P802.1CS-2020/Cor1/D2.0
October 13, 2023
(Corrigendum to IEEE Std 802.1CS™–2020)

Draft Standard for Local and metropolitan area networks—

Link-local Registration Protocol

Corrigendum 1: Corrections to Management Modules and Protocol Encoding

Unapproved draft, prepared by the
Time-Sensitive Networking (TSN) Task Group of IEEE 802.1

Sponsored by the
LAN/MAN Standards Committee of the IEEE Computer Society

This page and the following cover pages are not part of the draft. They provide revision and other information for IEEE 802.1 Working Group members and partipants in the IEEE Standards Association ballot process, and will be updated as convenient. Blank pages allow for the future addition of cross-references to changed text without forcing renumbering of all pages in the draft. Pages are numbered from 1 (including cover pages) for the convenience of reviewers whose PDF viewers do not easily accommodate different numbering sequences. Pages will of course be renumbered prior to publication as a standard.

The text proper of this draft begins with the Title page.

Draft status

This draft P802.1CS-2020/Cor1/D2.0 is being issued for initial Standards Association ballot. The 802.1 Working Group ballot that preceded this Standards Association ballot closed with no Disapprove ballots and no comments. The text of the draft (including page numbering) is unchanged from the balloted draft, with the exception of updates to these cover pages (not part of this draft) and the consequent update of the date shown in the header of most pages.

Information in the cover letter for this ballot is repeated below.

Editorial suggestions

Please note, that professional editing takes place once the document is approved, and as such, required editorial changes will be reviewed/made then (e.g., punctuation, grammar, formatting, style consistency). The following editorial suggestions will also be considered as part of this initial ballot: << >>.

MIB and YANG modules

The YANG module is attached in plain text (UTF-8) format to the draft pdf. Also included is a txt file showing the differences between the YANG module in the base document IEEE Std 802.1CS-2020, and the replacement YANG module specified in this corrigendum draft.

This page intentionally left blank.

This page intentionally left blank.

This page intentionally left blank.

This page intentionally left blank.

This page intentionally left blank.

P802.1CS-2020/Cor1/D2.0
October 13, 2023

(Corrigendum to IEEE Std 802.1CS™–2020)

Draft Standard for Local and metropolitan area networks—

Link-local Registration Protocol

Corrigendum 1: Corrections to Management Modules and Protocol Encoding

Unapproved draft, prepared by the

Time-Sensitive Networking (TSN) Task Group of IEEE 802.1

Sponsored by the

LAN/MAN Standards Committee of the IEEE Computer Society

IEEE Standards Activities Department
445 Hoes Lane
Piscataway, NJ 08854, USA

Abstract: Corrects errors in the YANG module, SNMP MIB and TLV encoding in IEEE Std 802.1CS-2020.

Keywords: Bridged Local Area Networks, bridges, bridging, IEEE 802®, IEEE 802.1CS™, IEEE 802.1Q™, Link-local Registration Protocol, local area networks (LANs), LRP, MAC Bridges, Time-Sensitive Networking, TSN, Virtual Bridged Local Area Networks (virtual LANs).

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Print: ISBN 978-X-XXX-XXX-X STDXXXXX
PDF: ISBN 978-X-XXX-XXX-X STDPDXXXXX

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (https://standards.ieee.org/ipr/disclaimers.html), appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents."

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA, and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE standards documents are supplied "AS IS" and "WITH ALL FAULTS."

Use of an IEEE standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents**.

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and Standards Coordinating Committees are not able to provide an instant response to comments, or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the IEEE SA myProject system.[1] An IEEE Account is needed to access the application.

Comments on standards should be submitted using the Contact Us form.[2]

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

1. Available at: https://development.standards.ieee.org/myproject-web/public/view.html#landing.
2. Available at: https://standards.ieee.org/content/ieee-standards/en/about/contact/index.html.

Copyrights

Photocopies

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit IEEE Xplore or contact IEEE.[3] For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE SA Website.[4] Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in IEEE Xplore. Users are encouraged to periodically check for errata.

Patents

IEEE Standards are developed in compliance with the IEEE SA Patent Policy.[5]

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the

3. Available at: https://ieeexplore.ieee.org/browse/standards/collection/ieee.
4. Available at: https://standards.ieee.org/standard/index.html.
5. Available at: https://standards.ieee.org/about/sasb/patcom/materials.html.

IEEE SA Website at http://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. IEEE Standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

Participants

<<The following lists will be updated in the usual way prior to publication>>

At the time this standard was completed, the IEEE 802.1 working group had the following membership:

Glenn Parsons, *Chair*
Jessy Rouyer, *Vice Chair*
János Farkas, *TSN Task Group Chair*
Norman Finn, *Editor*

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

A.N. Other

<<The above lists will be updated in the usual way prior to publication>>

When the IEEE-SA Standards Board approved this standard on <dd> <month> <year>, it had the following membership:

Chair
Vice-Chair
Past Chair
Secretary

*Member Emeritus

<<The above lists will be updated in the usual way prior to publication>>

Introduction

This introduction is not part of IEEE Std 802.1CS-2020/Cor1, IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks—Corrigendum 1: Corrections to Management Modules and Protocol Encoding

This corrigendum to IEEE Std 802.1CS-2020 corrects errors in the YANG module, SNMP MIB and TLV encoding.

Contents

Tables

Draft Standard for Local and Metropolitan Networks —

Link-local Registration Protocol

Corrigendum 1: Corrections to Management Modules and Protocol Encoding

(Corrigendum to IEEE Std 802.1CS™–2020)

NOTE—The editing instructions contained in this corrigendum define how to merge the material contained therein into the existing base standard and its amendments to form the comprehensive standard.

The editing instructions are shown in ***bold italics***. Four editing instructions are used: change, delete, insert, and replace. ***Change*** is used to make corrections in existing text or tables. The editing instruction specifies the location of the change and describes what is being changed by using ~~strikethrough~~ (to remove old material) and underscore (to add new material). ***Delete*** removes existing material. ***Insert*** adds new material without disturbing the existing material. Deletions and insertions may require renumbering. If so, renumbering instructions are given in the editing instruction. ***Replace*** is used to make changes in figures or equations by removing the existing figure or equation and replacing it with a new one. Editing instructions, change markings, and this note will not be carried over into future editions because the changes will be incorporated into the base standard.

Change ABSTRACT as follows:

Abstract: This standard specifies protocols, procedures, and managed objects for a Link-local Registration Protocol (LRP) to replicate a registration database from one end to the other of a point-to-point link and to replicate changes to parts of that database. A facility is provided to purge the replicated database if the source becomes unresponsive. LRP is optimized for databases on the order of ~~1 Mbyte~~a million bytes.

1. Overview

1.1 Scope

Change 1.1 as follows:

This standard specifies protocols, procedures, and managed objects for a Link-local Registration Protocol (LRP) to replicate a registration database from one end to the other of a point-to-point link and to replicate changes to parts of that database. A facility is provided to purge the replicated database if the source becomes unresponsive. Provision is made for a proxy system to operate LRP on behalf of a controlled system. LRP is optimized for databases on the order of ~~1 Mbyte~~a million bytes.

6. Link-local Registration Protocol

6.3 Objectives and non-objectives

6.3.1 Objectives of LRP

Change 6.3.1 as follows:

The objectives of LRP are to:

a) Serve application instances as described in 6.2.

b) Remain independent of the syntax and semantics of the LRP application data contained in the applicant and registrar databases.

c) Provide a facility for an application instance to discover its peers, and to make and break associations between application instances.

d) Support multiple options for the transport mechanisms used to carry LRPDUs, in order to provide a range of capability/complexity trade-offs.

e) Transfer the application instances' data quickly, reliably, and efficiently with respect to bandwidth.

f) Efficiently transfer changes to parts of a database, without retransmitting the whole database.

g) Serve applicant and registrar databases whose size, per port, is on the order of magnitude of ~~1 Mbyte~~a million bytes.

h) Support only point-to-point associations between application instances.

i) Support multiple point-to-point associations on one port.

j) By the use of TCP as an LRP-DT transport mechanism, provide for placing the application instance and Portal in a separate Proxy system from its target port, in order to facilitate central control of a network.

k) Support proxying for a Controlled system without requiring any new behaviors of the Controlled system, in order to make it possible to deploy a new LRP application without implementing an application instance in every system.

l) Make efficient use of TCP connections.

m) Support resource-constrained end systems.

12. YANG models for LRP

12.2 The YANG framework

Change 12.2 as follows:

This clause has been developed according to the YANG guidelines published in RFC6087 ([D12]) as applicable to IEEE standards. The YANG framework applies hierarchy in the following areas:

1) The uniform resource name (URN), as specified in IEEE Std 802d. ~~The structure of the URN is such that "ieee" is the root (i.e., name-space identifier), followed by the standard, then the working group developing the standard.~~
2) The YANG objects form a hierarchy of configuration and operational data structures that define the YANG model. These hierarchical relationships are described in 11.2 and 12.5.

12.6 Definition of LRP YANG module

Delete the YANG module in 12.6:

Insert the following YANG module in 12.6:

```
module ieee802-dot1cs-lrp {
  yang-version 1.1;

  namespace "urn:ieee:std:802.1CS:yang:ieee802-dot1cs-lrp";
  prefix "dot1cs";

  import ietf-system { prefix "sys"; }
  import ietf-yang-types { prefix "yang"; }
  import ietf-interfaces { prefix if; }
  import ieee802-types { prefix ieee; }
  import ietf-inet-types { prefix inet; }

  organization
    "Institute of Electrical and Electronics Engineers";
  contact
    "WG-URL: http://ieee802.org/1/
     WG-EMail: stds-802-1-l@ieee.org
       Contact: IEEE 802.1 Working Group Chair
       Postal: C/O IEEE 802.1 Working Group
       IEEE Standards Association
            445 Hoes Lane
            Piscataway
            NJ 08854
            USA

    E-mail: stds-802-1-chairs@ieee.org";

  description
    "This module provides management of systems that support the
    Link-local Registration Protocol (LRP).

    Copyright (C) IEEE (2023).

    This version of this YANG module is part of IEEE Std 802.1CS-2020/Cor1;
    see the standard itself for full legal notices.";

  revision 2023-05-16 {
    description
      "Published as part of IEEE Std 802.1CS-2020/Cor1.

      The following reference statement identifies each referenced
      IEEE Standard as updated by applicable amendments.";

    reference
      "IEEE Std 802.1CS:
```

```
    IEEE Std 802.1CS-2020 Link-local Registration Protocol,
    IEEE Std 802.1CS-2020 Corrigendum 1: Corrections to
    Management Modules and Protocol Encoding.";
}

 revision 2020-12-03 {
  description
    "First defined in IEEE P802.1CS-2020";
  reference
    "IEEE Std 802.1CS-2020.";
}
/*--------------------*/
/* Feature            */
/*--------------------*/

feature lrp {
  description
    "Feature Link-local Registration Protocol";
}

/*--------------------*/
/* Type Definitions   */
/*--------------------*/
typedef lrp-dt-address-union {
  type union {
    type ieee:mac-address;
    type inet:ipv4-address;
    type inet:ipv6-address;
  }
}

/*--------------------*/
/* Configuration Data */
/*--------------------*/
/*
  Link-local Registration Protocol
*/
augment "/sys:system" {
  description "Link-local Registration Protocol";
  container lrp {
    if-feature lrp;
    description
      "Configure the Link-local Registration Protocol";
      leaf lrp-ack-timer-init {
        type uint32;
        units "milliseconds";
        config false;
        description
          "A read-only integer n specifying the number of milliseconds for
          ackTimerInit (D.2.12.6 of IEEE Std 802.1Q-2018)";
        reference
          "11.3.1 of IEEE Std 802.1CS";
      }
      leaf lrp-reconnect-max {
        type uint16;
        units "seconds";
        description
          "An integer number of seconds which is the maximum value for
          instReconnectReset.";
        reference
          "11.3.2 of IEEE Std 802.1CS";
      }
      list portal {
      key "portal-id";
      config false;
      leaf portal-id {
        type uint32;
        config false;
        description
          "Local Identifier of portal";
        reference
          "Clause 10 of IEEE Std 802.1CS";
```

```
        }
        leaf target-port-interface-ref {
          type if:interface-ref;
          config false;
          description
            "The interface reference identifying the target
            port to which this portal is attached to.";
          reference
            "8.2.2.1 of IEEE Std 802.1CS";
        }
        leaf lrp-dt-instance-id {
          type leafref {
            path "/sys:system/dot1cs:lrp/dot1cs:lrp-dt-instance/dot1cs:instance-id";
          }
          config false;
          description
            "The LRP-DT instance id that this portal is
            attached to.";
          reference
            "8.2.2.1 of IEEE Std 802.1CS";
        }
        leaf application-id {
          type string;
          config false;
          description
            "The value transmitted in, and expected to be received in, the
            appId field of Hello LRPDUs.";
          reference
            "8.2.2.1 of IEEE Std 802.1CS";
        }
        leaf my-chassis-id {
          type ieee:chassis-id-type ;
          config false;
          description
            "The value transmitted in the My Chassis ID TLV, and expected to be
            received in the Neighbor Chassis ID TLV, of Hello LRPDUs";
          reference
            "8.2.2.2 of IEEE Std 802.1CS";
        }
        leaf my-port-id {
          type ieee:port-id-type ;
          config false;
          description
            "The value transmitted in the My Port ID TLV, and expected to be
            received in the Neighbor Port ID TLV, of Hello LRPDUs.";
          reference
            "8.2.2.3 of IEEE Std 802.1CS";
        }
        leaf neighbor-chassis-id {
          type ieee:chassis-id-type ;
          config false;
          description
            "The value transmitted in the Neighbor Chassis ID TLV, and expected
            to be received in the My Chassis ID TLV, of Hello LRPDUs.";
          reference
            "8.2.2.4 of IEEE Std 802.1CS";
        }
        leaf neighbor-port-id {
          type ieee:port-id-type ;
          config false;
          description
            "The value transmitted in the Neighbor Port ID TLV, and expected to
            be received in the My Port ID TLV, of Hello LRPDUs.";
          reference
            "8.2.2.5 of IEEE Std 802.1CS";
        }
        leaf my-hello-status {
          type enumeration {
            enum hs-looking {
              value 1;
              description
                "This Portal has not yet received a successful Associate Portal
```

```
            request.";
          }
          enum hs-connecting {
            value 2;
            description
              "This Portal has received a successful Associate Portal
              request, and a Hello LRPDU with the hsLooking status. The
              Portal is ready to receive all LRPDUs.";
          }
          enum hs-connected {
            value 3;
            description
              "This Portal is up and ready to transfer LRP application data.
              The Portal is allowed to transmit all LRPDUs.";
          }
        }
        config false;
        description
          "An enumerated value to be transmitted in the Hello status field
          of any Hello LRPDU.";
        reference
          "8.2.2.8 of IEEE Std 802.1CS";
      }
      leaf local-overflow {
        type boolean;
        config false;
        description
          "Contains the last Boolean input from the Database overflow
          request. A value of TRUE indicates that the partner applicant
          database has exceeded the capacity of the local registrar LRP
          application.";
        reference
          "8.2.2.10 of IEEE Std 802.1CS";
      }
      leaf neighbor-overflow {
        type boolean;
        config false;
        description
          "A Boolean copied from the last-received database overflow bit [bit
          8] in the Error status field of the last-received Hello LRPDU.";
        reference
          "8.2.2.11 of IEEE Std 802.1CS";
      }
      leaf neighbor-acknowledged {
        type boolean;
        config false;
        description
          "A Boolean, equal to the AND of all of the actAcknowledged
          variables for all of the Applicant state machines (records) on this
          Portal.";
        reference
          "8.2.2.12 of IEEE Std 802.1CS";
      }
      leaf my-app-hello-info {
        type string;
        config false;
        description
          "Value to put in the Application Information TLV of a
          transmitted Hello LRPDU.";
        reference
          "8.2.2.16 of IEEE Std 802.1CS";
      }
      leaf last-received-status {
        type string;
        config false;
        description
          "Used by the Receive Hello state machine to record the
          Hello status field of a Hello LRPDU received from the
          neighbor Portal.";
        reference
          "8.2.2.22 of IEEE Std 802.1CS";
      }
```

```
        leaf applicant-active-records {
          type uint32;
          config false;
          description
            "An integer reporting the number of records in the applicant
            database.";
          reference
            "11.5.1 of IEEE Std 802.1CS";
        }
        leaf registrar-active-records {
          type uint32;
          config false;
          description
            "An integer reporting the number of records in the registrar
            database.";
          reference
            "11.5.2 of IEEE Std 802.1CS";
        }
        leaf sent-hellos {
          type yang:counter64;
          config false;
          description
            "The number of Hello LRPDUs transmitted by the Send Hello state machines.";
          reference
            "11.5.3 of IEEE Std 802.1CS";
        }
        leaf accepted-hellos {
          type yang:counter64;
          config false;
          description
            "The number of valid Hello LRPDUsreceived by the Receive Hello
            state machine.";
          reference
            "11.5.4 of IEEE Std 802.1CS";
        }
        leaf discarded-hellos {
          type yang:counter64;
          config false;
          description
            "The number of invalid Hello LRPDUs discarded by the Receive Hello
            state machine";
          reference
            "11.5.5 of IEEE Std 802.1CS";
        }
        leaf sent-records  {
          type yang:counter64;
          config false;
          description
            "The number of Record LRPDUs transmitted by the Applicant state
            machine.";
          reference
            "11.5.6 of IEEE Std 802.1CS";
        }
        leaf accepted-records {
          type yang:counter64;
          config false;
          description
            "The number of valid Record LRPDUs received by the Partial list
            state machine.";
          reference
            "11.5.7 of IEEE Std 802.1CS";
        }
        leaf discarded-records {
          type yang:counter64;
          config false;
          description
            "The number of invalid Record LRPDUs discarded by the Partial list
            state machine.";
          reference
            "11.5.8 of IEEE Std 802.1CS";
        }
        leaf record-errors  {
```

```
        type yang:counter64;
        config false;
        description
          "The number of records discarded from otherwise-valid Record LRPDUs
          by regReceiveWriteRecord due to inconsistencies between the
          Checksum, Application data, and Data length fields.";
        reference
          "11.5.9 of IEEE Std 802.1CS";
      }
      leaf sent-partials  {
        type yang:counter64;
        config false;
        description
          "The number of Partial List LRPDUs transmitted by the Applicant
          state machine.";
        reference
          "11.5.10 of IEEE Std 802.1CS";
      }
      leaf accepted-partials {
        type yang:counter64;
        config false;
        description
          "The number of valid Partial List LRPDUs received by the Applicant
          state machine.";
        reference
          "11.5.11 of IEEE Std 802.1CS";
      }
      leaf discarded-partials {
        type yang:counter64;
        config false;
        description
          "The number of invalid Partial List LRPDUs discarded by the
          Applicant state machine.";
        reference
          "11.5.12 of IEEE Std 802.1CS";
      }
      leaf sent-complete  {
        type yang:counter64;
        config false;
        description
          "The number of Complete List LRPDUs transmitted by the Applicant
          state machine.";
        reference
          "11.5.13 of IEEE Std 802.1CS";
      }
      leaf accepted-completes {
        type yang:counter64;
        config false;
        description
          "The number of valid Complete List LRPDUs received by the Applicant
          state machine.";
        reference
          "11.5.14 of IEEE Std 802.1CS";
      }
      leaf discarded-completes{
        type yang:counter64;
        config false;
        description
          "The number of invalid Complete List LRPDUs discarded by the
          Applicant state machine.";
        reference
          "11.5.15 of IEEE Std 802.1CS";
      }
      leaf discarded-unknowns {
        type yang:counter64;
        config false;
        description
          "The number of LRPDUs of unknown type discarded by the Applicant
          state machine or Partial list state machine.";
        reference
          "11.5.16 of IEEE Std 802.1CS";
      }
```

```
    } // end portal
    list lrp-dt-instance {
    key "instance-id";
    config false;
    leaf instance-id {
      type uint32;
      config false;
      description
        "Local data transport instance";
      reference
        "Clause 7 of IEEE Std 802.1CS";
      }
      leaf active-tcp-open {
        type boolean;
        config false;
        description
          "A Boolean value that is TRUE if and only if instMyAddress and
          instNeighborAddress are TCP addresses and this LRP-DT instance is
          using the active , not the passive , form of TCP OPEN";
        reference
          "7.3.2.1 of IEEE Std 802.1CS";
      }
      leaf my-dt-address {
        type lrp-dt-address-union;
        config false;
        description
          "The address of the local system for this LRP-DT instance; the
          address used as a destination address by the neighbor LRP-DT
          instance. The address includes a type (MAC, IPv4, or IPv6) and an
          address of that type.";
        reference
          "7.3.2.2 of IEEE Std 802.1CS";
      }
      leaf my-tcp-port{
        type inet:port-number;
        config false;
        description
          "The local port number for this TCP connection, or 0, if this
          connection uses ECP, instead of TCP.";
        reference
          "7.3.2.3 of IEEE Std 802.1CS";
      }
      leaf neighbor-dt-address {
        type lrp-dt-address-union;
        config false;
        description
          "The address of the neighbor LRP-DT instance; the address used as a
          destination address by this LRP-DT instance. The address includes a
          type (MAC, IPv4, or IPv6) and an address of that type.";
        reference
          "7.3.2.4 of IEEE Std 802.1CS";
      }
      leaf neighbor-tcp-port {
        type inet:port-number;
        config false;
        description
          "The remote port number for this TCP connection, or 0, if this
          connection uses ECP, instead of TCP.";
        reference
          "7.3.2.5 of IEEE Std 802.1CS";
      }
      leaf discarded-lrpdus {
        type yang:counter64;
        config false;
        description
          "A counter indicating the number of LRPDUs discarded by the LRP-DT
          instance that cannot be assigned to a Portal for processing.";
        reference
          "11.4.1 of IEEE Std 802.1CS";
      }
    } // end lrp-dt-instance
  } // end lrp
```

```
  } // end augment system
} // end ieee802-dot1cs-lrp

