IEEE P802.1ABdh Update to LSVR
(Note: P802.1ABdh == XLLDP == LLDPv2)

IETF-112

Online

Paul Congdon (Tallac Networks)

Steve Haddock(Stephen Haddock Consulting LLC)

November 9, 2021

Disclaimer

- This presentation should be considered as the personal view of the presenter not as a formal position, explanation, or interpretation of IEEE.

- Per IEEE-SA Standards Board Bylaws, December 2017
  - “At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.”

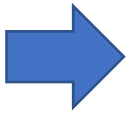
Background - References

- Motivated by the requirements of L3DL and other use cases needing to advertise more than 1 frame's worth of information
  - https://www.ieee802.org/1/files/public/docs2019/new-congdon-lsvr-disco-requirements-for-LLDPv2-0119-v01.pdf
- Last update at IETF-110 with previous background – March 2021
  - https://datatracker.ietf.org/meeting/110/materials/slides-110-lsvr-2-lldpv2-status-update-ieee-liaison-slot-00
- IEEE 802 approval to start P802.1ABdh – Standard for Local and Metropolitan Area Networks - Station and Media Access Control Connectivity Discovery Amendment: Support for Multiframe Protocol Data Units – September 2019
  - https://standards.ieee.org/project/802_1ABdh.html
- IEEE 802.1 Project Page
  - https://1.ieee802.org/tsn/802-1abdh/

IEEE 802.1ABdh timeline

- Project Authorization Request (PAR) – Sep 9, 2019
- Task Group Balloting (TG) – Oct 23, 2020; Jan 13, 2021
- Working Group Balloting (WG) – Mar 4, 2021; April 30, 2021; May 24, 2021
- Standards Association Balloting (SA) – Aug 25, 2021; Oct 4, 2021
- RevCom Approval – motion on Nov 16, 2021
- Publication – December 2021?

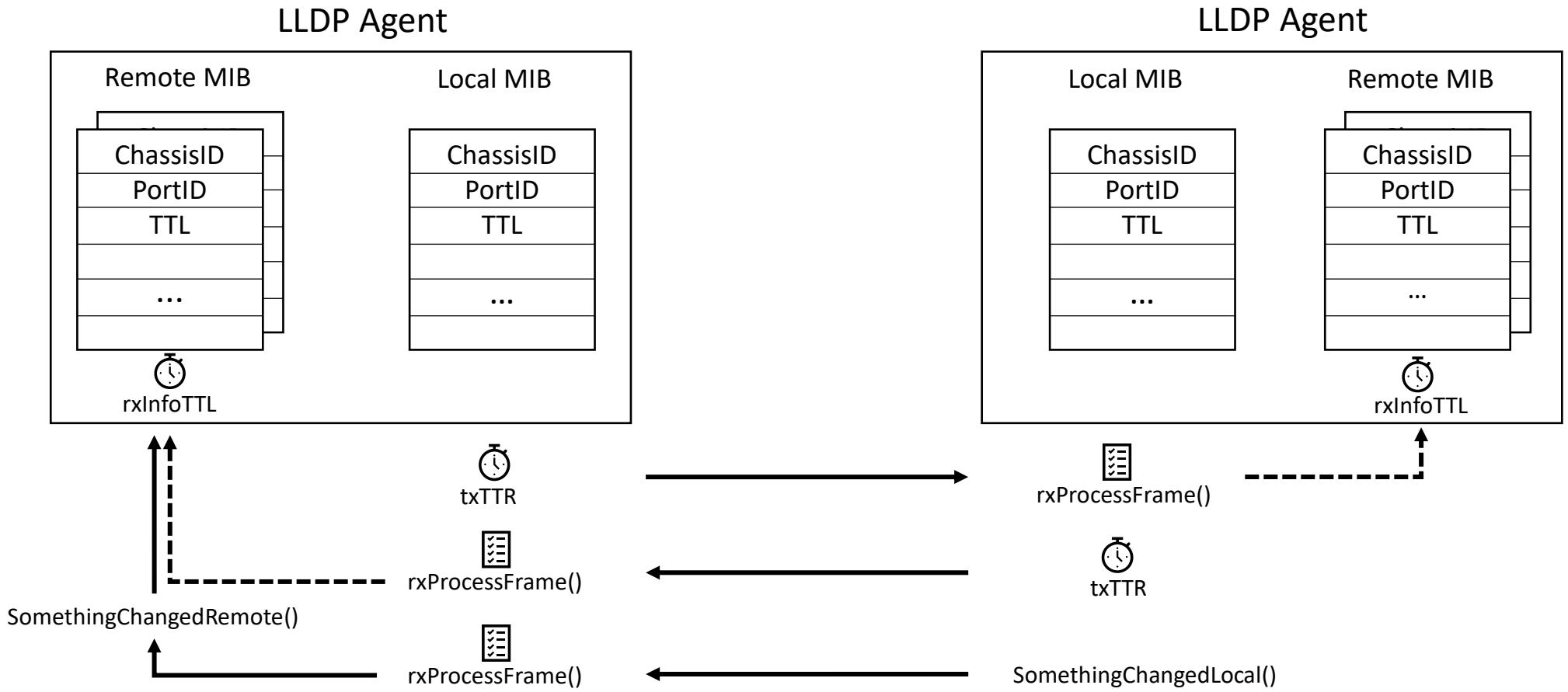
You are here!



