



L2BSA User Guide

Version 22.7.1, 25 July 2022

Registered Address	Support	Sales
26, Kingston Terrace, Princeton, New Jersey 08540, United States		
		+91 80 4850 5445
http://www.rtbrick.com	support@rtbrick.com	sales@rtbrick.com

©Copyright 2022 RtBrick, Inc. All rights reserved. The information contained herein is subject to change without notice. The trademarks, logos and service marks ("Marks") displayed in this documentation are the property of RtBrick in the United States and other countries. Use of the Marks are subject to RtBrick's Term of Use Policy, available at <https://www.rtbrick.com/privacy>. Use of marks belonging to other parties is for informational purposes only.

Table of Contents

1. Introduction	3
1.1. L2BSA Services and Subscribers	5
1.2. A10-NSP L2X Services	6
1.3. Supported Platforms	6
2. L2BSA Configuration	7
2.1. AAA Profile Configuration	7
2.2. Intermediate Agent Configuration	7
3. L2BSA Operational Commands	9
4. L2BSA Operational State API	11
4.1. L2BSA Services API	11
4.2. L2BSA Subscribers API	15
4.3. A10-NSP L2X Services API	16

1. Introduction

Layer 2 Bitstream Access (L2BSA) refers to a scenario in which a service provider makes his access infrastructure available to other service providers. These are retail service providers who provide their own Internet services. In Germany this service is mandated by the Federal Network Agency (German: Bundesnetzagentur or BNetzA) which is the regulatory office for electricity, gas, telecommunications, post and railway markets. It is a federal agency of the Federal Ministry for Economic Affairs and Energy and headquartered in Bonn, Germany.

The definition of the L2BSA service was defined by the so called NGA Forum, an advisory board founded in May 2010 by the Bundesnetzagentur for promoting dialogue between the Bundesnetzagentur, network operators, manufacturers, states and local authorities on NGA rollout.

https://www.bundesnetzagentur.de/EN/Areas/Telecommunications/Companies/MarketRegulation/NGAForum/NGAForum_node.html

The L2BSA specification defines two interfaces, the so called U interface (User Interface) at the customer location and the A10-NSP interface (A10 Network Service Provider) between the service provider networks. Those interface types were originally introduced in the Broadband Forum TR-101 (Migration to Ethernet Based Broadband Aggregation).

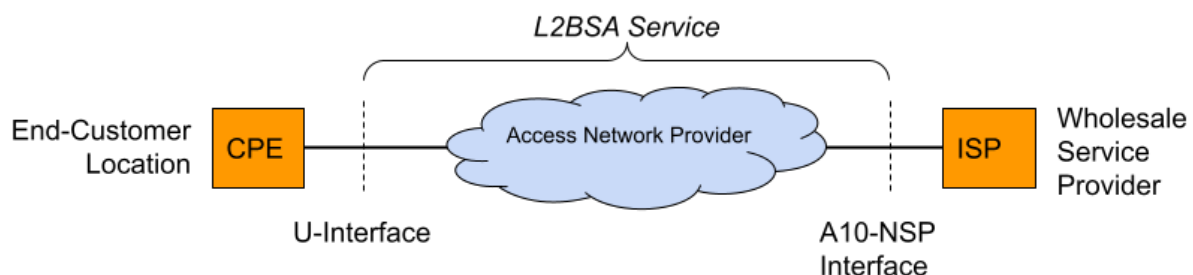


Figure 1. L2BSA Interfaces

The U interface is defined as a transparent Layer 2 interface which can be used with or without VLAN tags by the wholesale service providers. This means that some CPE will send their traffic untagged while other CPE are configured for tagged traffic which needs to be transparently forwarded between U interface and A10-NSP interface.

The A10-NSP interface is defined as a link aggregation bundle interface with one or more interfaces and LACP enabled or disabled. All traffic on this interface is at least single tagged with the so called S-VLAN tag which identifies the U interface. This limits the amount of L2BSA services to 4094 per A10-NSP interface caused by usable VLAN range. Therefore some providers need multiple A10-NSP interfaces if they need to address more than the 4094 services.