```

13. MIB modules for LRP[1]

13.5 MIB modules[2]

13.5.1 LRP Textual conventions MIB

Change the MIB module in 13.5.1 with as follows:

```
LRP-TC-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    Unsigned32
        FROM SNMPv2-SMI
    ieee802dot1mibs
        FROM IEEE8021-TC-MIB
    TEXTUAL-CONVENTION
        FROM SNMPv2-TC;

ieee8021LrpTcMIB MODULE-IDENTITY
    ~~LAST-UPDATED "202012030000Z" -- December 3, 2020~~
    LAST-UPDATED "202303140000Z" -- March 14, 2023
    ORGANIZATION "IEEE 802.1 Working Group"
    CONTACT-INFO
            "WG-URL:   http://1.ieee802.org
             WG-EMail: stds-802-1-l@ieee.org

             Contact:  IEEE 802.1 Working Group Chair
             Postal:   C/O IEEE 802.1 Working Group
                       IEEE Standards Association
                       445 Hoes Lane
                       Piscataway
                       NJ 08854
                       USA
             E-mail:   stds-802-1-chairs@ieee.org"

    DESCRIPTION
            "Textual conventions used throughout IEEE Std 802.1CS.

            Unless otherwise indicated, the references in this
            MIB module are to IEEE Std 802.1CS-2020.

            ~~Copyright (C) IEEE (2021). This version of this MIB module~~
            ~~is included in clause 13 of IEEE Std 802.1CS-2020;~~
            Copyright (C) IEEE (2023). This version of this MIB module
            is included in clause 13 of IEEE Std 802.1CS-2020/Cor1-2023;
            see the standard itself for full legal notices."

    REVISION    "202303140000Z" -- March 14, 2023
    DESCRIPTION "OID changed to avoid conflict with a MIB defined in
                 IEEE Std 802.1CBcv-2021.
                "
    REVISION    "202012030000Z" -- December 3, 2020
    DESCRIPTION "This MIB module included in IEEE Std 802.1CS-2020.
                "

    ~~::= { ieee802dot1mibs 34 }~~
    ::= { ieee802dot1mibs 38 }
```

[1]An ASCII version of this MIB module can be obtained by Web browser from the IEEE 802.1 Website at http://www.ieee802.org/1/pages/MIBS.html.

[2]*Copyright release for MIBs*: Users of this standard may freely reproduce the MIBs contained in this subclause so that they can be used for their intended purpose.

```
--
-- ************************************************************
-- Textual Conventions
-- ************************************************************

LrpHelloStatus ::= TEXTUAL-CONVENTION
    STATUS  current
    DESCRIPTION
       "This specifies the current state of the Hello Receive State
        Machine.  It can take the following values:

        hsLooking(1) This Portal has not yet received a successful
                     Complete Portal create request.
        hsConnecting(2) This Portal has received a successful
                     Complete Portal create request (10.2.4), and a
                     Hello LRPDU with the hsLooking status.
                     The Portal is ready to receive all LRPDUs.
        hsConnected(3) This Portal is up and ready to transfer
                     LRP application data. The Portal is allowed to
                     transmit all LRPDUs
       "
    REFERENCE
       "8.2.2.8"
    SYNTAX  INTEGER {
        hsLooking    (1),
        hsConnecting (2),
        hsConnected  (3)
    }

LrpAppId ::= TEXTUAL-CONVENTION
    DISPLAY-HINT    "x"
    STATUS          current
    DESCRIPTION
       "Identifies an LRP application type.
        A 32 bit number. The most-significant 24 bits of the integer are
        an OUI or CID (obtainable from the IEEE Registration Authority),
        and the least-significant 8 bits are assigned by the owner of
        the OUI or CID. This creates a world-wide unique identity for
        the LRP application type.
       "
    REFERENCE   "9.2"
    SYNTAX      Unsigned32

LrpInetAddressInfo ::= TEXTUAL-CONVENTION
    STATUS  current
    DESCRIPTION
       "An LRP TCP Discovery TLV has some number of
        Application descriptors, each of which can have either one or
        two Address info fields.  The Address info field tells whether
        the following Address field is present or not, and if present,
        whether it contains an IPv4 or an IPv6 address.
        LrpInetAddressInfo can take the following values:

        noAddress(0),   Address info present, Address field not present
        addrIPv4(1),    Address info present, Address field has IPv4
        addrIPv6(2),    Address info present, Address field has IPv6
        notPresent(256) Address info not present
       "
    REFERENCE   "C.2.2.6.2"
    SYNTAX      INTEGER {
        noAddress(0),
        addrIPv4(1),
        addrIPv6(2),
        notPresent(256)
```

```
    }

END
```

13.5.2 LRP MIB

Change the MIB module in 13.5.2 with as follows:

```
LRP-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Unsigned32,
    Counter64
        FROM SNMPv2-SMI
    TruthValue
        FROM SNMPv2-TC
    MODULE-COMPLIANCE,
    OBJECT-GROUP
        FROM SNMPv2-CONF
    AddressFamilyNumbers
        FROM IANA-ADDRESS-FAMILY-NUMBERS-MIB
    InetPortNumber
        FROM INET-ADDRESS-MIB
    InterfaceIndex
        FROM IF-MIB
    LldpV2ChassisIdSubtype,
    LldpV2ChassisId,
    LldpV2PortIdSubtype,
    LldpV2PortId,
    LldpV2ManAddress
        FROM LLDP-V2-TC-MIB
    ieee802dot1mibs
        FROM IEEE8021-TC-MIB
    LrpAppId
        FROM LRP-TC-MIB;

ieee8021LrpMIB MODULE-IDENTITY
    LAST-UPDATED "202012030000Z" -- December 3, 2020
    LAST-UPDATED "202303140000Z" -- March 14, 2023
    ORGANIZATION "IEEE 802.1 Working Group"
    CONTACT-INFO
            "WG-URL:   http://1.ieee802.org
             WG-EMail: stds-802-1-l@ieee.org

             Contact:  IEEE 802.1 Working Group Chair
             Postal:   C/O IEEE 802.1 Working Group
                       IEEE Standards Association
                       445 Hoes Lane
                       Piscataway
                       NJ 08854
                       USA
             E-mail:   stds-802-1-chairs@ieee.org"
    DESCRIPTION
           "Management Information Base module for configuration of the
            Link-local Registration Protocol.

            This MIB module supports the managed objects described in
            clause 11.

            Unless otherwise indicated, the references in this
            MIB module are to IEEE Std 802.1CS-2020.

            Copyright (C) IEEE (2021). This version of this MIB module
            is included in clause 13 of IEEE Std 802.1CS-2020;
```

```
            Copyright (C) IEEE (2023). This version of this MIB module
            is included in clause 13 of IEEE Std 802.1CS-2020/Cor1-2023;
            see the standard itself for full legal notices."

    REVISION     "202303140000Z" -- March 14, 2023
    DESCRIPTION "OID changed to avoid conflict with a MIB defined in
                 IEEE Std 802.1CBcv-2021.
                "
    REVISION     "202012030000Z" -- December 3, 2020
    DESCRIPTION "This MIB module included in IEEE Std 802.1CS-2020.
                "

   ::= { ieee802dot1mibs 35 }
   ::= { ieee802dot1mibs 39 }

lrpObjects              OBJECT IDENTIFIER ::= { ieee8021LrpMIB 1 }
lrpConformance          OBJECT IDENTIFIER ::= { ieee8021LrpMIB 2 }

--
-- LRP MIB Objects
--

lrpConfiguration        OBJECT IDENTIFIER ::= { lrpObjects 1 }
lrpStatistics           OBJECT IDENTIFIER ::= { lrpObjects 2 }


-- ************************************************************
--                  L R P    C O N F I G
-- ************************************************************


-- ********************************************************************
-- The table containing information about each LRP-DT instance.
-- ********************************************************************

lrpDtInstanceTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LrpDtInstanceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A table presenting basic information about each LRP-DT instance
        in the system.
       "
    REFERENCE
       "11.4"
    ::= { lrpConfiguration 1 }

lrpDtInstanceEntry OBJECT-TYPE
    SYNTAX      LrpDtInstanceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A list of basic information about one LRP-DT instance.
       "
    INDEX { lrpDtInstNumber }
    ::= { lrpDtInstanceTable 1 }

LrpDtInstanceEntry ::= SEQUENCE {
        lrpDtInstNumber             Unsigned32,
        lrpDtInstActiveTcp          TruthValue,
        lrpDtInstAddressTypes       AddressFamilyNumbers,
        lrpDtInstMyAddress          LldpV2ManAddress,
        lrpDtInstMyTcpPort          InetPortNumber,
        lrpDtInstNeighborAddress    LldpV2ManAddress,
        lrpDtInstNeighborTcpPort    InetPortNumber
```

```
    }

lrpDtInstNumber OBJECT-TYPE
    SYNTAX      Unsigned32(1..4294967295)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A small integer identifying an LRP-DT instance.  Each
        LRP-DT instance in a system has a unique lrpDtInstNumber.

        This object SHALL NOT contain the value 0.
       "
    REFERENCE
       "11.2"
    ::= { lrpDtInstanceEntry 1 }

lrpDtInstActiveTcp OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "TRUE if and only if lrpDtInstAddressTypes indicates an IPv4
        or IPv6 address, and this LRP-DT instance uses an active TCP
        OPEN, as opposed to a passive TCP OPEN, to initiate the TCP
        connection.
       "
    REFERENCE
       "11.4, 7.3.2.1"
    ::= { lrpDtInstanceEntry 2 }

lrpDtInstAddressTypes OBJECT-TYPE
    SYNTAX      AddressFamilyNumbers
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "An enumerated value specifying the format of the addresses
        in lrpDtInstMyAddress and lrpDtInstNeighborAddress.

        If lrpDtInstAddressTypes has a value indicating a type of
        Inetnet Protocol address, then this LRP-DT instance uses TCP.
        Otherwise, it uses ECP.
       "
    REFERENCE
       "11.4, 7.3.2.2, 7.3.2.4"
    ::= { lrpDtInstanceEntry 3 }

lrpDtInstMyAddress OBJECT-TYPE
    SYNTAX      LldpV2ManAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The address used by the local end of the LRP-DT instance.  The
        format of the address is determined by lrpDtInstAddressTypes.
       "
    REFERENCE
       "11.4, 7.3.2.2"
    ::= { lrpDtInstanceEntry 4 }

lrpDtInstMyTcpPort OBJECT-TYPE
    SYNTAX      InetPortNumber
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The  local  TCP  port  number  used  for  the  TCP  connection,  or  0,
       if this connection uses ECP, instead of TCP.
```

```
        "
    REFERENCE
        "11.4, 7.3.2.3"
    ::= { lrpDtInstanceEntry 5 }

lrpDtInstNeighborAddress OBJECT-TYPE
    SYNTAX      LldpV2ManAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The address used by the partner end of the LRP-DT instance, if
         lrpDtInstActiveTcp is TRUE or lrpDtInstAddressTypes indicates
         a MAC address.  Otherwise (this LRP-DT instance was created
         from a passive TCP OPEN), lrpDtInstNeighborAddress contains a
         zero-length string.  The format of the address is determined by
         lrpDtInstAddressTypes.
        "
    REFERENCE
        "11.4, 7.3.2.4"
    ::= { lrpDtInstanceEntry 6 }

lrpDtInstNeighborTcpPort OBJECT-TYPE
    SYNTAX      InetPortNumber
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The  remote  TCP  port  number  used  for  the  TCP  connection,  or  0,
         if this connection uses ECP, instead of TCP.
        "
    REFERENCE
        "11.4, 7.3.2.5"
    ::= { lrpDtInstanceEntry 7 }


-- ****************************************************************
-- The table containing information about each LRP-DS Portal.
-- ****************************************************************

lrpPortalTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LrpPortalEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing the per-portal set of counters that record
         LRP events.  There is an entry in the table for every portal in
         a system.
        "
    REFERENCE
        "8.2.2"
    ::= { lrpConfiguration 2 }

lrpPortalEntry OBJECT-TYPE
    SYNTAX      LrpPortalEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of counters for events occurring on one Portal.
        "
    INDEX { lrpPortalNumber }
    ::= { lrpPortalTable 1 }

LrpPortalEntry ::= SEQUENCE {
        lrpPortalNumber             Unsigned32,
        lrpPortalIfIndex            InterfaceIndex,
        lrpPortalDtInstanceIndex    Unsigned32,
```

```
        lrpPortalAppId              LrpAppId,
        lrpPortalMyChassisIdType    LldpV2ChassisIdSubtype,
        lrpPortalMyChassisId        LldpV2ChassisId,
        lrpPortalMyPortIdType       LldpV2PortIdSubtype,
        lrpPortalMyPortId           LldpV2PortId,
        lrpPortalNbrChassisIdType   LldpV2ChassisIdSubtype,
        lrpPortalNbrChassisId       LldpV2ChassisId,
        lrpPortalNbrPortIdType      LldpV2PortIdSubtype,
        lrpPortalNbrPortId          LldpV2PortId,
        lrpPortalLocalOverflow      TruthValue
    }

lrpPortalNumber OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A small integer identifying a portal.  Each portal in a system
        has a unique lrpPortalNumber.

        This object SHALL NOT contain the value 0.
       "
    ::= { lrpPortalEntry 1 }

lrpPortalIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The interface index identifying the target port to which this
        portal is attached.  The value is 0, if there is none.
       "
    ::= { lrpPortalEntry 2 }

lrpPortalDtInstanceIndex OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The same value as the lrpDtInstNumber object of the
        lrpDtInstanceEntry describing the LRP-DT instance to which this
        Portal is attached.
       "
    REFERENCE
       "8.2.2.1"
    ::= { lrpPortalEntry 3 }

lrpPortalAppId OBJECT-TYPE
    SYNTAX      LrpAppId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The application ID used for this Portal.
       "
    REFERENCE
       "8.2.2.1"
    ::= { lrpPortalEntry 4 }

lrpPortalMyChassisIdType OBJECT-TYPE
    SYNTAX      LldpV2ChassisIdSubtype
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The My Chassis ID TLV type field used for this Portal.
       "
```

```
    REFERENCE
       "8.2.2.2"
    ::= { lrpPortalEntry 5 }

lrpPortalMyChassisId OBJECT-TYPE
    SYNTAX      LldpV2ChassisId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The My Chassis ID TLV Chassis ID field used for this Portal.
       "
    REFERENCE
       "8.2.2.2"
    ::= { lrpPortalEntry 6 }

lrpPortalMyPortIdType OBJECT-TYPE
    SYNTAX      LldpV2PortIdSubtype
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The My Port ID TLV type field used for this Portal.
       "
    REFERENCE
       "8.2.2.3"
    ::= { lrpPortalEntry 7 }

lrpPortalMyPortId OBJECT-TYPE
    SYNTAX      LldpV2PortId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The My Port ID TLV Port ID field used for this Portal.
       "
    REFERENCE
       "8.2.2.3"
    ::= { lrpPortalEntry 8 }

lrpPortalNbrChassisIdType OBJECT-TYPE
    SYNTAX      LldpV2ChassisIdSubtype
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The Neighbor Chassis ID TLV type field used for this Portal.
       "
    REFERENCE
       "8.2.2.2"
    ::= { lrpPortalEntry 9 }

lrpPortalNbrChassisId OBJECT-TYPE
    SYNTAX      LldpV2ChassisId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The Neighbor Chassis ID TLV Chassis ID field used for this
        Portal.
       "
    REFERENCE
       "8.2.2.2"
    ::= { lrpPortalEntry 10 }

lrpPortalNbrPortIdType OBJECT-TYPE
    SYNTAX      LldpV2PortIdSubtype
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```

