Protocol for Extending LLDP

Paul Bottorff
Paul.Bottorff@hpe.com

Extending LLDP Agenda:

–Introduce New Definitions

–Motivations and Objectives

–Link Layer Discovery Extension Protocol Principles

–Shared CSMA/CD Ethernet Worst Case Operation

–Summary and New Definition Discussion

New Definitions: Hold Discussion Until End Of Presentation

- **Link Layer Discovery Foundation PDU (LLDFPDU,F-PDU):** This is the single LLDPv1 PDU. In context this can be shortened to “foundation PDU” or F-PDU.
- **Link Layer Discovery Extension PDU (LLDXPDU,X-PDU):** This is an extension PDU for the LLDP database. In context this can be shortened to “extension PDU” or X-PDU.
- **Link Layer Discovery Extension Request PDU (LLDXREQPDU,XREQ-PDU):** This PDU is a request for transmission of one or more X-PDUs. In context this can be shortened to “request PDU” or XREQ-PDU.
- **Link Layer Discovery Extension Protocol (LLDXP):** This is the protocol used to exchange the extension PDUs of a multi-frame database.
- **Manifest TLV:** This is an LLDP TLV which describes each X-PDU of an Extended LLDP Database.
- **Extension PDU Identifier TLV (XID TLV):** This is an LLDP TLV carried in an X-PDU used to help identify the PDU.
- **Extension Request TLV (XREQ TLV):** This is an LLDP TLV carried in a RFXPDU and used to identify the requested X-PDUs.

Extending LLDP Agenda:

–Introduce New Definitions

–Motivations and Objectives

–Link Layer Discovery Extension Protocol Principles

–Shared CSMA/CD Ethernet Worst Case Operation

–Summary and New Definition Discussion

Discovery Protocols

- New IETF work on Link State Vector Routing (lsvr) has resulting in development of a discovery protocol called Layer 3 Data Link (l3dl) also IETF bgp group has a contribution for neighbor discovery protocol
  - The lsvr draft in progress is: https://tools.ietf.org/pdf/draft-ietf-lsvr-l3dl-02.pdf
  - The idr contribution is: https://tools.ietf.org/pdf/draft-xu-idr-neighbor-autodiscovery-11.pdf
- Work recently completed at IEEE on extensions to Virtual Station Interface Discovery and Configuration Protocol (VDP, 802.1Q-2018 clauses 40, 41, and 43) extends VDP to cover IP addressing for split NVE of NVO3 (802.1Qcy-2019)
- Work in progress at IEEE on Auto Attach (P802.1Qcj) which is currently described for Provider Backbone Bridges
  - Open source for LLDP auto attach is at: https://github.com/auto-attach/aa-lldpd
  - Provides discovery of VID to I-SID mapping for BEBs attaching to servers
- New IEEE project on LLDPv2 (802.1ABdh)
  - Purpose is to extend LLDP to scale existing LLDP applications, and add enhancements for router and TSN applications
  - The LLDPv2 project will be an amendment of 802.1AB-2016 (P802.1ABdh)
  - The LLDPv2 project will allow LLDPv2 databases consisting of multiple frames
  - LLDPv2 and LLDPv1 nodes will interoperate as though they were LLDPv1 with a single frame database
  - LLDPv2 should be sufficient to fill most discovery needs without the additional protocols

Objectives for New LLDPv2 Method

- Support LLDP databases larger than a single frame
  - Optimize LLDPv2 for databases around 100K bytes
  - For reference IETF currently believes database sizes around 64K bytes are sufficient
- Support the ability to limit the LLDP frame size to meet timing constraints imposed by some TSN applications
  - Do we need to split TLVs over multiple PDUs?
  - How big do these databases need to be?
- Support the ability to communicate with an LLDPv1 implementation
  - Only the LLDPv1 database would be exchanged between and LLDPv1 and LLDPv2 implementation
- Support shared media, optimize for point-to-point though allows shared
  - Duplicate MAC addressing should be handled by the extension protocol
- Ensure the integrity of the full set of TLVs received by partners
  - Do we also need to provide a means to authenticate the LLDP database? The IETF has this requirement.