Objectives for New LLDPv2 Method

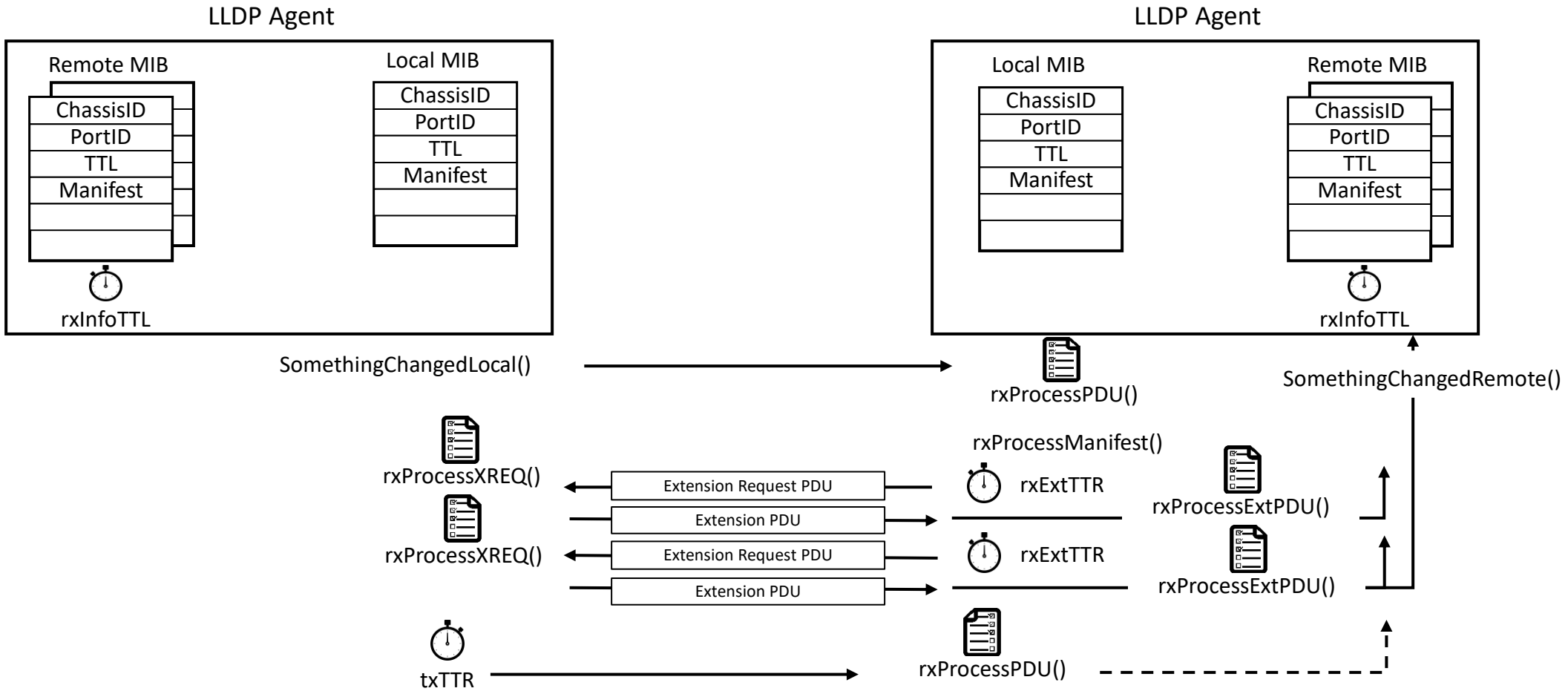
1. Support sending more information than can fit in a single frame
   - Optimize LLDPv2 for around 100K bytes
   - For reference IETF currently believes around 64K bytes are sufficient
2. Support the ability to limit the LLDP frame size to meet timing constraints imposed by some Time Sensitive Networking (TSN) applications
3. Support the ability to communicate with an LLDPv1 implementation
   - Only the single LLDPv1 frame information would be exchanged between and LLDPv1 and LLDPv2 implementation
4. Ensure the integrity of the full set of TLVs received by partners
5. Support pacing of PDUs by receivers to prevent overloading constrained implementations
6. Reduce network traffic by reducing periodic transmission to the minimum
   - Only update the foundation LLDPv1 PDU periodically
   - Extension PDUs are only transmitted/updated on demand from receivers
   - Update extension PDUs only when they have changed