The term A10 relates to the end-to-end ADSL network reference model depicted in the figure below. The Core Network reference model is a subset of the end-to-end architecture; it is composed of two functional blocks and three reference points. The Access Node and the Regional Broadband Network are the two functional blocks. U, V and A10 are the three reference points.

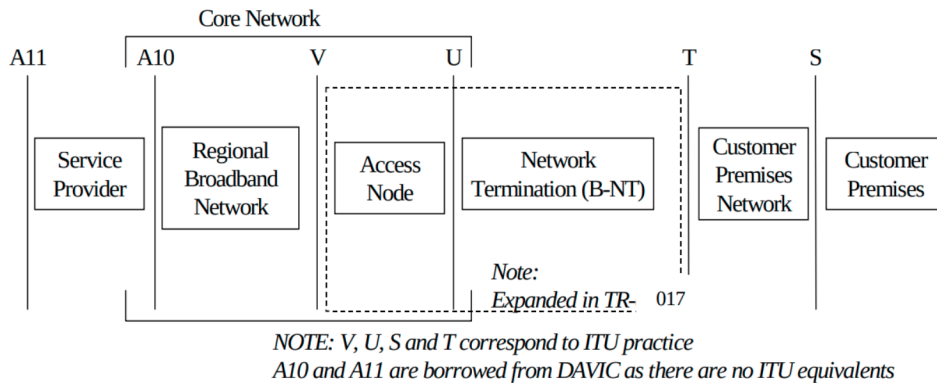


Figure 2. TR-025

Source: <https://www.broadband-forum.org/download/TR-025.pdf>

The mapping between U interface and A10-NSP/S-VLAN is managed by the L2BSA service provider and typically changes frequently triggered by re-provisioning actions. Therefore all PPPoE discovery as well as DHCPv4/v6 packets must be enriched with additional line identification headers (Agent-Remote-Id, Agent-Circuit-Id, Actual-Data-Rate, ...) by the L2BSA service provider in upstream direction (from U to A10-NSP interface) to allow the wholesale provider to identify the actual U interface for traffic received on the A10-NSP interface. This functionality is referred to as the intermediate agent functionality.

The U interface is terminated on the access node which might be an OLT for fiber based access or MSAN for xDSL, therefore the U interface can also be called the customer access line or port. The access node will add an additional VLAN for each access line to uniquely identify them between access node and leaf switch. This VLAN is called the ANP tag (Access-Node-Port).

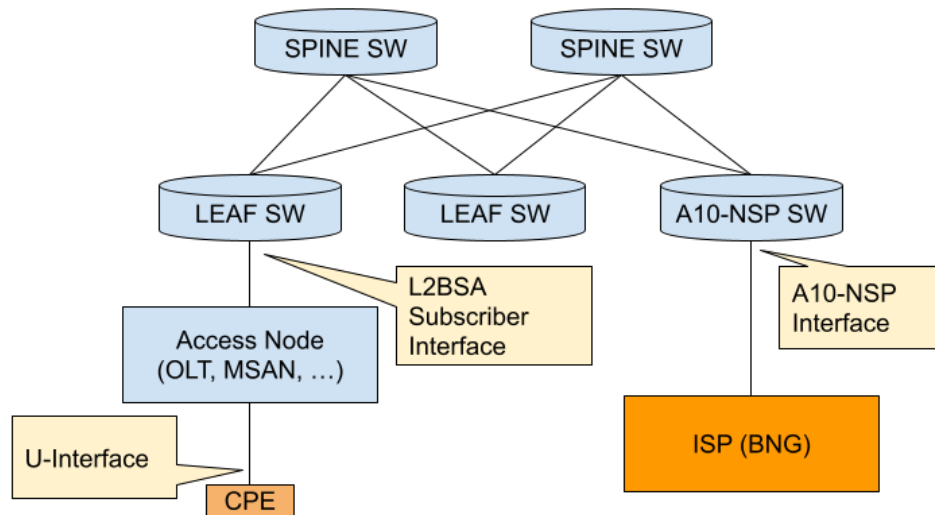


Figure 3. L2BSA Fabric

For L2BSA services all traffic received with a given ANP tag will be transparently forwarded to the corresponding A10-NSP interface via Layer 2 cross-connect (L2X) services. We introduce a new subscriber type for L2BSA in our RBFS access infrastructure to manage those services on the leaf switches and to allow further advanced access services like QoS or subscriber accounting.