Objectives for New LLDPv2 Method

- Support pacing of PDUs to receivers to prevent overloading low level network firmware
  - Historically OSPF and IS-IS have had problems from lack of flow and congestion management
- 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
- Other optimizations and considerations which might be useful
  - Computational load requirements for LLDPv2 receivers to update and validate PDUs
  - Larger TLVs or is using multiple TLVs appears sufficient
  - TLVs spanning multiple extension database PDUs, is this required for TSN
  - Database authentication, is high want for IETF and other applications
    - Part of separate authentication extension
    - Key exchange requirements

Extending LLDP Agenda:

–Introduce New Definitions

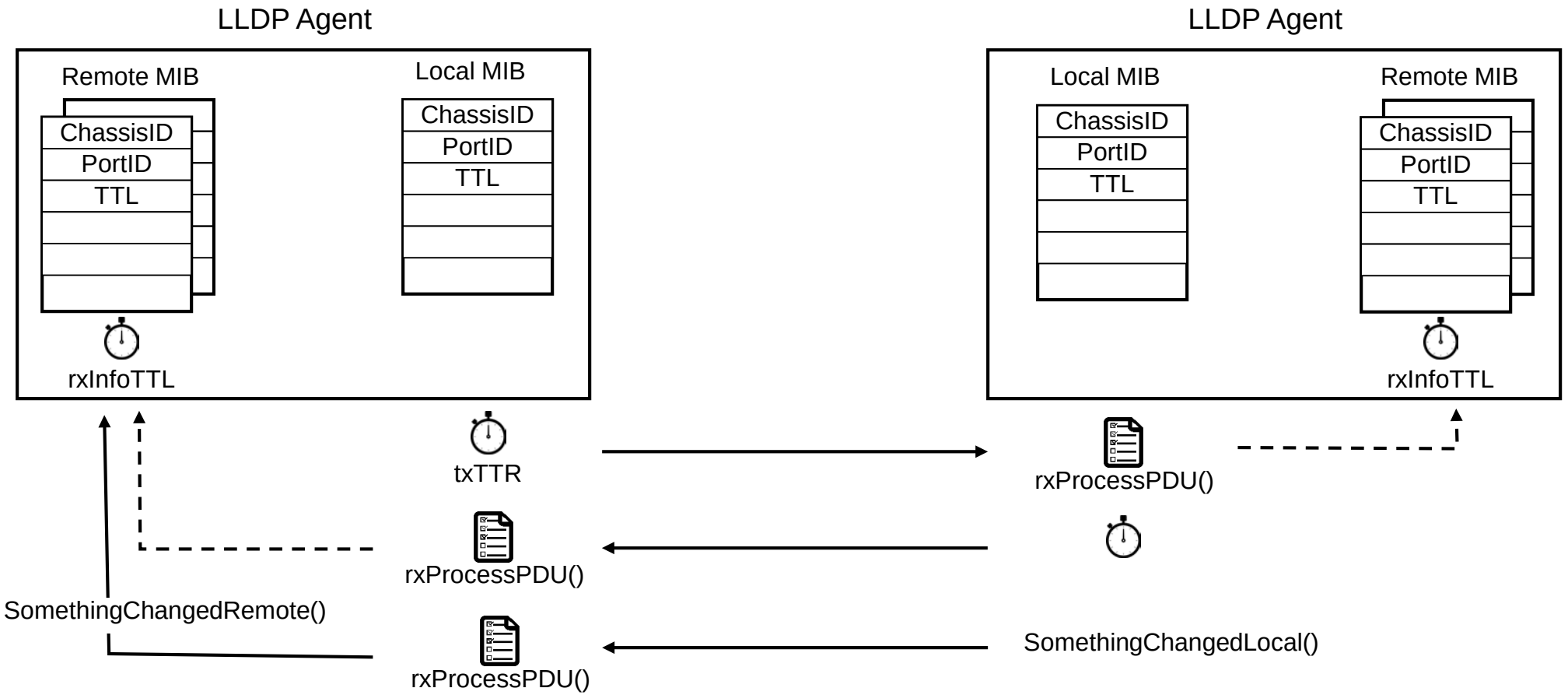
–Motivations and Objectives

–Link Layer Discovery Extension Protocol Principles

–Shared CSMA/CD Ethernet Worst Case Operation

–Summary and New Definition Discussion

Current LLDP Operation



NOTE: Remote and Local MIBs are databases that must fit within a single PDU length PDU
Replace all values of the Remote MIB with contents of LLDPDU when something changes

Proposal: Foundation PDU (F-PDU)