Current LLDP operation reminder



NOTE: Think of the Remote and Local MIBs as a database that must fit into a single PDU
Replace all values of the Remote MIB with contents of LLDPDU when something changes

LLDP Extension Operation Proposal: Receiver Pacing



NOTE: Send LLDPDU as specified by LLDPv1 when something changes and periodically
Only send extension LLDPDU when explicitly requested by a XREQ
Only issue XREQ when manifest shows the local copy is out of date

3 new TLVs and 2 new PDUs

(see backup for details)

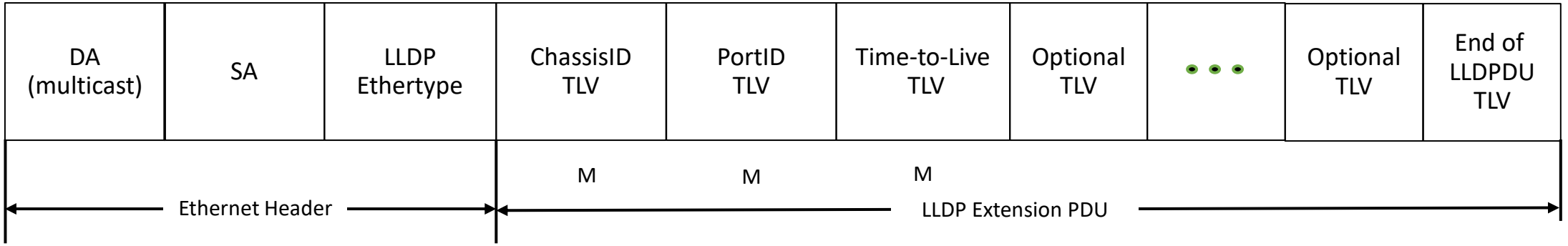
- New TLVs
  - Manifest TLV
    - Describes a list of additional Extension LLDPDUs available to be transmitted
    - Included in the original LLDPv1 Normal LLDPDU
  - Extension Request TLV
    - Requests one or more individual Extension LLDPDUs from the sender
  - Extension PDU Identifier TLV
    - Identifies an individual Extension LLDPDU from the Extension Request TLV / Manifest TLV
- PDUs
  - Normal LLDPDU
    - The existing LLDPDU, sent periodically, typically uses a scope limited Multicast address
    - May contain a Manifest TLV (which will be ignored by existing LLDPv1 implementations)
  - Extension Request LLDPDU
    - Sent to a specified MAC address (i.e. unicast)
    - Contains the Extension Request TLV
  - Extension LLDPDU
    - Sent to a specified MAC address (i.e. unicast)
    - Contains the Extension Identifier TLV and other optional standard TLVs

P802.1ABdh Next Steps

- Awaiting final publication
- Dust off https://datatracker.ietf.org/doc/draft-congdon-lsvr-lldp-tlvs/?
- Open Source implementation?