```
        "The Neighbor Port ID TLV type field used for this Portal.
        "
    REFERENCE
        "8.2.2.3"
    ::= { lrpPortalEntry 11 }

lrpPortalNbrPortId OBJECT-TYPE
    SYNTAX      LldpV2PortId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Neighbor Port ID TLV Port ID field used for this Portal.
        "
    REFERENCE
        "8.2.2.3"
    ::= { lrpPortalEntry 12 }

lrpPortalLocalOverflow OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "A Boolean indicating whether or not the local registrar
         database has overflowed its alloted memory.
        "
    REFERENCE
        "8.2.2.10"
    ::= { lrpPortalEntry 13 }

-- ********************************************************
--           P O R T A L    S T A T I S T I C S
-- ********************************************************

lrpPortalCountersTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LrpPortalCountersEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing the per-portal set of counters that record
         LRP events.  There is an entry in the table for every portal in
         a system.
        "
    REFERENCE
        "11.5"
    ::= { lrpStatistics 1 }

lrpPortalCountersEntry OBJECT-TYPE
    SYNTAX      LrpPortalCountersEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of counters for events occurring on one Portal.
        "
    INDEX { lrpPortalNumber }
    ::= { lrpPortalCountersTable 1 }

LrpPortalCountersEntry ::= SEQUENCE {
        lrpPortalApplicantActiveRecords Unsigned32,
        lrpPptCtRegistrarActiveRecords  Unsigned32,
        lrpPptCtSentHellos              Counter64,
        lrpPptCtAcceptedHellos          Counter64,
        lrpPptCtDiscardedHellos         Counter64,
        lrpPptCtSentRecords             Counter64,
        lrpPptCtAcceptedRecords         Counter64,
        lrpPptCtDiscardedRecords        Counter64,
```

```
        lrpPptCtRecordErrors            Counter64,
        lrpPptCtSentPartials            Counter64,
        lrpPptCtAcceptedPartials        Counter64,
        lrpPptCtDiscardedPartials       Counter64,
        lrpPptCtSentCompletes           Counter64,
        lrpPptCtAcceptedCompletes       Counter64,
        lrpPptCtDiscardedCompletes      Counter64,
        lrpPptCtDiscardedUnknowns       Counter64
    }

lrpPortalApplicantActiveRecords OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The number of records in the applicant database.
       "
    REFERENCE
       "11.5.1"
    ::= { lrpPortalCountersEntry 1 }

lrpPptCtRegistrarActiveRecords OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The number of records in the registrar database.
       "
    REFERENCE
       "11.5.2"
    ::= { lrpPortalCountersEntry 2 }

lrpPptCtSentHellos OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "Incremented once for each Hello LRPDU transmitted by the
        Send Hello state machines.
       "
    REFERENCE
       "11.5.3"
    ::= { lrpPortalCountersEntry 3 }

lrpPptCtAcceptedHellos OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "Incremented once for each Hello LRPDU received by the
        Receive Hello state machine.
       "
    REFERENCE
       "11.5.4"
    ::= { lrpPortalCountersEntry 4 }

lrpPptCtDiscardedHellos OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "Incremented once for each invalid Hello LRPDU discarded by the
        Receive Hello state machine
       "
    REFERENCE
```

```
        "11.5.5"
    ::= { lrpPortalCountersEntry 5 }

lrpPptCtSentRecords OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Incremented once for each Record LRPDU transmitted by the
         Applicant state machine.
        "
    REFERENCE
        "11.5.6"
    ::= { lrpPortalCountersEntry 6 }

lrpPptCtAcceptedRecords OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Incremented once for each valid Record LRPDU received by the
         Partial list state machine.
        "
    REFERENCE
        "11.5.7"
    ::= { lrpPortalCountersEntry 7 }

lrpPptCtDiscardedRecords OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Incremented once for each invalid Record LRPDU discarded by the
         Partial list state machine.
        "
    REFERENCE
        "11.5.8"
    ::= { lrpPortalCountersEntry 8 }

lrpPptCtRecordErrors OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Incremented once for each record discarded from a Record LRPDU
         because of inconsistencies among the Checksum, Application data,
         and Data length fields.
        "
    REFERENCE
        "11.5.9"
    ::= { lrpPortalCountersEntry 9 }

lrpPptCtSentPartials OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Incremented once for each Partial List LRPDU transmitted by the
         Applicant state machine.
        "
    REFERENCE
        "11.5.10"
    ::= { lrpPortalCountersEntry 10 }

lrpPptCtAcceptedPartials OBJECT-TYPE
```

```
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "Incremented once for each valid Partial List LRPDU received by
        the Applicant state machine.
       "
    REFERENCE
       "11.5.11"
    ::= { lrpPortalCountersEntry 11 }

lrpPptCtDiscardedPartials OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "Incremented once for each invalid Partial List LRPDU discarded
        by the Applicant state machine.
       "
    REFERENCE
       "11.5.12"
    ::= { lrpPortalCountersEntry 12 }

lrpPptCtSentCompletes OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "Incremented once for each Complete List LRPDU transmitted by
        the Applicant state machine.
       "
    REFERENCE
       "11.5.13"
    ::= { lrpPortalCountersEntry 13 }

lrpPptCtAcceptedCompletes OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "Incremented once for each valid Complete List LRPDU received
        by the Applicant state machine.
       "
    REFERENCE
       "11.5.14"
    ::= { lrpPortalCountersEntry 14 }

lrpPptCtDiscardedCompletes OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "Incremented once for each invalid Complete List LRPDU discarded
        by the Applicant state machine.
       "
    REFERENCE
       "11.5.15"
    ::= { lrpPortalCountersEntry 15 }

lrpPptCtDiscardedUnknowns OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "Incremented once for each LRPDU of unknown type discarded by
```

```
        the Applicant state machine or Partial list state machine.
       "
    REFERENCE
       "11.5.16"
    ::= { lrpPortalCountersEntry 16 }


-- ****************************************************************
--        L R P - D T   I N S T A N C E   S T A T I S T I C S
-- ****************************************************************

lrpDtInstanceCountersTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LrpDtInstanceCountersEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A table containing the per-LRP-DT instance set of counters that
        record LRP events.  There is an entry in the table for every
        LRP-DT instance in a system.
       "
    REFERENCE
       "11.4"
    ::= { lrpStatistics 2 }

lrpDtInstanceCountersEntry OBJECT-TYPE
    SYNTAX      LrpDtInstanceCountersEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A list of statistics about one LRP-DT instance.
       "
    INDEX { lrpDtInstNumber }
    ::= { lrpDtInstanceCountersTable 1 }

LrpDtInstanceCountersEntry ::= SEQUENCE {
        lrpDtInstDiscardedLrpdus             Counter64
    }

lrpDtInstDiscardedLrpdus OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The number of received Link-local Registration Protocol Data
        Units discarded by the LRP-DT instance because it could not
        determine to which Portal it should be given.
       "
    REFERENCE
       "11.4, 11.4.1"
    ::= { lrpDtInstanceCountersEntry 1 }


--
-- ****************************************************
--           L R P   M I B   C O N F O R M A N C E
-- ****************************************************
--

lrpCompliances OBJECT IDENTIFIER ::= { lrpConformance 1 }
lrpGroups      OBJECT IDENTIFIER ::= { lrpConformance 2 }

-- compliance statements

lrpCompliance MODULE-COMPLIANCE
    STATUS  current
```

```
    DESCRIPTION
            "A compliance statement for all SNMP entities that
             implement the LRP MIB.

