

IEEE-IETF Liaison response & IEEE 802.1 WG LLDPv2 proposal

IETF-104

Prague

Paul Congdon (Huawei/Tallac)

IETF LSVR Liaison to IEEE 802.1

- NOTE: This is an individual perspective. I'm not officially representing IEEE 802.1
- IEEE 802.1 received an inbound liaison from IETF LSVR providing background and requesting comment on the LSoE work item
 - <http://www.ieee802.org/1/files/public/docs2019/liaison-IETF-LSVR-Link-State-over-Ethernet-0119.pdf>
- IEEE 802.1 WG drafted and approved a response at the March IEEE 802 Plenary meeting
 - <http://www.ieee802.org/1/files/public/docs2019/liaison-response-IETF-LSVR-Link-State-over-Ethernet-0319-v01.pdf>

Liaison response summary

- NOTE: This is an individual perspective. I'm not officially representing IEEE 802.1
- Summary: The 802.1 WG...
 - is aware of the work in LSVR
 - has concerns about confusion with SPB (Shortest Path Bridging). Mostly a naming issue – LSoE doesn't itself maintain link state
 - has concerns about the development of multiple L2 discovery protocols
 - is entertaining proposals to amend LLDP to support LSVR requirements
 - believes the LSVR 'liveness' requirement may not be suitable for LLDP, but other L2 solutions exist to maintain and report liveness.
 - welcomes LSVR participation and collaboration in LLDPv2 amendment project.

Background

- Initial LLDPv2 proposal - presented on 1/7/2019 at TSN call
 - <http://www.ieee802.org/1/files/public/docs2019/new-congdon-lldpv2-consideration-0119-v01.pdf>
- Evaluation of LLDPv2 proposal against LSVR requirements – Presented at IEEE 802.1 Interim in Hiroshima
 - <http://www.ieee802.org/1/files/public/docs2019/new-congdon-lsvr-disco-requirements-for-LLDPv2-0119-v01.pdf>
- Next steps discussion at IEEE 802 Plenary in March 2019
 - <http://www.ieee802.org/1/files/public/docs2019/new-congdon-LLDPv2-0319-v01.pdf>

Summary of proposed LLDPv2 support for LSoE exchange

Details found in : <http://www.ieee802.org/1/files/public/docs2019/new-congdon-lsvr-disco-requirements-for-LLDPv2-0119-v01.pdf>

Information Exchanged	Supportable by an LLDP TLV and Protocol	Comments
MyMac Address	<input checked="" type="checkbox"/>	
Local ID / Remote ID	<input checked="" type="checkbox"/>	
Attribute List	<input checked="" type="checkbox"/>	
Authentication Data	<input type="checkbox"/>	May exceed current LLDP TLV length restrictions
Encapsulation & Addresses	<input checked="" type="checkbox"/>	Must be split across multiple TLVs & PDUs (as with LSoE)
Keepalives	<input checked="" type="checkbox"/>	Frequency may not be appropriate for existing LLDP
Acks	<input checked="" type="checkbox"/>	Implicit part of LLDPv2 proposal