Backup - Details

Normal LLDP PDU (LLDPDU)



- LLDP Header
  - DA is multicast (defines the 'scope (i.e. reach)' of the LLDPDU)
  - SA is the MAC address of the sender
  - LLDP Ethertype
- Chassis ID + Port ID + TTL are mandatory TLVs
  - The Chassid ID and Port ID form a logical MSAP identifier. This plus the DA MAC Address identify the sending LLDP agent/port
  - The TTL defines the lifetime of the information, unless refreshed
- End of LLDPDU TLV
  - Optional

Manifest TLV: Added to the Normal LLDPDU

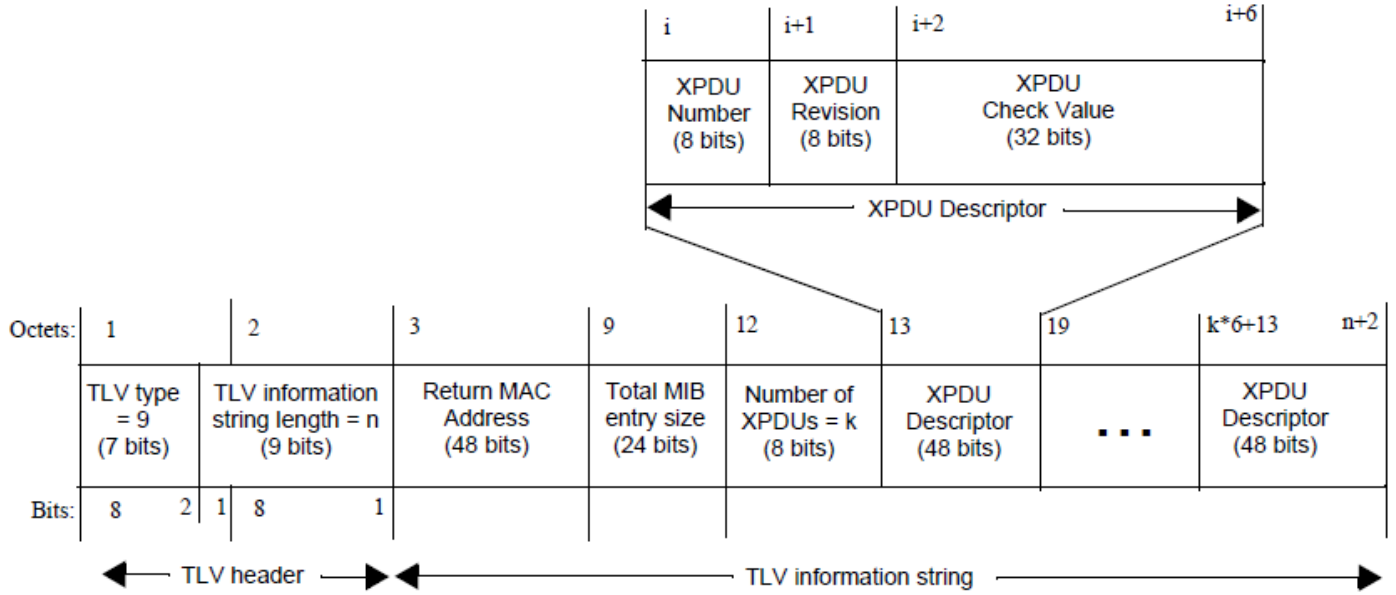
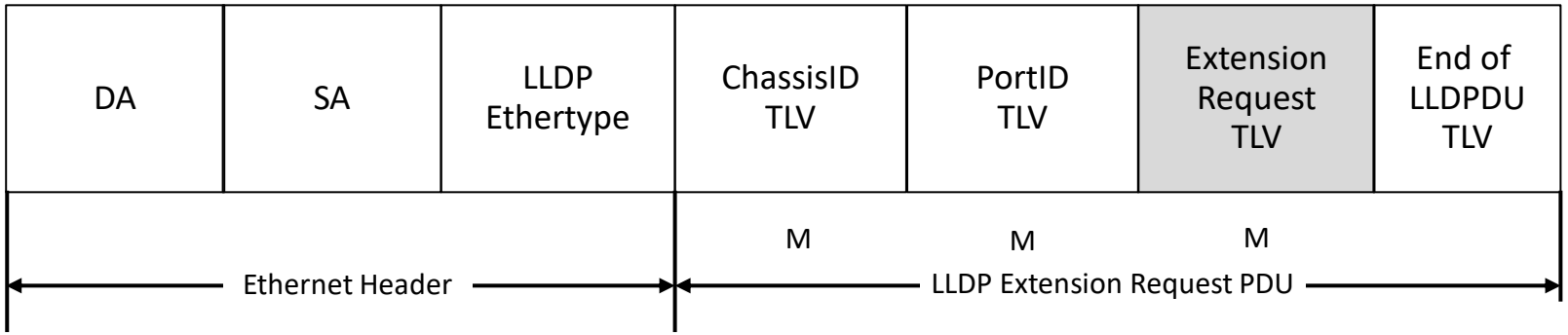