- The current LLDPv1 PDU with a Manifest TLV is the foundation PDU (F-PDU)
- The foundation PDU is exchanged using the existing LLDPv1 protocol without modifications
- All databases are created as LLDPv1 databases, no extension PDUs create new databases
- An extended LLDP database is composed of the foundation PDU and n-1 extension PDUs
- A manifest TLV placed in the LLDPv1 foundation PDU identifies all extension PDUs
- If no manifest TLV is present in the foundation PDU then no extension PDUs exist for the LLDP database
- The upper limit to the number of PDUs is determined by the LLDPv1 TLV size limit (512) and the format of the manifest TLV
  - Note: When we have a small max PDU size the manifest TLV size can be further limited resulting in limiting the database size
- The manifest TLV carries an identifier for each extension PDU
- Any change in an extension PDU is reflected as a change in the manifest TLV, therefore
- Any change in an extension PDU will result in a change to the foundation PDU

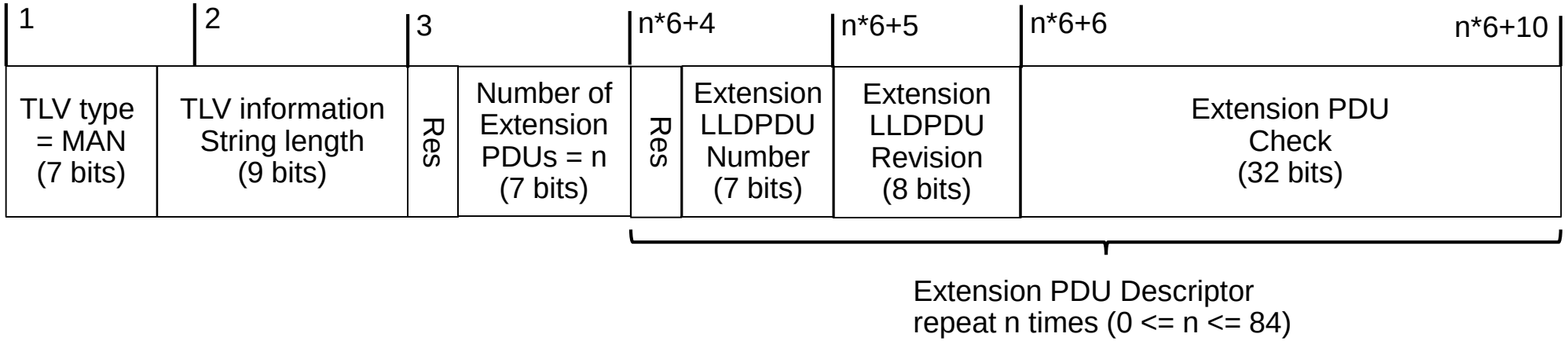
Proposal: Extension PDUs (X-PDUs)

- The extension LLDPDU will be ignored by LLDPv1
  - An alternate Ethertype is used for LLDPv2 PDUs to guarantee PDUs are never directed to LLDPv1
- Each extension PDU has three mandatory TLVs in the beginning of the PDU:
  - Each extension PDU contains the first two mandatory TLVs of a LLDPDUv1 (ChassisID + PortID)
  - Each extension PDU contains a new extension TLV that identifies the PDU
  - Before an extension PDU is added to a database it's {ChassisID, PortID, ExtensionID} must match the manifest TLV
- Each extension PDU is transmitted as a unicast in response to a receiver request
  - Extension PDUs are only transmitted in response to requests
  - The DA of an Extension PDU is the SA of the request
- The TTL in foundation PDU relates to all extension PDUs

Proposal: Extension Request PDU (XREQ-PDU)

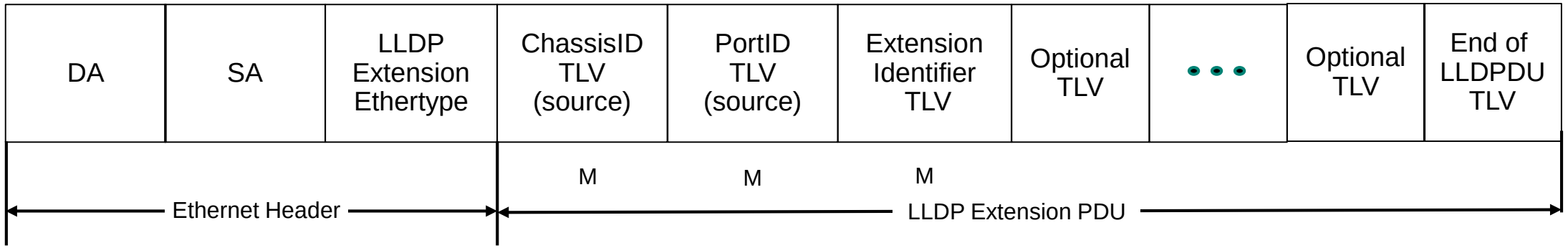
- The extension Request PDU will be ignored by LLDPv1
  - An alternate Ethertype is used for LLDPv2 PDUs to guarantee PDUs are never directed to LLDPv1
- Each extension request PDU has three mandatory TLVs in the beginning of the PDU:
  - Each contains the first two mandatory TLVs of a LLDPDU (ChassisID + PortID)
    - However the ChassisID and PortID are for the destination rather than the source
  - Each contains a new extension request TLV that identifies the PDUs a list of extension PDU to be transmitted
- An extension request (XREQ-PDU) is sent between peers to request transmission of an extension PDU
  - The LLDP extension protocol supports multiple peers on a shared media
  - Transmission of X-PDUs is only in response to an XREQ-PDU generated by the receiving system
  - A receivers requests X-PDU transmission when it determines the current X-PDU does not match the manifest TLV
  - Receivers can have only a single XREQ-PDU pending at a time
  - A single XREQ-PDU can request transmission of multiple X-PDUs
  - The receiver controls the transmission rate by controlling the number of X-PDUs requested and the timing between XREQ-PDUs
  - Receivers time out the requested X-PDU responses
  - Transmitters periodically send the foundation F-PDU which can update the manifest TLV in turn resulting XREQs for X-PDUs
- Each extension request PDU is transmitted as a unicast
  - The DA of an extension request PDU is the SA of the foundation PDU

