost 100
priority 10
!
!
router bgp 64050
timers bgp 30 90
bgp router-id 10.0.100.1
address-family vpnv4 unicast
!
```

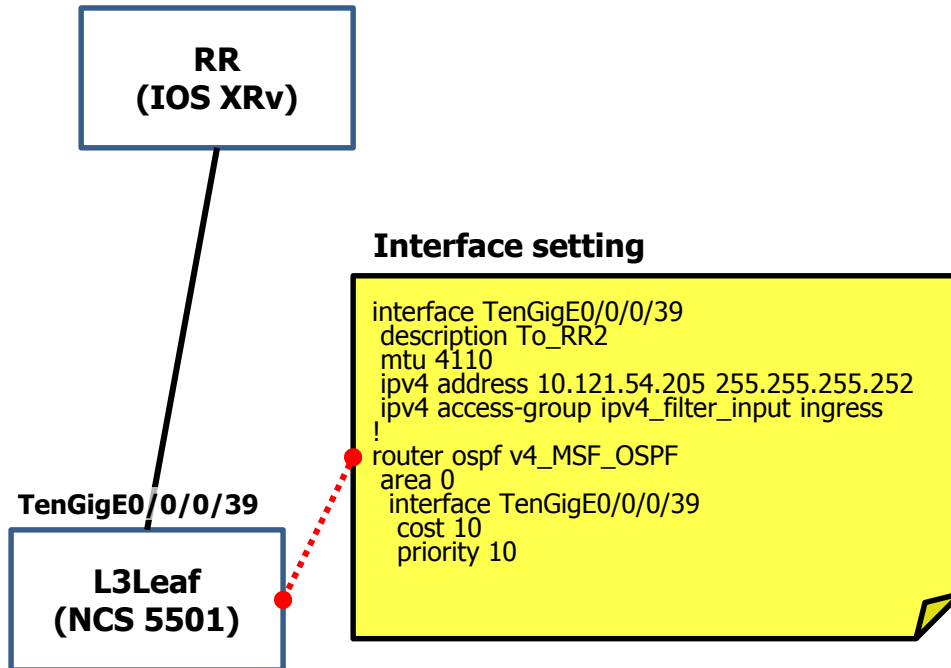
```
neighbor 10.0.1.1
remote-as 64050
update-source Loopback0
address-family vpnv4 unicast
route-policy PASS_ALL in
route-reflector-client
route-policy PASS_ALL out
!
!
neighbor 10.0.1.2
remote-as 64050
update-source Loopback0
address-family vpnv4 unicast
route-policy PASS_ALL in
route-reflector-client
route-policy PASS_ALL out
!
!
neighbor 10.0.1.3
remote-as 64050
update-source Loopback0
address-family vpnv4 unicast
route-policy PASS_ALL in
route-reflector-client
route-policy PASS_ALL out
!
!
neighbor 10.0.1.4
remote-as 64050
update-source Loopback0
address-family vpnv4 unicast
route-policy PASS_ALL in
route-reflector-client
route-policy PASS_ALL out
!
!
neighbor 10.0.1.5
remote-as 64050
update-source Loopback0
address-family vpnv4 unicast
route-policy PASS_ALL in
route-reflector-client
route-policy PASS_ALL out
!
!
neighbor 10.0.1.6
remote-as 64050
update-source Loopback0
address-family vpnv4 unicast
route-policy PASS_ALL in
route-reflector-client
route-policy PASS_ALL out
!
!
```

```
mpls ldp
router-id 10.0.100.1
interface GigabitEthernet0/0/0/0
discovery hello holdtime 15
discovery hello interval 5
!
!
ssh server vrf default
end
```

Configuration of the BGP neighbor
When you add any nodes, you need to add it.

<Appendix> sample configuration to connect RR

NCS 5001



QFX 5100