Figure 8-13—Manifest TLV format

- Return MAC Address
  - Specifies the MAC address to use in subsequent Extension Request LLDPDUs
- Number of extension PDUs descriptors in the manifest
- Each Extension PDU is identified by a:
  - Extension LLDPDU number, this number is included in the manifest to facilitate PDU deletion and insertion
  - Extension LLDPDU revision, updated modulo 256 on every change to the extension LLDPDU
  - Extension LLDPDU check: low 32 bits of MD5

Extension Request LLDPDU



- Identifying Extension Request LLDPDU
  - DA is the return MAC address specified in the Manifest TLV
  - SA is the MAC address of the sender's port
  - LLDP Ethertype
  - 3rd TLV is an Extension Request TLV
- Chassis ID + Port ID + Extension Request TLV are mandatory TLVs
- End of LLDPDU TLV
  - Optional

Extension Request TLV

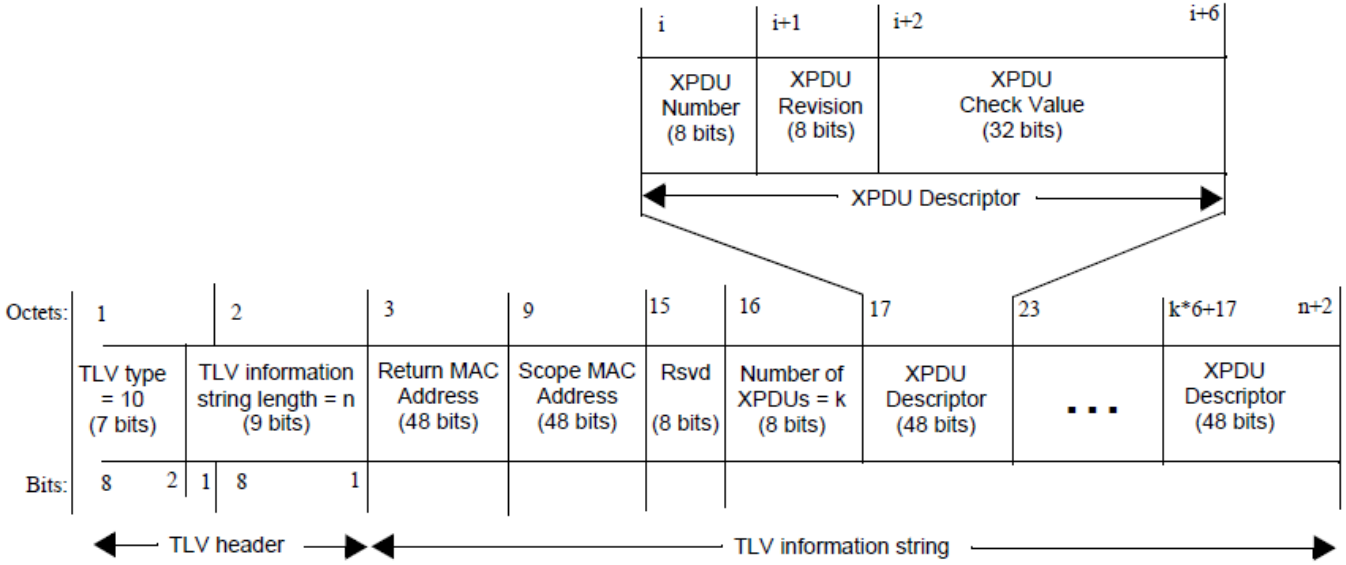
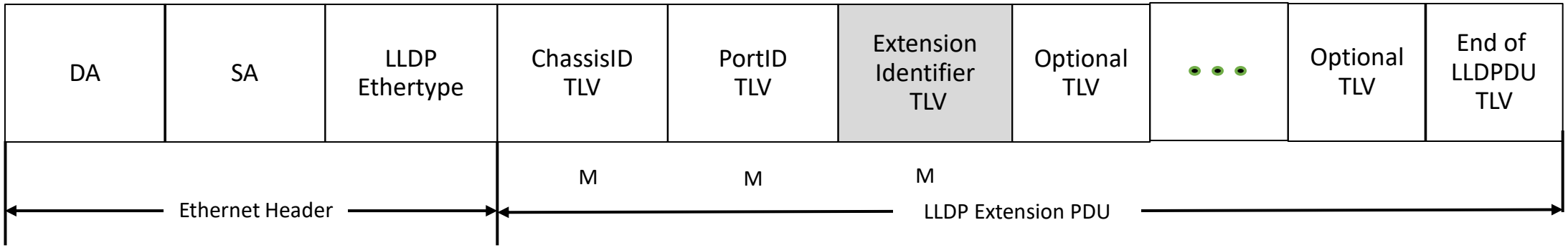