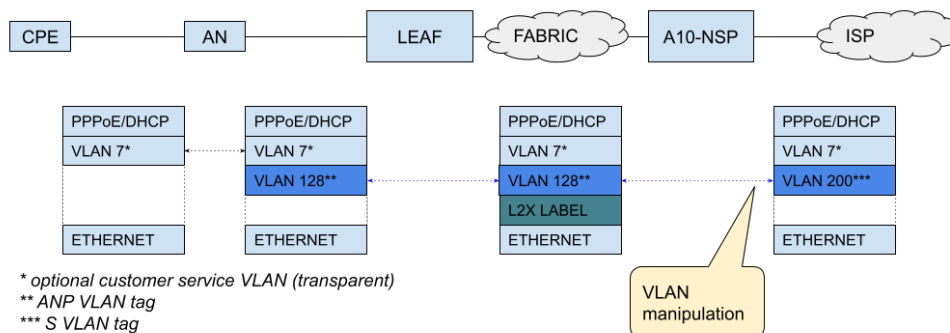


Figure 4. L2BSA Traffic

1.1. L2BSA Services and Subscribers

RBFS distinguishes between L2BSA services and subscribers. L2BSA services are created using the Operational State API (`/l2bsa`) stored as ephemeral objects in the configuration daemon. Those objects are excluded from the actual configuration and not restored after reboot. The L2BSA service represents a defined interface to add, update and delete L2BSA services. Those services are managed by external orchestrators or service managers and are not changed or deleted by RBFS.

Adding such a L2BSA service triggers the creation of a corresponding L2BSA subscriber which represents the internal state. Each L2BSA subscriber is a full access subscriber and therefore most subscriber features will work for L2BSA as well. This includes features like RADIUS authentication, accounting, CoA or dynamic QoS. Each L2BSA subscriber persists as long as the service is still present

also if the subscriber has already been terminated. This allows external applications to collect all states and counters before the subscriber is finally deleted triggered by L2BSA service delete.

1.2. A10-NSP L2X Services

For each L2BSA service there must also be an A10-NSP L2X endpoint created using the Operational State API ([/a10nsp/l2x](#)).

The A10-NSP endpoint is the place where all VLAN manipulation is done meaning where ANP and S-VLAN are swapped.

1.3. Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the Platform Guide for the features and the sub-features that are or are not supported by each platform.

2. L2BSA Configuration

This document assumes a working base configuration explaining the L2BSA specific additions only.

2.1. AAA Profile Configuration

Each L2BSA service refers to a mandatory AAA configuration profile as explained in detail in the Subscriber Management Configuration Guide. The following example shows the minimum configuration with authentication and accounting disabled.

```

supervisor@leaf1: cfg> show config access aaa-profile aaa-l2bsa-default
{
  "rtbrick-config:aaa-profile": [
    {
      "profile-name": "aaa-l2bsa-default",
      "authentication": {
        "order": "NONE"
      },
      "accounting": {
        "order": "NONE"
      }
    }
  ]
}

```

2.2. Intermediate Agent Configuration

The intermediate agent function must be enabled for each physical interface used by L2BSA services.

```

supervisor@leaf1: cfg> set access l2bsa intermediate-agent interface ifp-
0/0/1
  <cr>
  pppoe-enable           Enable/disable the L2BSA intermediate agent for PPPoE
Discovery
  dhcpv4-enable         Enable/disable the L2BSA intermediate agent for
DHCPv4
  dhcpv6-enable         Enable/disable the L2BSA intermediate agent for
DHCPv6
  vlan-mode             L2BSA intermediate agent VLAN mode
  inner-vlan-min        Inner VLAN min
  inner-vlan-max        Inner VLAN max

