

IEEE P802.1CS Link-local Registration Protocol Draft 1.3 introduction

Norman Finn
Huawei Technologies Co. Ltd
cs-finn-D1-3-introduction-0318-v01

P802.1CS Clause changes

Draft 1.2	Draft 1.3
Clauses 1-5	Same clauses 1-5
6 PDU formats	6 LRP introduction
7 LRP	7 procedures & state machines
	8 PDU formats
8 App Interface	9 App interface
9 Managed objects	10 Managed objects
	11 YANG
	12 MIB

Clarity added by settling “Issues” in final comment disposition on D1.2.

- An “LRP-DT instance” and a “Portal” are both now a collection of specific state machines, and well-defined.
- Figures were added in 1.6, 6.2, 6.5.2, 6.6.2, and 6.7, on top of original figure in 6.2 (was 7.1.1) to clarify the basic operation of LRP. (We still need at least one more, showing proxy relationships.)
- Much text in the form of an approximate description expressing the editor’s intentions (e.g. the BNF description of PDU formats and the instance/portal creation) has been either:
 - Shortened and made a true introduction; or
 - Expanded into a real specification.

Issues that came up in implementing the comment resolutions

- Configured LRP-DT instances and portals vs. creation via LLDP
- Link-local vs. routable IP addresses are a problem.
- TCP passive OPEN
- Demultiplexing conversations on LRP-DT instance by appld.
- Editor's pleas for help.

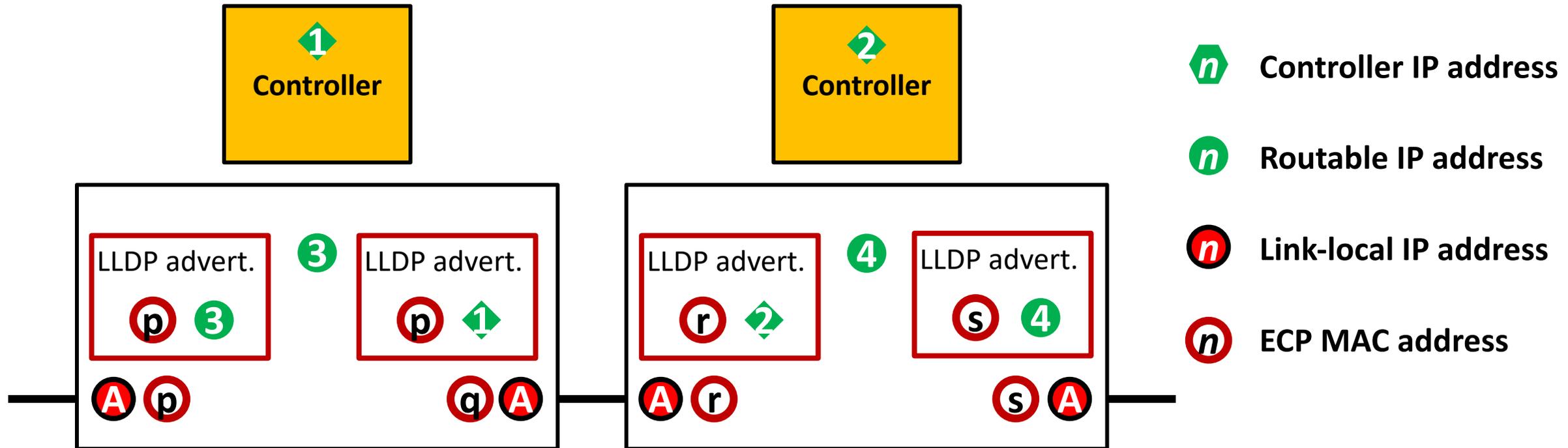
Configured LLDP instances

- LLDP is great for making automatic connections. But, what if you don't want to make connections automatically. (Also, how do you configure an LRP proxy system [central controller]?)
- Added a managed object (IrpGlobLldpInstanceList) that constructs a fake LLDP instance that can drive the standard state machines.
- Also added an “Enable fixed Portal request” primitive so that an application can give this item to LRP.

Link-local vs. Routable IP addresses

- Resolution said, “increase LLDP TLV size to accommodate **both** IPv4 and IPv6 addresses”, instead of one or the other, as in D1.2.
- Adding link-local addresses could double that. Too big!
- Anyway, link-local IP addresses are not invented by LRP, and LRP doesn't tell how to find them.
- Also, we only want IPv4 and IPv6, not all L3 protocols.
- So, Link-local IP addresses have been handled by:
 - Changing to an LRP-only “address type”
 - Addresses in LLDP TLV are always routable, and address type can say, “link-local allowed, also.” Finding link-local address is not specified in P802.1CS.

Proxy/served systems



- Link-local IP address advertised with a Boolean in LLDP, not IP address.
- Can advertise system routable IP address or controller IP address.
- Preference order: 1: routable IP (v6 > v4). 2: link-local IP. 3: ECP MAC

TCP passive OPEN

- LRP-DT instance creation supports the usual TCP “passive OPEN” request. The state machines start a passive OPEN when it’s needed to field SYN messages to a routable IP address, and when the SYN comes in, the connection creates (or is given to) the right LRP-DT instance.

Demuxing LRP-DT connection by appld

- We concluded in Geneva that we could demultiplex the LRP-DT connection in the LRP-DT instance by looking at the appld.
- We forgot that, if a routable IP address is used, there may be LRP-DT connections for the same appld for multiple ports, on a single connection, and hence, a single LRP-DT instance.
- Adding the LLDP Chassis ID and Port ID to every LRPDU makes them too large.
- So, D1.3 has a “My Port Number” that is exchanged in the Hello LRPDUs. LRPDUs are demuxed by {appld, my port #, your port #}.

Editor can use some help

Especially for:

- Who makes the TCP connection (active OPEN) and who receives it (passive OPEN)? (see 7.2.1)
 - "Lowest address" may or may not work. Does Network Address Translation mess this up?
- Checksums (Paul Bottorff) See Z.1 for pros/cons.
 - Do we use CRC-32 or a more advanced checksum on each record?
- Counter variables on ephemeral structures. (see 10.2)

Thank you