             This version defines compliance requirements for
             LRP MIB module.
            "
    MODULE  -- this module
        MANDATORY-GROUPS {
            lrpDsDtGroup
        }

    ::= { lrpCompliances 1 }

-- MIB groupings

lrpDsDtGroup    OBJECT-GROUP
    OBJECTS {
        lrpDtInstActiveTcp,
        lrpDtInstAddressTypes,
        lrpDtInstMyAddress,
        lrpDtInstMyTcpPort,
        lrpDtInstNeighborAddress,
        lrpDtInstNeighborTcpPort,
        lrpPortalIfIndex,
        lrpPortalDtInstanceIndex,
        lrpPortalAppId,
        lrpPortalMyChassisIdType,
        lrpPortalMyChassisId,
        lrpPortalMyPortIdType,
        lrpPortalMyPortId,
        lrpPortalNbrChassisIdType,
        lrpPortalNbrChassisId,
        lrpPortalNbrPortIdType,
        lrpPortalNbrPortId,
        lrpPortalLocalOverflow,
        lrpPortalApplicantActiveRecords,
        lrpPptCtRegistrarActiveRecords,
        lrpPptCtSentHellos,
        lrpPptCtAcceptedHellos,
        lrpPptCtDiscardedHellos,
        lrpPptCtSentRecords,
        lrpPptCtAcceptedRecords,
        lrpPptCtDiscardedRecords,
        lrpPptCtRecordErrors,
        lrpPptCtSentPartials,
        lrpPptCtAcceptedPartials,
        lrpPptCtDiscardedPartials,
        lrpPptCtSentCompletes,
        lrpPptCtAcceptedCompletes,
        lrpPptCtDiscardedCompletes,
        lrpPptCtDiscardedUnknowns,
        lrpDtInstDiscardedLrpdus
    }
    STATUS  current
    DESCRIPTION
            "The collection of objects which are used to monitor the
             status of LRP-DS and LRP-DT.
            "
    ::= { lrpGroups 1 }

END
```

13.5.3 LLDPv2 LRP extension MIB

Change the MIB module in 13.5.3 with as follows:

```
LLDP-V2-LRP-EXT-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Unsigned32
        FROM SNMPv2-SMI
    TruthValue
        FROM SNMPv2-TC
    MODULE-COMPLIANCE,
    OBJECT-GROUP
        FROM SNMPv2-CONF
    TimeFilter
        FROM RMON2-MIB
    InterfaceIndex
        FROM IF-MIB
    InetAddress,
    InetAddressIPv4,
    InetAddressIPv6,
    InetPortNumber
        FROM INET-ADDRESS-MIB
    LldpV2DestAddressTableIndex
        FROM LLDP-V2-TC-MIB
    lldpXdot1StandAloneExtensions
        FROM LLDP-EXT-DOT1-EVB-EXTENSIONS-MIB
    LrpAppId,
    LrpInetAddressInfo
        FROM LRP-TC-MIB;

lldpXDot1LrpExtensions MODULE-IDENTITY
    LAST-UPDATED "202012030000Z" -- December 3, 2020
    LAST-UPDATED "202303140000Z" -- March 14, 2023
    ORGANIZATION "IEEE 802.1 Working Group"
    CONTACT-INFO
       "WG-URL: http://www.ieee802.org/1/
        WG-EMail: stds-802-1-l@ieee.org

        Contact: IEEE 802.1 Working Group Chair
        Postal:  C/O IEEE 802.1 Working Group
                 IEEE Standards Association
                 445 Hoes Lane
                 Piscataway
                 NJ 08854
                 USA
        E-mail:  stds-802-1-chairs@ieee.org"
    DESCRIPTION
       "The LLDP Management Information Base extension module for IEEE
        802.1 organizationally-defined discovery information, as
        specified in IEEE Std 802.1CS, Link-local Registration Protocol
        (LRP).

        The Link-Layer Discovery Protocol (LLDP) is defined in
        IEEE Std 802.1AB.

        lldpXdot1StandAloneExtensions is the OUI for LLDP-EXT-DOT1-EVB-EXTENSIONS-MIB.
        which defines managed objects for IEEE 802.1-defined
        organizationally-specified LLDP Type-Length Value (TLV)
        discovery information. lldpXdot1StandAloneExtensions is branched
        from lldpV2Extensions (defined in LLDP-V2-MIB) using the
        Organizationally Unique Identifier (OUI) value 00-80-C1, which
        belongs to IEEE 802.1. An OUI is a 24 bit globally-unique number
```

```
        assigned by the IEEE Registration Authority -- see:

            http://standards.ieee.org/develop/regauth/oui/index.html

        In turn, lldpXDot1LrpExtensions and lldpV2ExtLrpConformance are
        branched from lldpXdot1StandAloneExtensions, and thus are also
        extensions from the IEEE 802.1 OUI.

        Unless otherwise indicated, the references in this MIB module
        are to IEEE Std 802.1CS-2020.

        Copyright (C) IEEE (2020). This version of this MIB module is
        included in clause 13 of IEEE Std 802.1CS-2020; see the
        Copyright (C) IEEE (2023). This version of this MIB module is
        included in clause 13 of IEEE Std 802.1CS-2020/Cor1-2023; see the
        standard itself for full legal notices."

    REVISION "202303140000Z" -- March 14, 2023
    DESCRIPTION
       "Description of lldpV2LocLrpTcpAddress1 corrected.
       "
    REVISION "2020120300000Z" -- December 3, 2020
    DESCRIPTION
       "This MIB module included in IEEE Std 802.1CS-2020.
       "

   ::= { lldpXdot1StandAloneExtensions 3 }

-- ----------------------------------------------------------------------
-- ----------------------------------------------------------------------
--
-- Organizationally Defined Information Extension - IEEE 802.1
-- Definitions to support the IEEE Std 802.1AB LLDP TLVs defined in
-- IEEE Std 802.1CS Link-local Registration Protocol (LRP)
--
-- ----------------------------------------------------------------------
-- ----------------------------------------------------------------------

lldpV2ExtLrpObjects    OBJECT IDENTIFIER ::= { lldpXDot1LrpExtensions 1 }

-- LLDP IEEE 802.1CS extension MIB groups

lldpV2ExtConfigLrp     OBJECT IDENTIFIER ::= { lldpV2ExtLrpObjects 1 }
lldpV2ExtLrpLocalData  OBJECT IDENTIFIER ::= { lldpV2ExtLrpObjects 2 }
lldpV2ExtLrpRemoteData OBJECT IDENTIFIER ::= { lldpV2ExtLrpObjects 3 }

-- ----------------------------------------------------------------------
-- IEEE 802.1 - Configuration for the LRP TLV set
-- ----------------------------------------------------------------------

--
-- The table specifying, for each LRP application, what IP
--    addresses    to    advertise    in    LRP    TCP    Discovery    TLVs    in    a
-- Controlled system.
--

lldpV2ConfigLrpTcpControlledTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2LrpConfigTcpControlledEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A table specifying what IP addresses are to be advertised as
                   the   address   of   the   Proxy   system   controlling   this   Controlled
        system, for each particular LRP application. These IP addresses
        and application identifiers can be transmitted in
```

```
        LRP TCP Discovery TLVs.
       "
    REFERENCE
       "11.6.1.1"
    ::= { lldpV2ExtConfigLrp 1 }

lldpV2ConfigLrpTcpControlledEntry OBJECT-TYPE
    SYNTAX      LldpV2LrpConfigTcpControlledEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A table specifying what IP addresses are to be advertised as
                   the   address   of   the   Proxy   system   controlling   this   Controlled
        system, for a particular LRP application. These IP addresses
        and application identifiers can be transmitted in
        LRP TCP Discovery TLVs.
       "
    INDEX { lldpV2ConfigLrpTcpControlledApplicationId }
    ::= { lldpV2ConfigLrpTcpControlledTable 1 }

LldpV2LrpConfigTcpControlledEntry ::= SEQUENCE {
        lldpV2ConfigLrpTcpControlledApplicationId   LrpAppId,
        lldpV2ConfigLrpTcpControlledTcpPortNumber   InetPortNumber,
        lldpV2ConfigLrpTcpControlledIpV4Enable      TruthValue,
        lldpV2ConfigLrpTcpControlledIpV4Address     InetAddressIPv4,
        lldpV2ConfigLrpTcpControlledIpV6Enable      TruthValue,
        lldpV2ConfigLrpTcpControlledIpV6Address     InetAddressIPv6

    }

lldpV2ConfigLrpTcpControlledApplicationId OBJECT-TYPE
    SYNTAX      LrpAppId
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
                     "The   application   identifier   to   which   the   rest   of   the
        lldpV2ConfigLrpTcpControlledEntry applies.
       "
    REFERENCE
       "9.2"
    ::= { lldpV2ConfigLrpTcpControlledEntry 1 }

lldpV2ConfigLrpTcpControlledTcpPortNumber OBJECT-TYPE
    SYNTAX      InetPortNumber
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
       "The destination TCP Port number to which TCP connections for
        LRP to the addresses in lldpV2ConfigLrpTcpControlledIpV4Address
        or lldpV2ConfigLrpTcpControlledIpV6Address, for the
        application in lldpV2ConfigLrpTcpControlledApplicationId, are
        to be made.

        If this object contains the value 0, then no
        Application descriptor with the indexed application ID is
        transmitted.

        The value of this object is restored from non-volatile
        storage after a re-initialization of the management system.
       "
    REFERENCE
       "C.2.2.6.1"
    ::= { lldpV2ConfigLrpTcpControlledEntry 2 }

lldpV2ConfigLrpTcpControlledIpV4Enable OBJECT-TYPE
```

```
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
       "Specifies whether or not the indexed LRP application is
        available through the LRP-DT TCP mechanism using TCP over IPv4.
        It thus controls whether the LRP TCP Discovery TLVs transmitted
        from this Controlled system include the IPv4 address in
        lldpV2ConfigLrpTcpControlledIpV4Address in an
        Application descriptor containing the indexed application ID.

        If lldpV2ConfigLrpTcpControlledIpV4Enable and
        lldpV2ConfigLrpTcpControlledIpV4Enable are both false(2), then
        no Application descriptor with the indexed application ID is
        transmitted.

        The value of this object is restored from non-volatile
        storage after a re-initialization of the management system.
       "
    REFERENCE
       "C.2.2.6.2"
    ::= { lldpV2ConfigLrpTcpControlledEntry 3 }

lldpV2ConfigLrpTcpControlledIpV4Address OBJECT-TYPE
    SYNTAX      InetAddressIPv4
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
       "Specifies an IPv4 address to be advertised in all of the
        LRP TCP Discovery TLVs that carry the indexed application ID
        that are transmitted by this Controlled system.

