L3DL

Three Drafts from LSVR WG

draft-ietf-lsvr-l3dl

draft-ietf-lsvr-l3dl-signing

draft-ietf-lsvr-l3dl-ulpc

IDR Interim 2021.10.18

randy@psg.com, sra@hactrn.net, keyur@arrcus.com, housley@vigilsec.com
Primary Goal (2018)

Layer 3 Topology Discovery and Liveless for LSVR / BGP-SPF
Find Neighbor(s)

Learn L3 P2P Addresses

Configure BGP/SPF
L3DL
Layer 3 Discovery & Liveness
draft-ietf-lsvr-l3dl

IDR Interim
2021.10.18
randy@psg.com, sra@hactrn.net, keyur@arrcus.com
L2 Discover L3 Attrs

Device

BGP-SPF

Link Check AFI/SAFIs

Ether PDUs

Device

BGP-SPF

Link Check AFI/SAFIs

Ether PDUs

Device

BGP-SPF

Link Check AFI/SAFIs

Ether PDUs
This is NOT a Routing Protocol

Discovers the Layer 3 Addresses on a PointToPoint Link
Ethernet Frame

Datagram

PDU

Ethernet Frame

Datagram

PDU

Ethernet Frame

Datagram

PDU

Ethernet Frame

Datagram

PDU

Ethernet Frame

Datagram

PDU
Layer 2 Transport can handle $2^{32}$ octet PDUs

$2^{24}$ Datagrams (in Ethernet Frames)
$2^{16}$ Octets/Datagram (except it is a Frame)
Big PDUs Over Ethernet

Jörg Ott did a Very Helpful Transport Directorate Early Review
Why not TCP?

• When this runs, there are no IP Addresses
• This protocol is to Learn L3 Addresses
• So it is a cheap TCP-like protocol
• Reassembly of out of order Datagrams
• Retransmission with Back-off
• PDUs are ACKnowledged
• Long Lived Sessions
• ...

2021.10.18 L3DL
Fully Stateful Session Per Peer

Graceful Restart

State May Be Resumed à la BGP
OPEN PDU

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

PDU Type = 1
Payload Length

Nonce

LLEI Length
My LLEI

AttrCount

Attribute List ...
Auth Type
Key Length

Key ...

Serial Number

Sig Type
Signature Length
Signature ...

Creative Commons: Attribution & Share Alike
Encaps etc PDUs

<table>
<thead>
<tr>
<th>PDU Type</th>
<th>Payload Length</th>
<th>~</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial Number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encapsulation List...</td>
<td>Sig Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Signature Length</td>
<td>Signature ...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Announce/Withdraw

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--------------------------------------------------+
| PDU Type = 4 | Payload Length |
+--------------------------------------------------+
| Count |
+--------------------------------------------------+
| Serial Number |
+--------------------------------------------------+
| IPv4 Address |
+--------------------------------------------------+
| PrefixLen |
| more ... |
| Sig Type |
+--------------------------------------------------+
| Signature Length |
| Signature ... |
+--------------------------------------------------+

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------------------------------------+
| Ann/With |
| Primary |
| Under/Over |
| Loopback |
| Reserved .. |
+---------------------------------------------+
Meant to Support BGP/SPF in DataCenter

I.e. Simple Topology so no Multicast Storms
We have Two Implementations
One Python3 (LSOE)
One in Golang
L3DL-Signing
Layer 3 Discovery and Liveness Signing
draft-ietf-lsvr-l3dl-signing

IDR Interim
2021.10.17

randy@psg.com, sra@hactrn.net, housley@vigilsec.com
OPEN PDU

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

<table>
<thead>
<tr>
<th>PDU Type = 1</th>
<th>Payload Length</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Nonce</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>LLEI Length</th>
<th>My LLEI</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>AttrCount</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Attribute List ...</th>
<th>Auth Type</th>
<th>Key Length</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Key ...</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Serial Number</th>
</tr>
</thead>
</table>

| Sig Type | Signature Length | Signature ...
|----------|------------------|-------------|

Creative Commons: Attribution & Share Alike
PDU Sender Signing

- The Key in the OPEN PDU SHOULD be the public key of an asymmetric key pair.
- The sender signs with the private key, of course.
- The device sending the OPEN may use one key for all links, a different key for each link, or some aggregation(s) thereof.
Two Flavors

Trust on First Use (TOFU)

PKI Based
Trust on First Use (TOFU)

- The OPEN key is generated on the sending device
- It is believed without question by the receiver
- Used to verify all subsequent PDUs from the same sender with the same Key Type
PKI-Based Keying

- An enrollment step is performed
- The public key is put into a certificate, which is signed by the operational environment's trust anchor
- The relying party can be confident that the public key is under control of the identified L3DL protocol entity
Do Not Be Afraid
This is NOT X.509

• These need not be X.509 certificates
• X.509 is much more complicated than we need
• They are just signatures of one key (the session key supplied in the Key field of the OPEN PDU) by another key (the trust anchor)
• Every device must have TA burned in
Verify is the Same

- The two methods are indistinguishable
- The key provided in the OPEN PDU is used to verify the signatures of subsequent PDUs
- The difference that PKI-based keys may be verified against the trust anchor when the OPEN PDU is received
The Choice of Which Keying is Left to the Operator
L3DL-ULPC
Upper Layer Protocol Configuration
draft-ietf-lsvr-l3dl-ulpc

IDR Interim
2021.10.28

randy@psg.com, sra@hactrn.net, keyur@arrcus.com
Meant to Allow Config of Arbitrary L3+ Protocols

So Far Only Defined for BGP as BGP needed for BGP/SPF in DataCenter
L3DL PDU for ULPC

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----------------------------------------------+-------------------+
|                    Type = 9                    |
|                                  Payload Length |
+-----------------------------------------------+-------------------+
~                     |          ULPC Type   |          AttrCount |
~                     |          Attribute List ...  |          Sig Type   |          Signature Len |
~                     |          Signature ...               |
Provide the minimal set of configuration parameters for BGP OPEN to succeed.
Not to replace or conflict with data exchanged by BGP OPEN
Multiple sources of truth are a recipe for complexity and pain
AS and Peering IP

0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Attr Type = 1 | Attr Len = 48 | My ASN ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Attr Type = 2 | Attr Len = 56 | My IPv4 Peering Address ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

Prefix Len
Auth Data and GSTM

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++
| Attr Type = 4 |
+++
| Attr Len |
~
BGP Authentication Data ...
~

The only Flag Currently Defined is GSTM
Yes, there is one for IPv6 😊
Arbitrary Attributes

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

<table>
<thead>
<tr>
<th>Type = 9</th>
<th>Payload Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ULPC Type</td>
<td>AttrCount</td>
</tr>
<tr>
<td>Attribute List ...</td>
<td>Sig Type</td>
</tr>
<tr>
<td>Signature ...</td>
<td></td>
</tr>
</tbody>
</table>

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

<table>
<thead>
<tr>
<th>Attr Type</th>
<th>Attr Len</th>
<th>Payload</th>
</tr>
</thead>
</table>

2021.10.18 L3DL

Creative Commons: Attribution & Share Alike
And the Base L3DL Protocol
Allowed and Marked Loopbacks etc.
That's It
We would want to hack to better fit IDR

e.g. Massive PDUs for Hundreds of Address Encapsulations Not Needed
That’s Really It 😊