Manifest TLV: Added to the Foundation LLDPv1 PDU



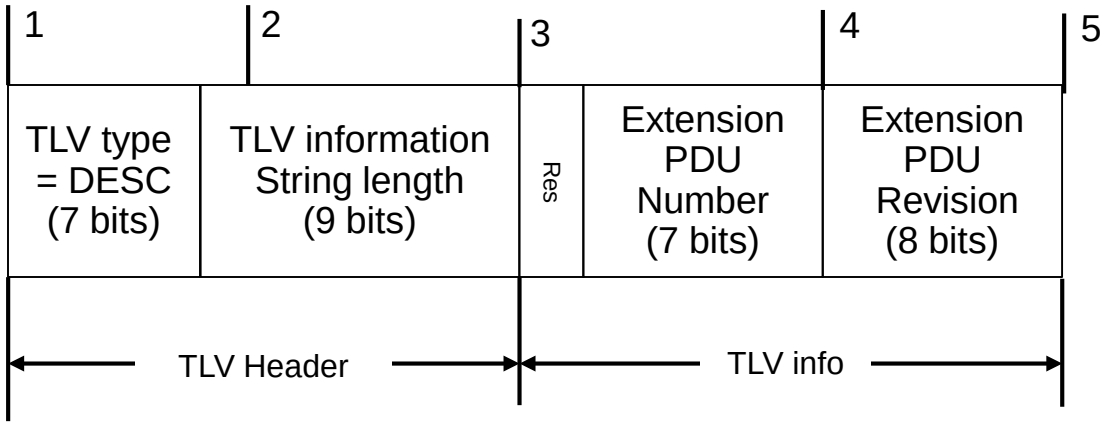
- Number of extension PDUs indicates the number of valid PDU descriptors in the manifest
  - Some implementations may fix the manifest TLV size however load it with a variable number of PDUs
  - If we don't need to hold the manifest TLV size constant, then the TLV length is sufficient to determine the number of manifest entries
- 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: for example 32 bits of MD5

Format for LLDP Extension PDUs (X-PDU)



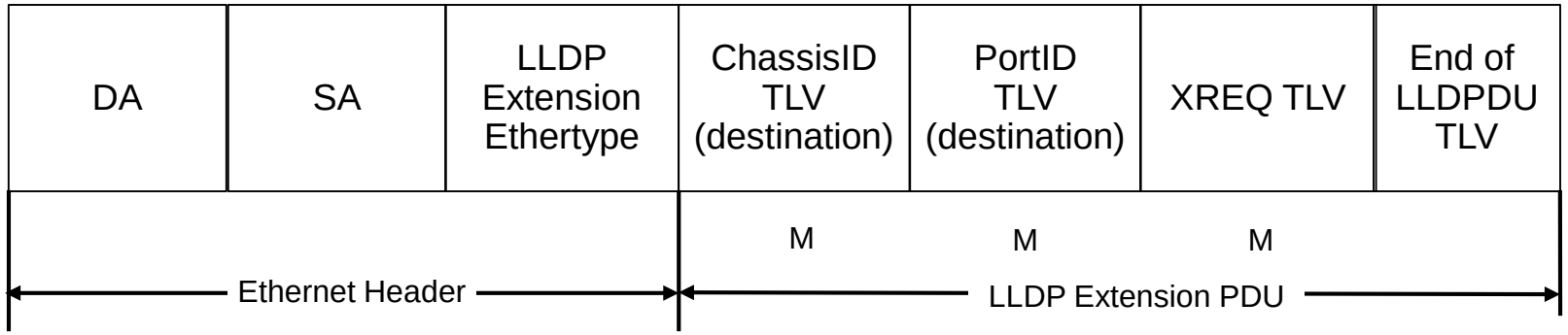
- LLDPv2 Ethertype
  - New LLDPv2 Ethertype for Extension PDUs prevents conflict with LLDPv1
  - Extension PDUs are identified by the presence of the Extension Desc TLV
  - Since extensions are not multicast and only delivered on request no new Ethertype is required, though one could be used if desired
- Chassis ID + Port ID are mandatory
  - The Chassid ID and Port ID of the PDU source
  - Note TTL from 1st PDU should apply and is not needed here
- Extension Identifier TLV is mandatory and must be the third TLV
  - Identifies this Extension PDU, the PDU revision