```


Attribute	Description
pppoe-enable	This option allows to enable/disable the intermediate agent function for PPPoE discovery packets. Default: true (enabled)
dhcpv4-enable	This option allows to enable/disable the intermediate agent function for DHCPv4 packets. Default: true (enabled)
dhcpv6-enable	This option allows to enable/disable the intermediate agent function for DHCPv6 packets. Default: true (enabled)
vlan-mode	This options enables the intermediate agent function for single, double or single and double tagged packets. Default: SINGLE_DOUBLE Values: SINGLE, DOUBLE, SINGLE_DOUBLE
inner-vlan-min/max	This options is applicable for double tagged packets only and allows to limit the intermediate agent function to a given inner VLAN range. Default: all

The following common example enables the intermediate agent for all packets (PPPoE, DHCPv4 and DHCPv6) for all single tagged packets and double tagged packets with inner VLAN between 1 and 3000.

```

supervisor@leaf1: cfg> show config access l2bsa intermediate-agent
{
  "rtbrick-config:intermediate-agent": {
    "interface": [
      {
        "interface-name": "ifp-0/0/1",
        "inner-vlan-min": 1,
        "inner-vlan-max": 3000,
      }
    ]
  }
}

```

3. L2BSA Operational Commands

The following commands allow to list or clear all L2BSA services from CLI.

```

supervisor@leaf1: op> show l2bsa service
Interface      ANP VLAN      Timestamp
hostif-0/0/1   1000         Mon Aug 30 09:21:14 GMT +0000 2021

supervisor@leaf1: op> clear l2bsa service all
Success

```

L2BSA subscribers can be managed like any other subscriber using the subscriber CLI commands.

```

supervisor@leaf1: op> show subscriber
Subscriber-Id      Interface      VLAN      Type      State
281479271677954   hostif-0/0/1   1000:0    L2BSA     ESTABLISHED

supervisor@leaf1: op> show subscriber filter type L2BSA
Subscriber-Id      Interface      VLAN      Type      State
281479271677954   hostif-0/0/1   1000:0    L2BSA     ESTABLISHED

supervisor@leaf1: op> show subscriber 281479271677954 detail
Subscriber-Id: 281479271677954
  Type: L2BSA
  State: ESTABLISHED
  Created: Mon Aug 30 09:23:34 GMT +0000 2021
  Interface: hostif-0/0/1
  Outer VLAN: 1000
  IFL: l2bsa-0/0/1/281479271677954
  Username: hostif-0/0/1:1000@l2bsa
  Agent-Remote-Id: DEU.L2BSA.API01
  Agent-Circuit-Id: 0.0.0.0/0.0.0.0 eth 01
  AAA-Profile: aaa-l2bsa-default
  Session-Timeout: 0 (disabled)
  Idle-Timeout: 0 (disabled)
  L2X:
    Instance: default
    Nexthop: 1.1.1.1 (ipv4/labeled-unicast)
    Ingress Label (TX): 1337
    Egress Label (RX): 7331
  Accounting:
    Session-Id: 281479271677954:1630315414
    Start-Time: 2021-08-30T09:23:34.847166+0000
    Interims Interval: 0 seconds

```

The CLI also provides some global intermediate agent function packet statistics about traffic received in control plane which belong to L2BSA subscribers.

```
supervisor@leaf1: op> show l2bsa intermediate-agent statistics
PPPoE Discovery Sent           : 0
PPPoE Discovery Updated        : 0
PPPoE Discovery Dropped        : 0
```

The sent statistics count all intermediate agent traffic forwarded back to L2X while updated only counts those which are also updated or enriched by access line information. Dropped are those packets received but dropped by L2BSA subscriber state which might be the case if the L2BSA subscriber is still present but in terminating state.