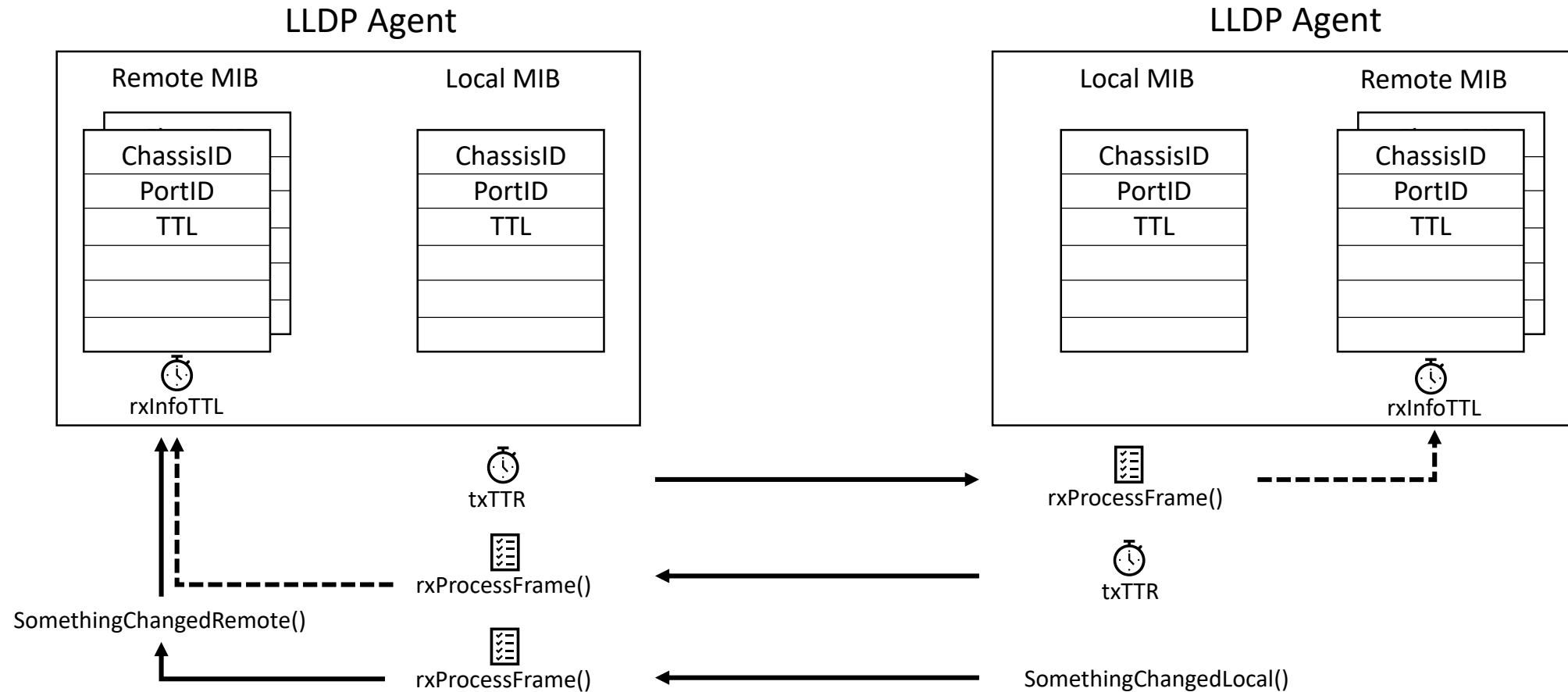
Motivation to amend LLDP

- Why do we need to update LLDP?
 - LLDP is widely deployed in many environments
 - The number of TLVs sent in LLDPDUs continues to grow
 - New standards continue to defined new objects
 - A large number of Vendor Specific TLVs
 - Alternative protocols are being proposed to get around the single PDU size limit
 - Relying on Jumbo frames to support more TLVs is problematic in many environments
 - Summary: We need to be able to exchange more TLVs. IETF Link State Vector Routing (LSVR) requirements are just one example use case where this is needed.

Objectives for a new version

- Support the ability to send more than 1 PDUs worth of TLVs
- Support the ability to limit LLDPDU size to meet timing constraints
- Support the ability to communicate with an LLDPv1 implementation (only the first PDUs worth of TLVs).
- Ensure the integrity of the full set of TLVs is received by partner
 - NOTE: This can be useful in v1 implementations as well
- Consider if there are other optimizations to address
 - E.g. Less frequent updates
 - E.g. New reachability addresses (Nearest-station or Nearest-Router)
 - E.g. allow larger TLVs and/or the ability of the contents to span multiple extension PDUs