Extension PDU Identifier TLV (XID TLV):



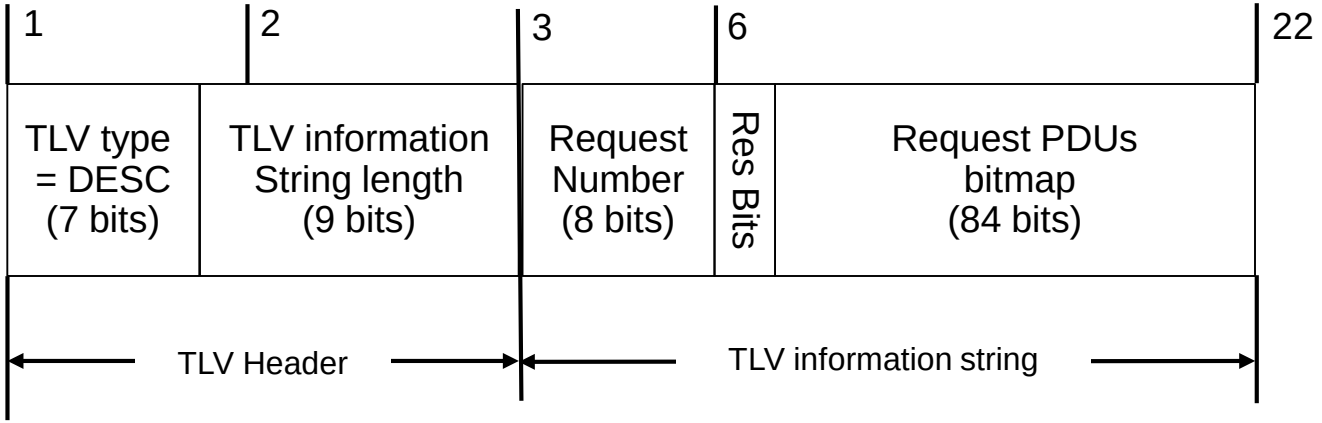
- Extension PDU Number is the designation number for this PDU
  - The PDU number is in the range from 1 – 84
  - Matched to the manifest extension PDU number
- Extension PDU revision number
  - Incremented modulo 256 whenever the extension LLDPDU is changed
  - Matched to the manifest to guarantee the 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

Request For Extension PDUs (XREQ-PDU)



- LLDP Extension Ethertype
  - New LLDP Ethertype for Extension PDUs to prevent conflict with LLDPv1 implementations
- ChasssisID and PortID TLVs are mandatory in a Request for Extension PDU
  - ChassisID is the first and PortID is the second TLV in the PDU
  - Unlike a standard LLDPDU the ChassisID and PortID identify the destination not the source
- Extension Request TLV is mandatory in a Request for Extension PDU
  - The Extension Request TLV is the third TLV in the PDU
  - Request PDUs are identified by the presence of the Request for Extension TLV

Extension Request PDUs TLV (XREQ TLV)



- Extension Request PDUs
  - A given chassis/port may only have a single XREQ TLV pending at a time
  - Multiple XREQs PDUs may be used to pace the PDUs at the receiver by withholding XREQs
  - A single XREQ PDU may request multiple Extension PDUs if the receiver has sufficient buffer for them
  - The bit map is used to identify the list of Extension LLDPDUs by number
    - The index to the bit map identifies the Extension LLDPDU number
- Extension LLDPDUs are not multicast, instead they are unicast
  - The extension LLDPDUs are sent to the SA address within the foundation LLDPDU
  - On a shared media each individual LLDP Agent must provide independent requests for extension frames
  - This allows the individual receivers to pace PDUs at rates that match their ability to handle the reception