4. L2BSA Operational State API

The API is split into the L2BSA and A10-NSP L2X service API. Details about those API endpoints, schema and attributes can be found in the the official RBFS Operational State OpenAPI documentation.

4.1. L2BSA Services API

Each L2BSA service is uniquely identified by physical interface name (IFP) and ANP VLAN identifier. All other attributes of L2BSA services can be changed dynamically by replacing the existing service object doing a full update meaning that updates must include all attributes and RBFS will automatically recognize the changes to the last version to apply those incrementally.

Endpoint	Description
GET /l2bsa	This endpoint returns a list of all L2BSA services.
PUT /l2bsa	This endpoint replaces all L2BSA services on the system by adding new, updating existing and deleting those not included in the request but present on the system. Therefore this endpoint can be used to delete all services if an empty list is provided in the request body.
GET /l2bsa/{ifp_name}	This endpoint returns a list of all L2BSA services on the given physical interface (IFP).
PUT /l2bsa/{ifp_name}	This endpoint works similar to PUT /l2bsa but affects only services matching the given given physical interface (IFP). Therefore this endpoint can be used to delete all services of a given physical interface if an empty list is provided in the request body.
GET /l2bsa/{ifp_name}/{anp}	This endpoint returns a single L2BSA service or 404 if no matching service is found.
PUT /l2bsa/{ifp_name}/{anp}	This endpoint adds or updates a single L2BSA service.
DELETE /l2bsa/{ifp_name}/{anp}	This endpoint deletes a single L2BSA service.

The following example shows an L2BSA service request body with only the

mandatory attributes set.

```
{
  "ifp_name": "ifp-0/0/1",
  "anp_vlan": 128,
  "aaa_profile_name": "aaa-l2bsa-default",
  "l2x": {
    "ingress_nextthop": "fd00:a4::21",
    "ingress_service_label": 1001,
    "egress_service_label": 1002
  }
}
```

The L2X ingress nextthop is typically the fabric loopback IPv4 or IPv6 address of the corresponding switch where the corresponding A10-NSP L2X service is placed. The ingress label is the upstream service label added for traffic send to the corresponding nextthop (sent label). The egress label is the downstream service label where traffic send from nextthop is expected to be received (receive label). This label must be unique per switch.



Only labels greater than 100.000 should be used for the L2X service labels because labels between 16 - 100.000 are reserved by other applications. L2X service labels must not conflict with other L2X services labels and the same label can be used for downstream and upstream.

L2BSA supports also dynamic QoS which can be controlled by RADIUS if enabled or directly via API as shown in the following example.

```
{
  "ifp_name": "ifp-0/0/1",
  "anp_vlan": 128,
  "aaa_profile_name": "aaa-l2bsa-default",
  "l2x": {
    "ingress_nextthop": "fd00:a4::21",
    "ingress_service_label": 1001,
    "egress_service_label": 1002
  },
  "qos": {
    "qos_profile_name": "l2bsa-qos-default",
    "parent_scheduler": "pon1",
    "shaper":
      "name=shaper_session,high=14000,low=2000;name=shaper_voice,high=2000",
    "policer": "level=4,cir=1m,cbs=256;level=3,cir=2m,cbs=512",
    "queue": "name=BestEffort,size=65000;name=Voice,size=20000"
  }
}
```

The QoS configuration profile is used to instantiate the QoS resources for the subscriber including schedulers, queues and shapers. This profile can be set using

the `qos_profile_name` attribute. All dynamic QoS settings like MFC, queue sizes, shaper and policer rates will be reset if this attribute is present also if not changed. Additional shaper or policer settings included will be applied to the new QoS configuration after reset.