        The value of this object is restored from non-volatile
        storage after a re-initialization of the management system.
       "
    REFERENCE
       "C.2.2.6.3"
    ::= { lldpV2ConfigLrpTcpControlledEntry 4 }

lldpV2ConfigLrpTcpControlledIpV6Enable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
       "Specifies whether or not the indexed LRP application is
        available through the LRP-DT TCP mechanism using TCP over IPv6.
        It thus controls whether the LRP TCP Discovery TLVs transmitted
        from this Controlled system include the IPv6 address in
        lldpV2ConfigLrpTcpControlledIpV6Address in an
        Application descriptor containing the indexed application ID.

        If lldpV2ConfigLrpTcpControlledIpV4Enable and
        lldpV2ConfigLrpTcpControlledIpV4Enable are both false(2), then
        no Application descriptor with the indexed application ID is
        transmitted.

        The value of this object is restored from non-volatile
        storage after a re-initialization of the management system.
       "
    REFERENCE
       "C.2.2.6.2"
    ::= { lldpV2ConfigLrpTcpControlledEntry 5 }

lldpV2ConfigLrpTcpControlledIpV6Address OBJECT-TYPE
    SYNTAX      InetAddressIPv6
```

```
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
       "Specifies an IPv6 address to be advertised in all of the
        LRP TCP Discovery TLVs that carry the indexed application ID
        that are transmitted by this Controlled system.

        The value of this object is restored from non-volatile
        storage after a re-initialization of the management system.
       "
    REFERENCE
       "C.2.2.6.3"
    ::= { lldpV2ConfigLrpTcpControlledEntry 6 }

--
-- lldpV2ConfigLrpEcpTxTable: configure the transmission of the
--                 LRP ECP Discovery TLVs on a set of ports.
--

lldpV2ConfigLrpEcpTxTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2ConfigLrpEcpTxEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "This table contains one or more rows per physical network
        connection known to this agent.  The agent may wish to
        ensure that only one lldpV2ConfigLrpEcpTxEntry is present for
        each local port, or it may choose to maintain multiple
        entries for the same local port."
    REFERENCE
       "11.6.2.1"
    ::= { lldpV2ExtLrpLocalData 1 }

lldpV2ConfigLrpEcpTxEntry OBJECT-TYPE
    SYNTAX      LldpV2ConfigLrpEcpTxEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "Information about a particular port component."
    INDEX   { lldpV2ConfigLrpEcpTxLocalIfIndex,
              lldpV2ConfigLrpEcpTxLocalDestMACAddress }
    ::= { lldpV2ConfigLrpEcpTxTable 1 }

LldpV2ConfigLrpEcpTxEntry ::= SEQUENCE {
    lldpV2ConfigLrpEcpTxLocalIfIndex        InterfaceIndex,
    lldpV2ConfigLrpEcpTxLocalDestMACAddress LldpV2DestAddressTableIndex,
    lldpV2ConfigLrpEcpTxEnable              TruthValue
    }

lldpV2ConfigLrpEcpTxLocalIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The interface index value used to identify the port
        associated with this entry. Its value is an index
        into the interfaces MIB

        The value of this object is used as an index to the
        lldpV2ConfigLrpEcpTxTable.
       "
    ::= { lldpV2ConfigLrpEcpTxEntry 1 }

lldpV2ConfigLrpEcpTxLocalDestMACAddress OBJECT-TYPE
    SYNTAX      LldpV2DestAddressTableIndex
```

```
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The index value used to identify the LLDPDU frame destination
        MAC address associated with this entry. Its value identifies
        the row in the lldpV2DestAddressTable where the MAC address
        can be found.

        The value of this object is used as an index to the
        lldpV2ConfigLrpEcpTxTable.
       "
    ::= { lldpV2ConfigLrpEcpTxEntry 2 }

lldpV2ConfigLrpEcpTxEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
       "The lldpV2ConfigLrpEcpTxEnable, which is defined
        as a truth value and configured by the network management,
        determines whether the IEEE 802.1 organizationally defined
        LRP ECP Discovery TLV transmission is allowed on a given
        LLDP transmission-capable port component.

        The value of this object is restored from non-volatile
        storage after a re-initialization of the management system."
    DEFVAL  { false }
    ::= { lldpV2ConfigLrpEcpTxEntry 3 }


--
-- lldpV2ConfigLrpTcpTxTable: configure the transmission of the
--                 LRP TCP Discovery TLVs on a set of ports.
--

lldpV2ConfigLrpTcpTxTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2ConfigLrpTcpTxEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "This table contains one or more rows per physical network
        connection known to this agent.  The agent may wish to
        ensure that only one lldpV2ConfigLrpTcpTxEntry is present for
        each local port, or it may choose to maintain multiple
        entries for the same local port."
    ::= { lldpV2ExtLrpLocalData 2 }

lldpV2ConfigLrpTcpTxEntry OBJECT-TYPE
    SYNTAX      LldpV2ConfigLrpTcpTxEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "Information about a particular port component."
    INDEX   { lldpV2ConfigLrpTcpTxLocalIfIndex,
              lldpV2ConfigLrpTcpTxLocalDestMACAddress }
    ::= { lldpV2ConfigLrpTcpTxTable 1 }

LldpV2ConfigLrpTcpTxEntry ::= SEQUENCE {
    lldpV2ConfigLrpTcpTxLocalIfIndex        InterfaceIndex,
    lldpV2ConfigLrpTcpTxLocalDestMACAddress LldpV2DestAddressTableIndex,
    lldpV2ConfigLrpTcpTxEnable              TruthValue
    }

lldpV2ConfigLrpTcpTxLocalIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
```

```
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The interface index value used to identify the port
        associated with this entry. Its value is an index
        into the interfaces MIB

        The value of this object is used as an index to the
        lldpV2ConfigLrpTcpTxTable.
       "
    ::= { lldpV2ConfigLrpTcpTxEntry 1 }

lldpV2ConfigLrpTcpTxLocalDestMACAddress OBJECT-TYPE
    SYNTAX      LldpV2DestAddressTableIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The index value used to identify the LLDPDU frame destination
        MAC address associated with this entry. Its value identifies
        the row in the lldpV2DestAddressTable where the MAC address
        can be found.

        The value of this object is used as an index to the
        lldpV2ConfigLrpTcpTxTable.
       "
    ::= { lldpV2ConfigLrpTcpTxEntry 2 }

lldpV2ConfigLrpTcpTxEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
           "The lldpV2ConfigLrpTcpTxEnable, which is defined
            as a truth value and configured by the network management,
            determines whether the IEEE 802.1 organizationally defined
            LRP TCP Discovery TLV transmission is allowed on a given
            LLDP transmission-capable port component.

            The value of this object is restored from non-volatile
            storage after a re-initialization of the management system."
    REFERENCE
            "9.1.2.1 of IEEE Std 802.1AB-2016"
    DEFVAL  { false }
    ::= { lldpV2ConfigLrpTcpTxEntry 3 }

-- ------------------------------------------------------------------------
-- IEEE 802.1CS LRP LLDP TLVs - Local System Information
-- ------------------------------------------------------------------------

--
-- lldpV2LocLrpEcpTable
--

lldpV2LocLrpEcpTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2LocLrpEcpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "This table contains one or more rows per physical network
        connection known to this agent.  The agent may wish to
        ensure that only one lldpV2ExtLrpLocEntry is present for
        each local port, or it may choose to maintain multiple
        lldpV2ExtLrpLocEntries for the same local port.
       "
    REFERENCE
```

```
       "11.6.2.2"
    ::= { lldpV2ExtLrpLocalData 3 }

lldpV2LocLrpEcpEntry OBJECT-TYPE
    SYNTAX      LldpV2LocLrpEcpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "Information about the C.2.1LRP ECP Discovery TLV that can
        be transmitted from a particular LLDP port component.
        Note that this MIB supports the transmission of only one
        LRP ECP Discovery TLV per port component.
       "
    INDEX   { lldpV2LocLrpEcpLocalIfIndex,
              lldpV2LocLrpEcpLocalDestMACAddress,
              lldpV2LocLrpEcpApplicationIndex }
    ::= { lldpV2LocLrpEcpTable 1 }

LldpV2LocLrpEcpEntry ::= SEQUENCE {
    lldpV2LocLrpEcpLocalIfIndex         InterfaceIndex,
    lldpV2LocLrpEcpLocalDestMACAddress  LldpV2DestAddressTableIndex,
    lldpV2LocLrpEcpApplicationIndex     Unsigned32,
    lldpV2LocLrpEcpApplicationId        LrpAppId
    }

lldpV2LocLrpEcpLocalIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The interface index value used to identify the port
        associated with this entry. Its value is an index
        into the interfaces MIB

        The value of this object is used as an index to the
        lldpV2LocLrpEcpTable.
       "
    ::= { lldpV2LocLrpEcpEntry 1 }

lldpV2LocLrpEcpLocalDestMACAddress OBJECT-TYPE
    SYNTAX      LldpV2DestAddressTableIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The index value used to identify the LLDPDU frame destination
        MAC address associated with this entry. Its value identifies
        the row in the lldpV2DestAddressTable where the MAC address
        can be found.

        The value of this object is used as an index to the
        lldpV2LocLrpEcpTable.
       "
    ::= { lldpV2LocLrpEcpEntry 2 }

lldpV2LocLrpEcpApplicationIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (0..255)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A small integer that selects one entry in the
        lldpV2LocLrpEcpTable.  For every entry in lldpV2LocLrpEcpEntry,
        there is one Application descriptor in the transmitted
        LRP ECP Discovery TLV.

        The value of the transmitted Application count field in the
```

```
        LRP ECP Discovery TLV is equal to the number of different values
        of lldpV2LocLrpEcpApplicationIndex for this port component.
       "
    REFERENCE "C.2.1.5, C.2.1.6"
    ::= { lldpV2LocLrpEcpEntry 3 }

lldpV2LocLrpEcpApplicationId OBJECT-TYPE
    SYNTAX      LrpAppId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The AppId in one Application descriptor in the transmitted
        LRP ECP Discovery TLV.
       "
    REFERENCE "C.2.1.5, C.2.1.6"
    ::= { lldpV2LocLrpEcpEntry 4 }

--
-- lldpV2LocLrpTcpTable - indexed by ifIndex.
--

lldpV2LocLrpTcpTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2LocLrpTcpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "This table contains one or more rows per physical network
        connection known to this agent, one for each port component.
       "
    ::= { lldpV2ExtLrpLocalData 4 }

lldpV2LocLrpTcpEntry OBJECT-TYPE
    SYNTAX      LldpV2LocLrpTcpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "Information about a particular LLDP port component's transmitted
        LRP TCP Discovery TLV.  Note that this MIB supports the
        transmission of only one LRP TCP Discovery TLV per port
        component.
       "
    INDEX   { lldpV2LocLrpTcpLocalIfIndex,
              lldpV2LocLrpTcpLocalDestMACAddress,
              lldpV2LocLrpTcpApplicationIndex }
    ::= { lldpV2LocLrpTcpTable 1 }

LldpV2LocLrpTcpEntry ::= SEQUENCE {
    lldpV2LocLrpTcpLocalIfIndex         InterfaceIndex,
    lldpV2LocLrpTcpLocalDestMACAddress  LldpV2DestAddressTableIndex,
    lldpV2LocLrpTcpApplicationIndex     Unsigned32,
    lldpV2LocLrpTcpApplicationId        LrpAppId,
    lldpV2LocLrpTcpPortNumber           InetPortNumber,
    lldpV2LocLrpTcpAddressInfo1         LrpInetAddressInfo,
    lldpV2LocLrpTcpAddress1             InetAddress,
    lldpV2LocLrpTcpAddressInfo2         LrpInetAddressInfo,
    lldpV2LocLrpTcpAddress2             InetAddress
}

lldpV2LocLrpTcpLocalIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The interface index value used to identify the port
        associated with this entry. Its value is an index
```

```
        into the interfaces MIB

        The value of this object is used as an index to the
        lldpV2LocLrpTcpTable.
       "
    ::= { lldpV2LocLrpTcpEntry 1 }

lldpV2LocLrpTcpLocalDestMACAddress OBJECT-TYPE
    SYNTAX      LldpV2DestAddressTableIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The index value used to identify the destination
        MAC address associated with this entry. Its value identifies
        the row in the lldpV2DestAddressTable where the MAC address
        can be found.