Extending LLDP Agenda:

–Introduce New Definitions

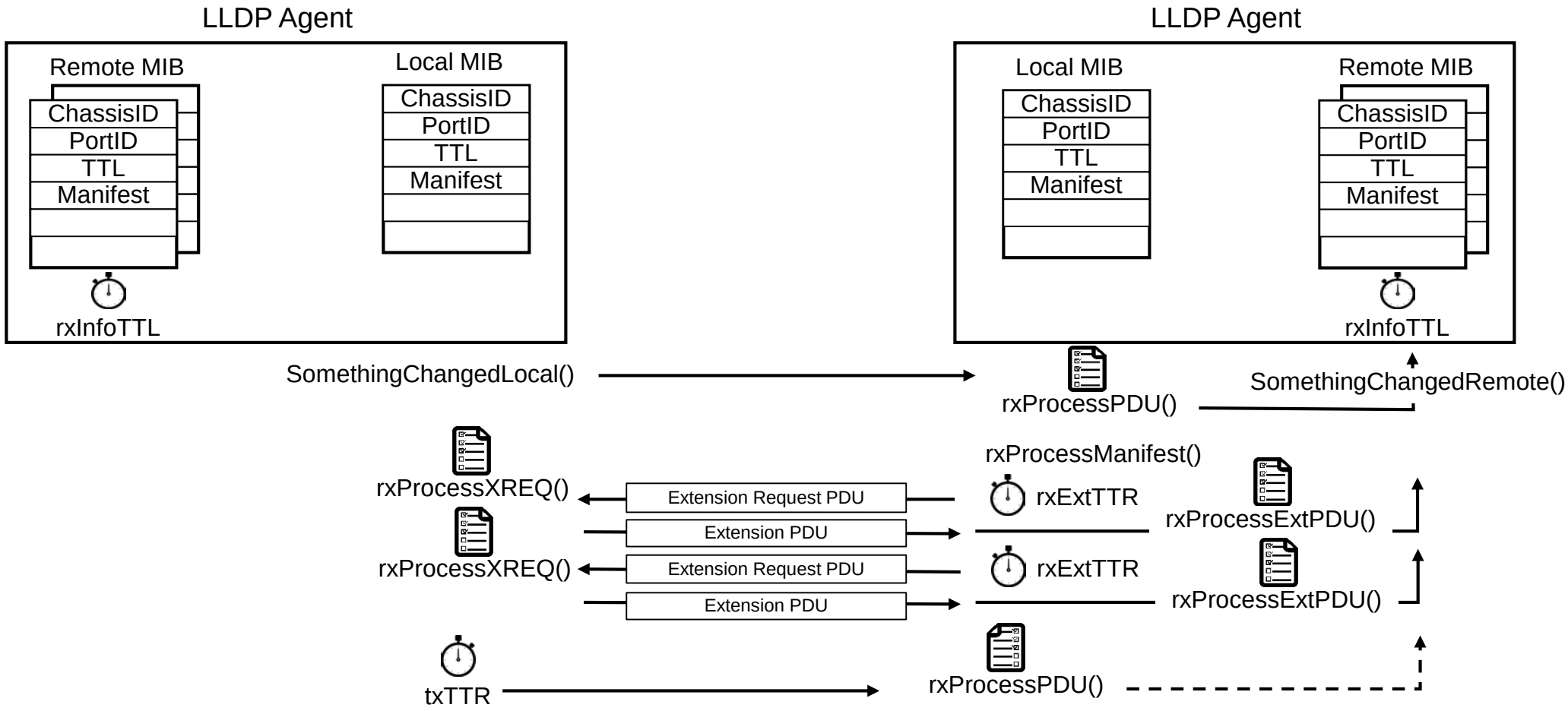
–Motivations and Objectives

–Link Layer Discovery Extension Protocol Principles

–Shared CSMA/CD Ethernet Worst Case Operation

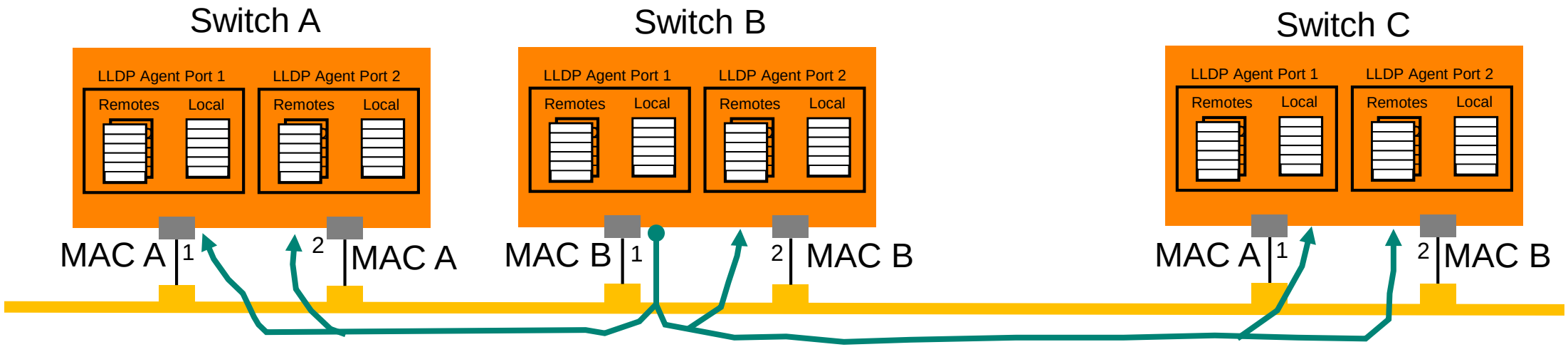
–Summary and New Definition Discussion