Current LLDP operation reminder

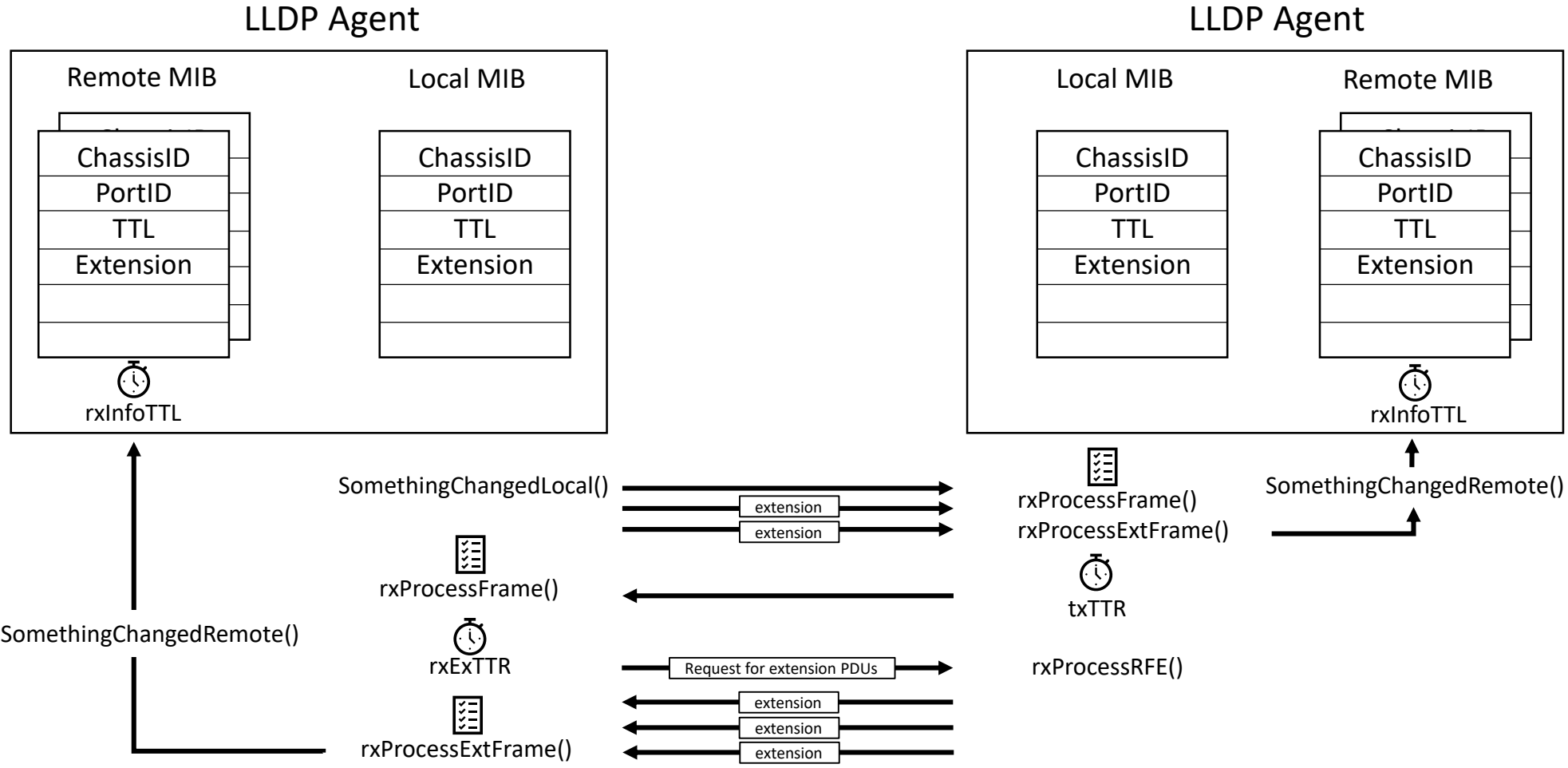


NOTE: Think of the Remote and Local MIBs as a database that must fit into a single PDU
Replace all values of the Remote MIB with contents of LLDPDU when something changes

Proposal

- Define a new mandatory (for v2 implementations) TLV that appears just after the current mandatory set of 3 TLVs.
 - ChassisID TLV + PortID TLV + TTL TLV + (new) ExtensionPDU TLV
 - Received by v1 implementations.
- In the new TLV, define a vector that specifies:
 - The number of extension PDUs to be sent
 - A way to uniquely identify each PDU (e.g. hash, checksum, version number or PDU number)
- The new extension PDUs will be ignored by v1 LLDP.
- The new PDUs need to have a mandatory format as well:
 - Includes at least the first two mandatory TLVs of a v1 PDU (ChassisID + PortID)
 - Includes new TLV that identifies the extension PDU.
- A new Request for Extension (RFE) message is sent from receiving peer when needed
 - Support multiple peers on shared media
 - When receiving station does not have all extension messages, an RFE is sent
- Optimizations:
 - There is no need to resend extension PDUs if nothing has changed.
 - Only periodically send the 1st PDU.
 - TTL in 1st PDU relates to all extension PDUs.
- NOTE: The maximum size of a TLV defines the maximum number of extension PDUs that can be included. (depends on identity field)