The assigned shaper instances can be updated using the `shaper` attribute which will apply to the QoS instance of the corresponding subscriber only, but not to the other subscribers using the same QoS profile. It is only possible to update existing shapers dynamically but it is not possible to create a new shaper.

```
"shaper": "<shaper-name>, <high-kbps>, <low-kbps>; <shaper-name>, ..."
```

This attribute can be also used as a key-value list which is automatically recognized by RBFS.

```
"shaper": "name=<shaper-name>, high=<high-kbps>, low=<low-kbps>; ..."
```

The `policer` attribute is also a string which contains a list of multiple policer level settings separated by semicolon. Each setting contains a level, cir, cbs, pir, pbs, max-cir and max-pir separated by comma.

```
"policer": "<level>, <cir>, <cbs>, <pir>, <pbs>, <max-cir>, <max-pir>; <level>..."
```

Example:

```
"policer": "1,2000,200;2,8000,800;3,0,800;4,0,800"
```

level: 1 (highest priority) to 4 (lowest priority)

cir: ingress policer committed information rate (kbps)

cbs: ingress policer committed burst size (kbps)

pir: ingress policer peak information rate (kbps)

pbs: ingress policer peak burst size (kbps)

max-cir: max ingress policer committed information rate (kbps)

max-pir: max ingress policer peak information rate (kbps)

If PIR and PBS are not defined, the values from CIR and CBS are used as PIR and PBS as well. The max CIR and max PIR attributes are optional default set to unlimited.

This attribute can be also used as a key-value list which is automatically recognized by RBFS.

```
"policer": "level=<level>, cir=<cir>, cbs=<cbs>, pir=<pir>, pbs=<pbs>, max-cir=<max-cir>, max-pir=<max-pir>; ..."
```

Example:

```
"policer": "level=4, cir=1m, cbs=256; cir=2m, cbs=512, level=3"
```

The **queue** attribute is a string which contains a list of multiple queue settings separated by semicolon. Each queue setting contains a queue name and queue size in bytes separated by comma.

```
"queue" : "<queue-name>,<size-bytes>;<queue-name>,<size-bytes>;..."
```

This attribute can be also used as a key-value list which is automatically recognized by RBFS.

```
"queue" : "name=<queue-name>,size=<size-bytes>;name=..."
```



The subscriber management infrastructure does not check if a referenced QoS profile, shaper, queue or policer exists or not. This is handled by forwarding infrastructure which continues processing the subscriber QoS settings as soon as the missing resource was added.

The parent scheduler element of the scheduler-map assigned to the subscriber can be selected with the **parent_scheduler** attribute. If not present, the scheduler-map will be directly bound to the local IFP where the session is established. The parent scheduler can be only set at the beginning and must not be changed.



Providing a QoS parent scheduler which is not present on the corresponding IFP will lead to black howling of all egress data traffic.

The optional **access_line_info** attribute contains the access line identification and characteristics attributes used by the intermediate agent function. A detailed list of all supported line attributes and values can be found in the the official RBFS Operational State OpenAPI documentation.



For access line attributes there is a difference between attributes not present or set to zero. Those not present will be also not added by intermediate agent function where those set to zero will be added by intermediate agent function with value also set to zero.

Example:

```

/usr/bin/curl --location --request PUT
'http://10.0.0.1:19091/api/v1/rbfs/elements/leaf1/services/opsd/proxy/l2bsa/i
fp-0%2f0%2f1/128' --header 'Content-Type: application/json' --data-raw '{
  "ifp_name": "ifp-0/0/1",
  "anp_vlan": 128,
  "aaa_profile_name": "aaa-l2bsa-default",
  "l2x": {
    "ingress_nextthop": "fd00:a4::21",
    "ingress_service_label": 1001,
    "egress_service_label": 1002
  },
  "qos": {
    "qos_profile_name": "l2bsa-qos-default",
    "parent_scheduler": "pon1",
    "shaper":
"name=shaper_session,high=14000,low=2000;name=shaper_voice,high=2000",
    "policer": "level=4,cir=1m,cbs=128;level=2,cir=1m,cbs=128",
    "queue": "name=BestEffort,size=65000;name=Voice,size=20000"
  },
  "access_line_info": {
    "agent_circuit_id": "0.0.0.0/0.0.0.0 eth 0/0",
    "agent_remote_id": "DEU.RTBRICK.L2BSA01",
    "upstream": {
      "actual_rate": 2048
    },
    "downstream": {
      "actual_rate": 16235
    },
    "pon_type": "GPON"
  }
}'

```

4.2. L2BSA Subscribers API

The existing subscribers API was extended to do some common actions using the L2BSA physical interface name (IFP) and ANP VLAN identifier instead of the subscriber-id.