Figure 8-14—Extension Request TLV format

- Return MAC Address
  - Specifies the MAC address to use in subsequent Extension LLDPDUs
- Scope MAC Address
  - Identifies the instance of LLDP requesting these extensions
- Number of extension PDUs descriptors in the manifest
- Each Extension PDU is identified by a:
  - Extension LLDPDU number, this number is included in the manifest to facilitate PDU deletion and insertion
  - Extension LLDPDU revision, updated modulo 256 on every change to the extension LLDPDU
  - Extension LLDPDU check: low 32 bits of MD5

LLDP Extension LLDPDU



- Identifying an Extension LLDPDU
  - DA is the return MAC address specified in the Extension Request LLDPDU
  - SA is the MAC address of the sender's port
  - LLDP Ethertype
  - 3rd TLV is an Extension TLV
- Chassis ID + Port ID are mandatory
- Extension Identifier TLV is mandatory and must be the third TLV
  - Identifies this Extension PDU, the PDU revision

Extension PDU Identifier TLV (XID TLV):

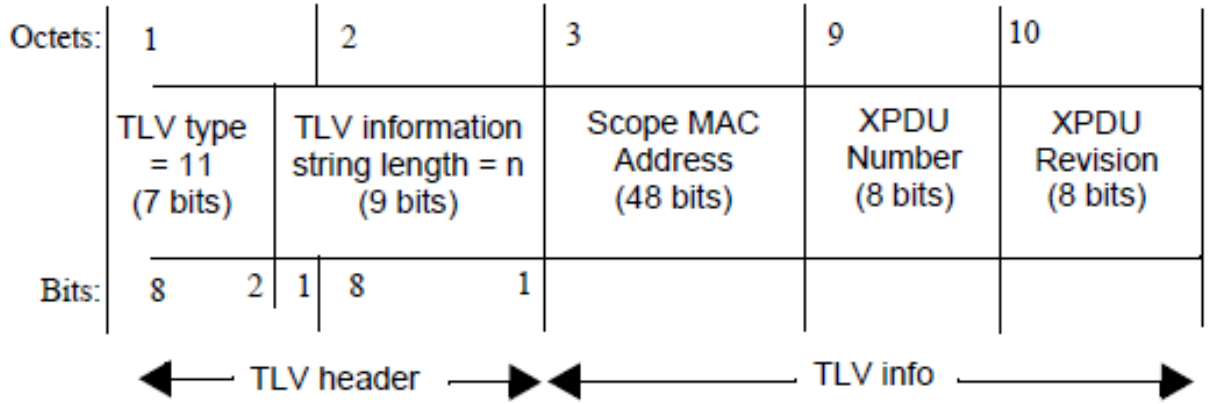


Figure 8-15—Extension Identifer TLV format

- Scope MAC Address
  - Identifies the instance of LLDP transmitting the extensions
- Extension PDU Number is the designation number for this PDU
  - The PDU number
  - Matched to the manifest extension PDU number
- Extension PDU revision number
  - Incremented modulo 256 whenever the extension LLDPDU is changed
  - Matched to the manifest to guarantee the extension LLDPDU is the one represented in the manifest
- Note the extension PDU check code is not carried in the Extension TLV and so must be calculated to match the manifest check code

IEEE 802 Patent Policy and Letters of Assurance

- Patent Policy: https://standards.ieee.org/about/sasb/patcom/patents.html
- LoA list: https://standards.ieee.org/about/sasb/patcom/patents.html