        The value of this object is used as an index to the
        lldpV2LocLrpTcpTable.
       "
    ::= { lldpV2LocLrpTcpEntry 2 }

lldpV2LocLrpTcpApplicationIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (0..255)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A small integer that selects one entry in the
        lldpV2LocLrpTcpTable.  For every entry in lldpV2LocLrpTcpEntry,
        there is one Application descriptor in the transmitted
        LRP TCP Discovery TLV.

        The value of the transmitted Application count field in the
        LRP TCP Discovery TLV is equal to the number of different values
        of lldpV2LocLrpTcpApplicationIndex for this port component.
       "
    REFERENCE "C.2.2.5, C.2.2.6"
    ::= { lldpV2LocLrpTcpEntry 3 }

lldpV2LocLrpTcpApplicationId OBJECT-TYPE
    SYNTAX      LrpAppId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The AppId in one Application descriptor in the transmitted
        LRP TCP Discovery TLV.
       "
    REFERENCE "C.2.2.6"
    ::= { lldpV2LocLrpTcpEntry 4 }

lldpV2LocLrpTcpPortNumber OBJECT-TYPE
    SYNTAX      InetPortNumber
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The contents of the TCP Port number field in the transmitted
        LRP TCP Discovery TLV.  lldpV2LocLrpTcpPortNumber SHALL NOT
        contain the value 0.
       "
    REFERENCE "C.2.2.6"
    ::= { lldpV2LocLrpTcpEntry 5 }

lldpV2LocLrpTcpAddressInfo1 OBJECT-TYPE
    SYNTAX      LrpInetAddressInfo
    MAX-ACCESS  read-only
```

```
    STATUS      current
    DESCRIPTION
       "The contents of the first Address info field in the transmitted
        Application descriptor.  lldpV2LocLrpTcpAddressInfo1 SHALL NOT
        contain the value, notPresent(256).
       "
    REFERENCE "C.2.2.6"
    ::= { lldpV2LocLrpTcpEntry 6 }

lldpV2LocLrpTcpAddress1 OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The contents of the first Address field in the transmitted
        Application descriptor.  If lldpV2LocLrpTcpAddressInfo1 has the
        value notPresent(256) or noAddress(0), lldpV2LocLrpTcpAddress1
        SHALL contain a zero-length octet string.  Otherwise,
        lldpV2LocLrpTcpAddress1 SHALL contain an IPv4 or IPv6 address,
        as specified by lldpV2LocLrpTcpAddressInfo1.
       "
    REFERENCE "C.2.2.6"
    ::= { lldpV2LocLrpTcpEntry 7 }

lldpV2LocLrpTcpAddressInfo2 OBJECT-TYPE
    SYNTAX      LrpInetAddressInfo
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The contents of the second Address info field in the transmitted
        Application descriptor.  lldpV2LocLrpTcpAddressInfo2 has the
        value notPresent(256) if there is no second Address info field
        in the Application descriptor.
       "
    REFERENCE "C.2.2.6"
    ::= { lldpV2LocLrpTcpEntry 8 }

lldpV2LocLrpTcpAddress2 OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The contents of the second Address field in the transmitted
        Application descriptor.  If lldpV2LocLrpTcpAddressInfo2 has the
        value notPresent(256) or noAddress(0), lldpV2LocLrpTcpAddress2
        SHALL contain a zero-length octet string.  Otherwise,
        lldpV2LocLrpTcpAddress2 SHALL contain an IPv4 or IPv6 address,
        as specified by lldpV2LocLrpTcpAddressInfo2.
       "
    REFERENCE "C.2.2.6"
    ::= { lldpV2LocLrpTcpEntry 9 }

-----------------------------------------------------------------------
-- IEEE 802.1CS LRP LLDP TLVs - Remote (Neighbor) System Information
-----------------------------------------------------------------------

--
-- lldpV2RemLrpEcpTable
--

lldpV2RemLrpEcpTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2RemLrpEcpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
```

```
            "This table contains one or more rows per physical network
            connection known to this agent.  The agent may wish to
            ensure that only one lldpV2ExtLrpRemEntry is present for
            each local port, or it may choose to maintain multiple
            lldpV2ExtLrpRemEntries for the same local port."
    REFERENCE
       "11.6.2.3"
    ::= { lldpV2ExtLrpRemoteData 1 }

lldpV2RemLrpEcpEntry OBJECT-TYPE
    SYNTAX      LldpV2RemLrpEcpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "Information about a particular port component."
    INDEX   { lldpV2RemLrpEcpTimeMark,
              lldpV2RemLrpEcpLocalIfIndex,
              lldpV2RemLrpEcpLocalDestMACAddress,
              lldpV2RemLrpEcpIndex,
              lldpV2RemLrpEcpApplicationIndex }
    ::= { lldpV2RemLrpEcpTable 1 }

LldpV2RemLrpEcpEntry ::= SEQUENCE {
    lldpV2RemLrpEcpTimeMark             TimeFilter,
    lldpV2RemLrpEcpLocalIfIndex         InterfaceIndex,
    lldpV2RemLrpEcpLocalDestMACAddress  LldpV2DestAddressTableIndex,
    lldpV2RemLrpEcpIndex                Unsigned32,
    lldpV2RemLrpEcpApplicationIndex     Unsigned32,
    lldpV2RemLrpEcpApplicationId        LrpAppId
    }

lldpV2RemLrpEcpTimeMark OBJECT-TYPE
    SYNTAX      TimeFilter
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A TimeFilter for this entry. See the TimeFilter textual
        convention in IETF RFC 4502 to see how TimeFilter works.
       "
    REFERENCE
       "IETF RFC 4502 section 6"
    ::= { lldpV2RemLrpEcpEntry 1 }


lldpV2RemLrpEcpLocalIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The interface index value used to identify the port
        associated with this entry. Its value is an index
        into the interfaces MIB

        The value of this object is used as an index to the
        lldpV2RemLrpEcpTable.
       "
    ::= { lldpV2RemLrpEcpEntry 2 }

lldpV2RemLrpEcpLocalDestMACAddress OBJECT-TYPE
    SYNTAX      LldpV2DestAddressTableIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The index value used to identify the LLDPDU frame destination
        MAC address associated with this entry. Its value identifies
```

```
        the row in the lldpV2DestAddressTable where the MAC address
        can be found.

        The value of this object is used as an index to the
        lldpV2RemLrpTcpTable.
       "
    ::= { lldpV2RemLrpEcpEntry 3 }


lldpV2RemLrpEcpIndex OBJECT-TYPE
    SYNTAX      Unsigned32(1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "This object represents an arbitrary local integer value used
        by this agent to identify a particular connection instance,
        unique only for the indicated remote system.

        An agent is encouraged to assign monotonically increasing
        index values to new entries, starting with one, after each
        reboot. It is considered unlikely that the
        lldpV2RemLrpEcpIndex can wrap between reboots.
       "
    ::= { lldpV2RemLrpEcpEntry 4 }

lldpV2RemLrpEcpApplicationIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (0..255)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A small integer that selects one entry in the
        lldpV2RemLrpEcpTable.  For every entry in lldpV2RemLrpEcpEntry,
        there is one Application descriptor in the received
        LRP ECP Discovery TLV.

        The value of the received Application count field in the
        LRP ECP Discovery TLV is equal to the number of different values
        of lldpV2RemLrpEcpApplicationIndex for this connection instance.
       "
    REFERENCE "C.2.1.5, C.2.1.6"
    ::= { lldpV2RemLrpEcpEntry 5 }

lldpV2RemLrpEcpApplicationId OBJECT-TYPE
    SYNTAX      LrpAppId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The AppId in one Application descriptor in the received
        LRP ECP Discovery TLV.
       "
    REFERENCE "C.2.1.5, C.2.1.6"
    ::= { lldpV2RemLrpEcpEntry 6 }

--
-- lldpV2RemLrpTcpTable
--

lldpV2RemLrpTcpTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2RemLrpTcpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains one or more rows per physical network
            connection known to this agent.  The agent may wish to
            ensure that only one lldpV2ExtLrpRemEntry is present for
```

```
            each local port, or it may choose to maintain multiple
            lldpV2ExtLrpRemEntries for the same local port."
    ::= { lldpV2ExtLrpRemoteData 2 }

lldpV2RemLrpTcpEntry OBJECT-TYPE
    SYNTAX      LldpV2RemLrpTcpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "Information about a particular port component."
    INDEX   { lldpV2RemLrpTcpTimeMark,
              lldpV2RemLrpTcpLocalIfIndex,
              lldpV2RemLrpTcpLocalDestMACAddress,
              lldpV2RemLrpTcpIndex,
              lldpV2RemLrpTcpApplicationIndex }
    ::= { lldpV2RemLrpTcpTable 1 }

LldpV2RemLrpTcpEntry ::= SEQUENCE {
    lldpV2RemLrpTcpTimeMark             TimeFilter,
    lldpV2RemLrpTcpLocalIfIndex         InterfaceIndex,
    lldpV2RemLrpTcpLocalDestMACAddress  LldpV2DestAddressTableIndex,
    lldpV2RemLrpTcpIndex                Unsigned32,
    lldpV2RemLrpTcpApplicationIndex     Unsigned32,
    lldpV2RemLrpTcpApplicationId        LrpAppId,
    lldpV2RemLrpTcpPortNumber           InetPortNumber,
    lldpV2RemLrpTcpAddressInfo1         LrpInetAddressInfo,
    lldpV2RemLrpTcpAddress1             InetAddress,
    lldpV2RemLrpTcpAddressInfo2         LrpInetAddressInfo,
    lldpV2RemLrpTcpAddress2             InetAddress
    }

lldpV2RemLrpTcpTimeMark OBJECT-TYPE
    SYNTAX      TimeFilter
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A TimeFilter for this entry. See the TimeFilter textual
        convention in IETF RFC 4502 to see how TimeFilter works.
       "
    REFERENCE
       "IETF RFC 4502 section 6"
    ::= { lldpV2RemLrpTcpEntry 1 }


lldpV2RemLrpTcpLocalIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The interface index value used to identify the port
        associated with this entry. Its value is an index
        into the interfaces MIB

        The value of this object is used as an index to the
        lldpV2RemLrpTcpTable.
       "
    ::= { lldpV2RemLrpTcpEntry 2 }

lldpV2RemLrpTcpLocalDestMACAddress OBJECT-TYPE
    SYNTAX      LldpV2DestAddressTableIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The index value used to identify the destination
        MAC address associated with this entry. Its value identifies
```

```
        the row in the lldpV2DestAddressTable where the MAC address
        can be found.