Endpoint	Description
GET /subscribers/l2bsa/{ifp_name}/{ anp}	This endpoint is an alias for GET /subscribers/{subscriber_id} returning detailed subscriber information or 404 if no matching subscriber is found.
DELETE /subscribers/l2bsa/{ifp_name}/{ anp}	This endpoint is an alias for DELETE /subscribers/{subscriber_id} which terminates a matching L2BSA subscriber.

Endpoint	Description
GET /subscribers/l2bsa/{ifp_name}/{anp}/adjusted-accounting	This endpoint is an alias for GET /subscribers/{subscriber_id}/adjusted-accounting returning the adjusted accounting information or 404 if no matching subscriber is found.

4.3. A10-NSP L2X Services API

Endpoint	Description
GET /a10nsp/l2x	This endpoint returns a list of all A10-NSP L2X services.
PUT /a10nsp/l2x	This endpoint replaces all A10-NSP L2X services on the system by adding new, updating existing and deleting those not included in the request but present on the system. Therefore this endpoint can be used to delete all services if an empty list is provided in the request body.
GET /a10nsp/l2x/{lag_interface_name}	This endpoint returns a list of all A10-NSP L2X services on the given LAG interface.
PUT /a10nsp/l2x/{lag_interface_name}	This endpoint works similar to PUT /a10nsp/l2x but affects only services matching the given LAG interface. Therefore this endpoint can be used to delete all services of a given LAG interface if an empty list is provided in the request body.
GET /a10nsp/l2x/{lag_interface_name}/{s_anp}	This endpoint returns a single A10-NSP L2X service or 404 if no matching service is found.
PUT /a10nsp/l2x/{lag_interface_name}/{s_anp}	This endpoint adds or updates a single A10-NSP L2X service.
DELETE /a10nsp/l2x/{lag_interface_name}/{s_anp}	This endpoint deletes a single A10-NSP L2X service.

The following example shows an A10-NSP L2X service request body with only the mandatory attributes set.

```
{
  "lag_interface_name": "lag-1",
  "s_vlan": 100,
  "anp_vlan": 128,
  "l2x": {
    "ingress_nexthop": "fd00:a4::22",
    "ingress_service_label": 1002,
    "egress_service_label": 1001
  }
}
```

The L2X ingress nexthop is typically the fabric loopback IPv4 or IPv6 address of the corresponding switch where the corresponding L2BSA service is placed. The ingress label is the upstream service label added for traffic send to the corresponding nexthop (send label). The egress label is the downstream service label where traffic send from nexthop is expected to be received (receive label). This label must be unique per switch.



Only labels greater than 100000 should be used for the L2BSA service labels because labels between 16 - 100000 are reserved by other services. L2BSA service labels must not conflict with other L2X services labels and the same label can be used for downstream and upstream.

Example:

```
/usr/bin/curl --location --request PUT
'http://10.0.0.1:19091/api/v1/rbfs/elements/a10nsp-
sw1/services/opspd/proxy/a10nsp/l2x/lag-1/100' \
--header 'Content-Type: application/json' \
--data-raw '{
  "lag_interface_name": "lag-1",
  "s_vlan": 100,
  "anp_vlan": 128,
  "l2x": {
    "ingress_nexthop": "fd00:a4::22",
    "ingress_service_label": 1002,
    "egress_service_label": 1001
  }
}'
```



Currently, L2BSA is supported only for the PPPoE subscribers. The support for the IpoE (DHCPv4/DHCPv6) subscribers will be available in the future releases.