Proposed LLDPv2 Operation



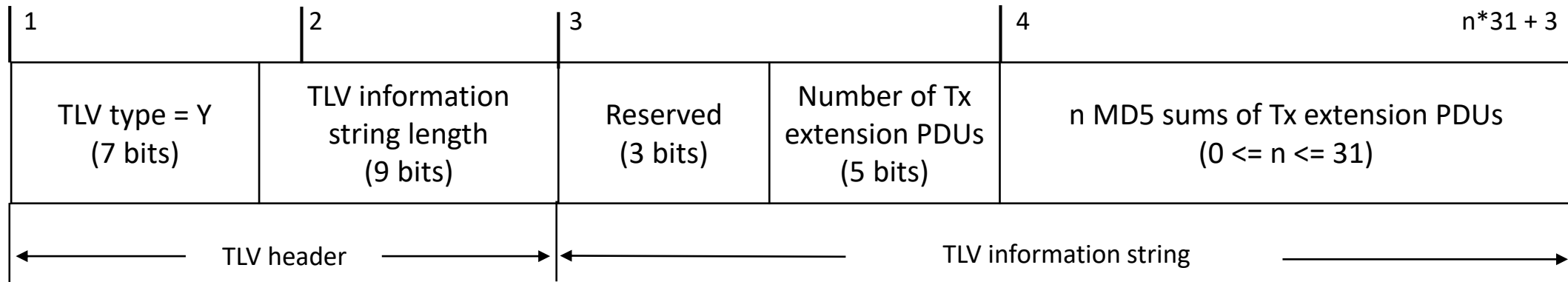
NOTE: Send primary LLDPDU and all extension PDU when something changes locally
 If extension data has NOT changed, no need to send anything other than primary LLDPDU
 If receiver does not have current ExtPDUs, send Request for Extension PDUs (RFE)

LLDPv2 Proposal Status

- 802.1 WG Approved to generate a PAR and CSD for a new project in May
- Design team continues to evolve the proposal to meet requirements and build an efficient protocol
- Details of current proposal provided in backup

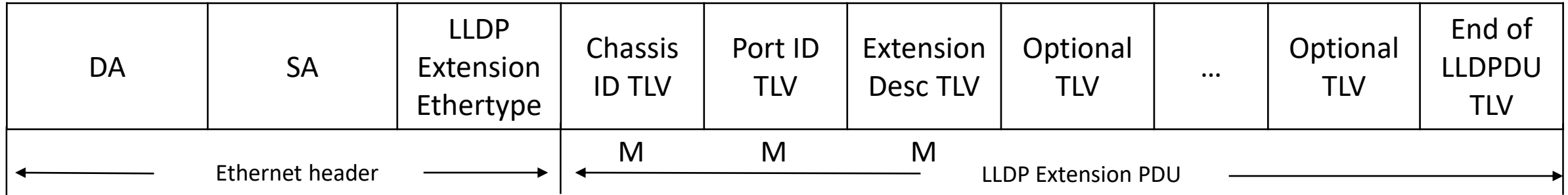
Backup - Details

Example Extension TLV



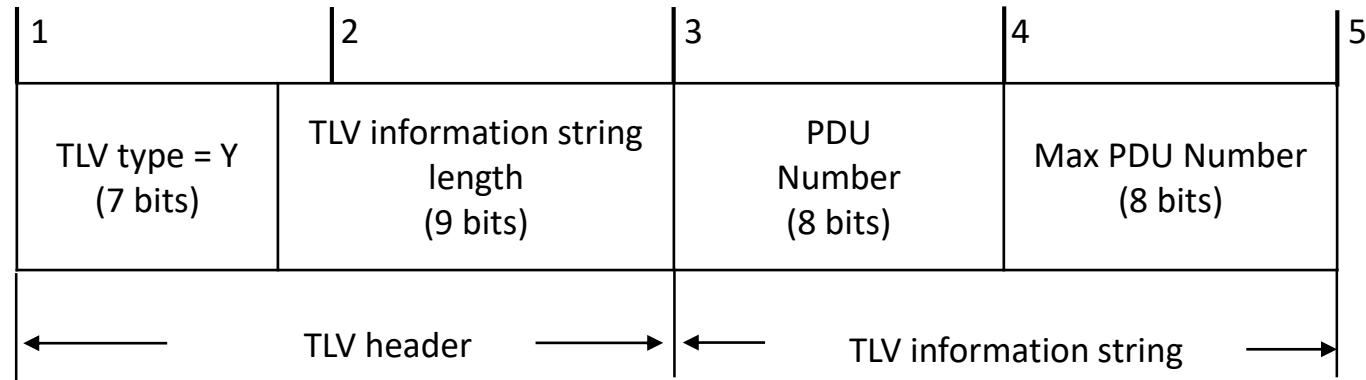
- TLV Type
 - probably use the next reserved type (i.e. 9)
- Number of Tx extension PDUs
 - If using MD5 Sum of 16 bytes, can only pack 31 sums into a TLV
- MD5 Sums
 - Should cover the entire extension LLDPDU