LLDP Extension Operation Proposal: Receiver Pacing



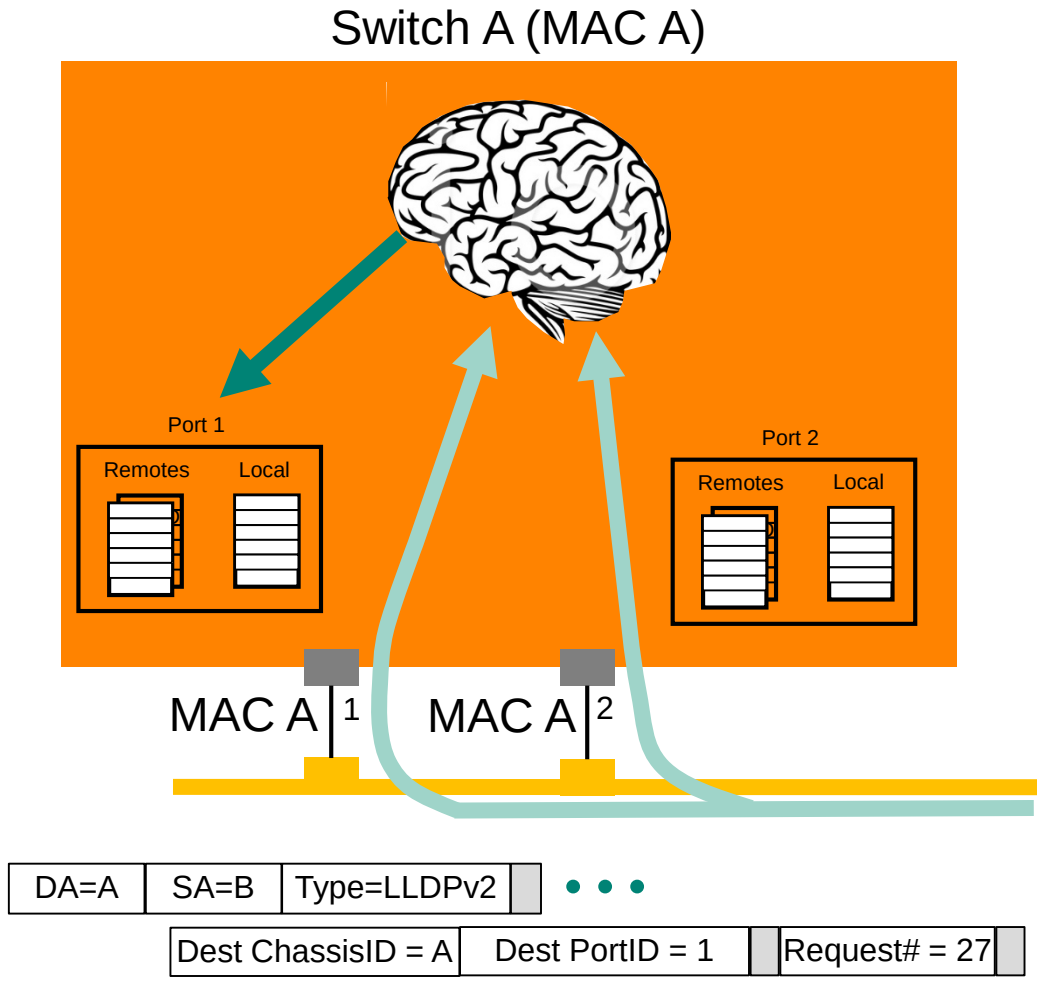
NOTE: Send LLDPDU as specified by LLDPv1 when something changes and periodically
Only send extension LLDPDU when explicitly requested by a XREQ
Only issue XREQ when manifest shows the local copy is out of date