        The value of this object is used as an index to the
        lldpV2RemLrpTcpTable.
       "
    ::= { lldpV2RemLrpTcpEntry 3 }


lldpV2RemLrpTcpIndex OBJECT-TYPE
    SYNTAX      Unsigned32(1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "This object represents an arbitrary local integer value used
        by this agent to identify a particular connection instance,
        unique only for the indicated remote system.

        An agent is encouraged to assign monotonically increasing
        index values to new entries, starting with one, after each
        reboot. It is considered unlikely that the
        lldpV2RemLrpTcpIndex can wrap between reboots.
       "
    ::= { lldpV2RemLrpTcpEntry 4 }

lldpV2RemLrpTcpApplicationIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (0..255)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A small integer that selects one entry in the
        lldpV2RemLrpTcpTable.  For every entry in lldpV2RemLrpTcpEntry,
        there is one Application descriptor in the received
        LRP TCP Discovery TLV.

        The value of the received Application count field in the
        LRP TCP Discovery TLV is equal to the number of different values
        of lldpV2RemLrpTcpApplicationIndex for this connection instance.
       "
    REFERENCE "C.2.2.5, C.2.2.6"
    ::= { lldpV2RemLrpTcpEntry 5 }

lldpV2RemLrpTcpApplicationId OBJECT-TYPE
    SYNTAX      LrpAppId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The AppId in one Application descriptor in the received
        LRP TCP Discovery TLV.
       "
    REFERENCE "C.2.2.6"
    ::= { lldpV2RemLrpTcpEntry 6 }

lldpV2RemLrpTcpPortNumber OBJECT-TYPE
    SYNTAX      InetPortNumber
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The contents of the TCP Port number field in the received
        Application descriptor.  lldpV2RemLrpTcpPortNumber SHALL NOT
        contain the value 0.
       "
    REFERENCE "C.2.2.6"
    ::= { lldpV2RemLrpTcpEntry 7 }

```

```
lldpV2RemLrpTcpAddressInfo1 OBJECT-TYPE
    SYNTAX      LrpInetAddressInfo
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The contents of the first Address info field in the received
        Application descriptor. ~~lldpV2RemLrpTcpAddressInfo1 SHALL NOT
        contain the value, notPresent(256).~~
       "
    REFERENCE "C.2.2.6"
    ::= { lldpV2RemLrpTcpEntry 8 }

lldpV2RemLrpTcpAddress1 OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The contents of the first Address field in the received
        Application descriptor.  If lldpV2RemLrpTcpAddressInfo1 has the
        value notPresent(256) or noAddress(0), lldpV2RemLrpTcpAddress1
        SHALL contain a zero-length octet string.  Otherwise,
        lldpV2RemLrpTcpAddress1 SHALL contain an IPv4 or IPv6 address,
        as specified by lldpV2RemLrpTcpAddressInfo1.
       "
    REFERENCE "C.2.2.6"
    ::= { lldpV2RemLrpTcpEntry 9 }

lldpV2RemLrpTcpAddressInfo2 OBJECT-TYPE
    SYNTAX      LrpInetAddressInfo
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The contents of the second Address info field in the received
        Application descriptor.  lldpV2RemLrpTcpAddressInfo2 has the
        value notPresent(256) if there is no second Address info field
        in the Application descriptor.
       "
    REFERENCE "C.2.2.6"
    ::= { lldpV2RemLrpTcpEntry 10 }

lldpV2RemLrpTcpAddress2 OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The contents of the second Address field in the received
        Application descriptor.  If lldpV2RemLrpTcpAddressInfo2 has the
        value notPresent(256) or noAddress(0), lldpV2RemLrpTcpAddress2
        SHALL contain a zero-length octet string.  Otherwise,
        lldpV2RemLrpTcpAddress2 SHALL contain an IPv4 or IPv6 address,
        as specified by lldpV2RemLrpTcpAddressInfo2.
       "
    REFERENCE "C.2.2.6"
    ::= { lldpV2RemLrpTcpEntry 11 }

-- ----------------------------------------------------------------------
-- Conformance Information for the basicSet TLV set
-- ----------------------------------------------------------------------

lldpV2ExtLrpConformance
    OBJECT IDENTIFIER ::= { lldpXdot1StandAloneExtensions 9 }
lldpV2ExtLrpCompliances
    OBJECT IDENTIFIER ::= { lldpV2ExtLrpConformance 1 }
lldpV2ExtLrpGroups
    OBJECT IDENTIFIER ::= { lldpV2ExtLrpConformance 2 }
```

```
1
2 -- compliance statements
3
4
5 lldpV2ExtLrpTxRxCompliance MODULE-COMPLIANCE
7     STATUS  current
8     DESCRIPTION
9             "A compliance statement for SNMP entities that implement
10            the IEEE 802.1 organizationally defined LLDP extension MIB.
11
12            This group is mandatory for all agents that implement the
13            LLDP 802.1 organizational extension in TX and/or RX mode
14            for the basicSet TLV set.
15
16            This version defines compliance requirements for
17            V2 of the LLDP MIB."
18    MODULE  -- this module
19    ::= { lldpV2ExtLrpCompliances 1 }
20
21
22
23 -- MIB groupings for the basicSet TLV set
24
25 lldpV2ExtLrpControlledTcpControlGroup    OBJECT-GROUP
27    OBJECTS {
28        lldpV2ConfigLrpTcpControlledTcpPortNumber,
29        lldpV2ConfigLrpTcpControlledIpV4Enable,
30        lldpV2ConfigLrpTcpControlledIpV4Address,
31        lldpV2ConfigLrpTcpControlledIpV6Enable,
32        lldpV2ConfigLrpTcpControlledIpV6Address
33    }
34    STATUS  current
35    DESCRIPTION
36       "The optional collection of objects which are required of a
37        Controlled system so that a Proxy system SNMP client can
38        configure the LRP TCP Discovery TLVs to be transmitted by the
39        Controlled system.
40       "
41    REFERENCE  "5.10:b"
42    ::= { lldpV2ExtLrpGroups 1 }
43
44 lldpV2ExtLrpEcpTlvGroup    OBJECT-GROUP
46    OBJECTS {
47        lldpV2ConfigLrpEcpTxEnable,
48        lldpV2LocLrpEcpApplicationId,
49        lldpV2RemLrpEcpApplicationId
50    }
51    STATUS  current
52    DESCRIPTION
53       "The optional collection of objects which are required of any
54        system implementing the LRP-DT ECP mechanism so that an SNMP
55        client can observe the LRP ECP Discovery TLVs transmitted and
56        received by the system."
57    REFERENCE  "Clause 5"
58    ::= { lldpV2ExtLrpGroups 2 }
59
60 lldpV2ExtLrpTcpTlvGroup    OBJECT-GROUP
62    OBJECTS {
63        lldpV2ConfigLrpTcpTxEnable,
64        lldpV2LocLrpTcpApplicationId,
65        lldpV2LocLrpTcpPortNumber,
66        lldpV2LocLrpTcpAddressInfo1,
67        lldpV2LocLrpTcpAddress1,
68        lldpV2LocLrpTcpAddressInfo2,
69        lldpV2LocLrpTcpAddress2,
```

```
        lldpV2RemLrpTcpApplicationId,
        lldpV2RemLrpTcpPortNumber,
        lldpV2RemLrpTcpAddressInfo1,
        lldpV2RemLrpTcpAddress1,
        lldpV2RemLrpTcpAddressInfo2,
        lldpV2RemLrpTcpAddress2
    }
    STATUS  current
    DESCRIPTION
       "The optional collection of objects which are required of any
        system implementing the LRP-DT TCP mechanism so that an SNMP
        client can observe the LRP TCP Discovery TLVs transmitted and
        received by the system."
    REFERENCE  "Clause 5"
    ::= { lldpV2ExtLrpGroups 3 }

END

```

Annex C

(normative)

IEEE 802.1 Organizationally Specific TLVs for LLDP

C.2 Organizationally Specific TLV definitions

C.2.2 LRP TCP Discovery TLV

C.2.2.6 Application descriptor

Change C.2.2.6 as follows:

An Application descriptor for the LRP TCP Discovery TLV contains four octets with an AppId, followed by one or two addresses to use to make the TCP association, as shown in Figure C-1. The first three octets contain the OUI or CID of the organization assigning the AppId, and the fourth octet identifies a specific LRP application that connects using TCP. Following these four octets is a two-octet TCP port number, followed in turn by the descriptions of one or two IP addresses, as shown in Table C-1.

Offset	0	3	4	6	7	varies	varies
	OUI or CID for AppId (3 octets)	AppId subtype (1 octet)	TCP port number (2 octets)	Address info 1 (1 octet)	Address 1 (0, 4, or 16 octets)	Address info 2 (1 octet)	Address 2 (0, 4, or 16 octets)

Figure C-1—LRP TCP Discovery TLV application descriptor format

Table C-1—Allowed address information encodings and lengths

Address info 1	Address 1 length	Address info 2	Address 2 length	Total length of address information
noAddress	0	addrIPv4	4	6
		addrIPv6	16	18
addrIPv4	4	not present[a]	0	5
		noAddress	0	6
		addrIPv6	16	22
addrIPv6	16	not present[a]	0	17
		noAddress	0	18
		addrIPv4	4	22

[a] Allowed only at the end of the TLV. See C.2.2.7.

C.2.2.7 LRP TCP Discovery TLV usage rules

Change section C.2.2.7 as follows:

The LRP TCP Discovery TLV is used to establish TCP associations among the Portals on a physical link. A Native or Controlled system should transmit this TLV if and only if the information transmitted in the LLDPDU matches the information in an entry in the imTargetPortList (7.2.2.1) in the Native system or the Controlled's Proxy system.

A system shall not transmit the same AppId value in more than one Application descriptor of the same LRP TCP Discovery TLV. A system can transmit more than one LRP TCP Discovery TLV for different sets of AppId values, but the same AppId value shall not appear in more than one LRP TCP Discovery TLV in the same LLDPDU. At least one IP address shall be included in an LRP TCP Discovery TLV. If two addresses are included, they shall be of different types (addrIPv4 vs. addrIPv6). The second AddressInfo field can be omitted from an application descriptor (be not present in Table C-2) only if it would be the last octet of the LRP TCP Discovery TLV.

NOTE—The provision for sending or receiving multiple LRP TCP Discovery TLVs is made to facilitate control of these LLDP TLVs by more than one LRP application.

This standard does not specify the actions to be taken if a received LRP TCP Discovery TLV violates these usage rules.

As explained in 9.2.7.7.2 of IEEE Std 802.1AB-2016, a received LRP TCP Discovery TLV is not invalid if the value of the TLV information string length field is larger than the total amount of information indicated by the Application count and Application descriptors. A system receiving such a TLV shall interpret the TLV as described in this standard, and ignore any information following the Application descriptor.