Example Extension PDU



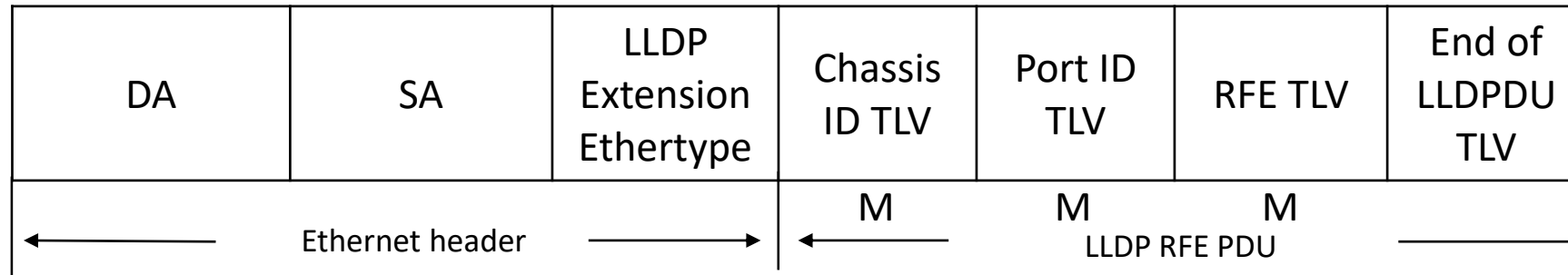
- LLDP Extension Ethertype
 - New Ethertype allows LLDPv1 implementations to ignore these frames
- Chassis ID + Port ID are mandatory
 - Note TTL from 1st PDU should apply and is not needed here
- Extension Description TLV
 - Numbers the extension PDU in the sequence of all extension PDUs

Example Extension Description TLV



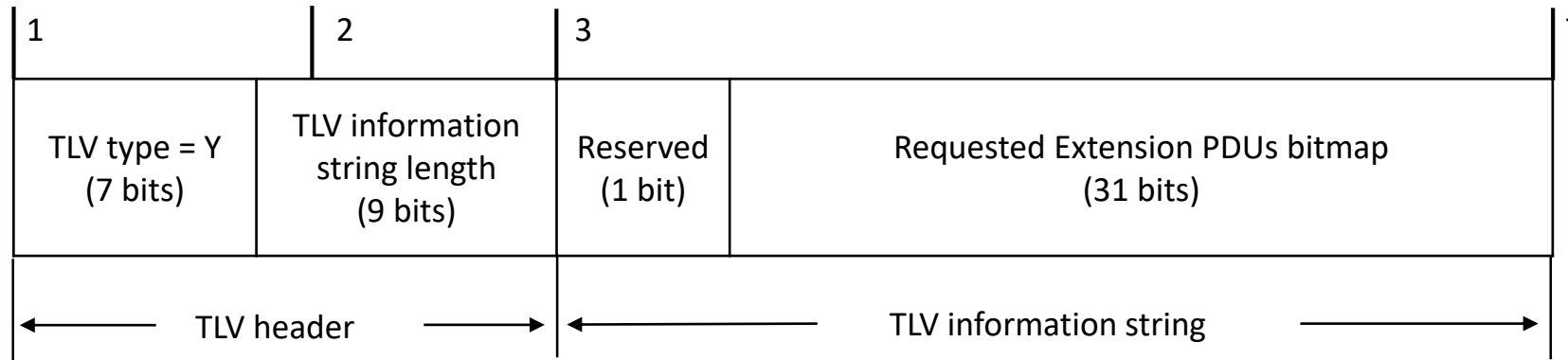
- TLV Type
 - Another new base TLV type (i.e. 10)
- PDU Number and Max PDU Number
 - For example PDU 1 of 5
 - Max ≤ 31 if using 16 byte checksums

Example Request for Extension PDUs



- LLDP Extension Ethertype
 - New Ethertype allows LLDPv1 implementations to ignore these frames
- Chassis ID + Port ID are mandatory
- Request for Extension PDUs TLV
 - Identifies extension PDUs that need to be (re)sent by peer

Example Requested Extension PDUs TLV



- TLV Type
 - Another new base TLV type (i.e. 11)
- Requested Extension PDUs bitmap
 - A bit set for each numbered extension PDU that is needed by the sender

Questions / comments / TBDs

- How to define the extension PDU TLV?
 - It needs to contain a vector of information for all extension PDUs
 - The smaller the identity field, the more extension PDUs that can be supported (e.g. CRC-16 or MD5 Hash?)
 - Should the MD5 Hash cover all PDUs or individual?
- When to send the Request for Extension PDUs (RFE)?
 - Receiver should wait some amount of time (with randomized jitter) before sending RFE.
 - RFE should be unicast and can request individual Extension PDUs.
- Retransmission strategy? SACK or just retransmit the entire sequence?