Example Duplicate MACs on CSMA/CD Ethernet



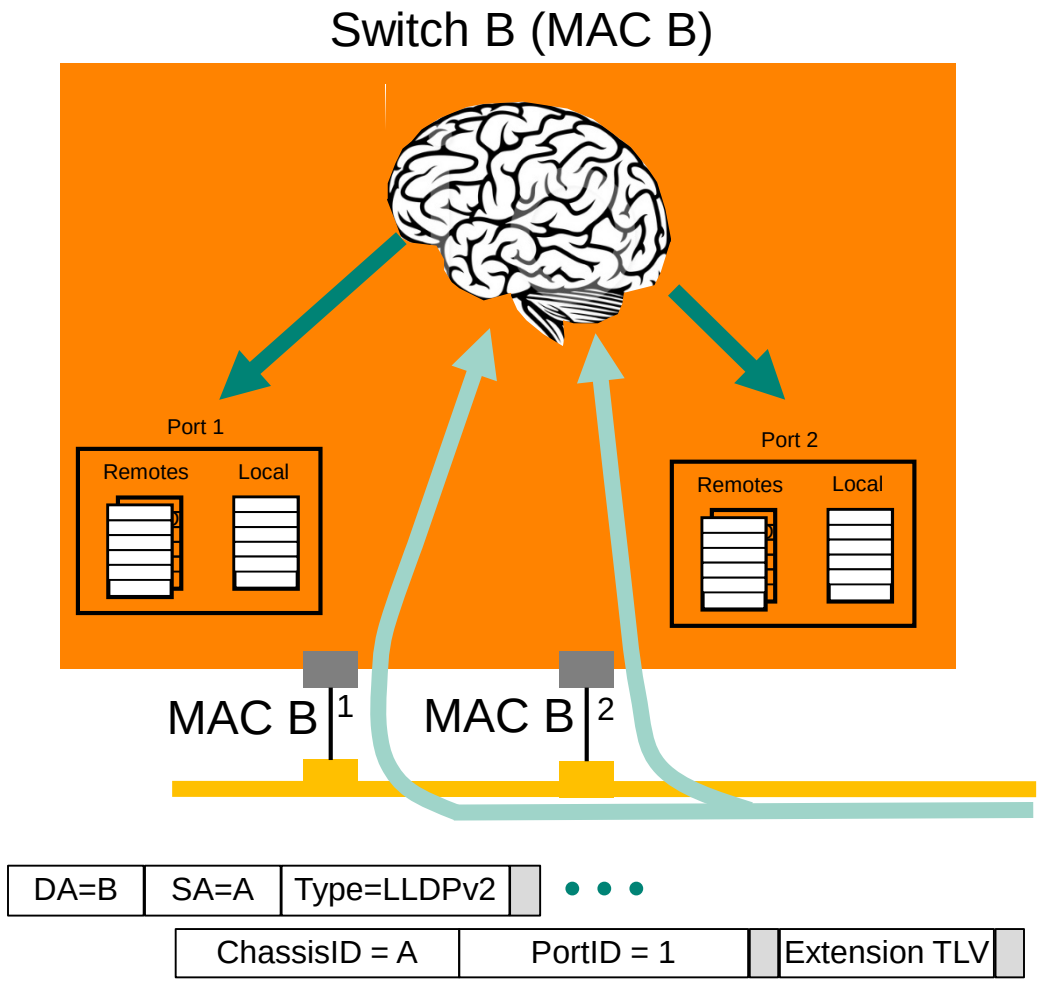
- In this example we have duplicate unicast port MACs both on the same switch and other switches
  - This example was chosen as a worst case example
- CSMA/CD delivered all frames to all ports connected to the LAN regardless of the destination address
  - We have other types of shared media, for instance and WiFI, token rings, EPON which may behave somewhat differently
  - On a CSMA/CD Ethernet end stations filter the delivered frames so they only receive unicast and multicasts programmed in the MAC
- Bridge ports operate as end stations directing frames to the bridge brain rather than the relay
  - IEEE specifies an internal MAC for each Bridge Port which is used as the source of control frames such as LLDP
  - Some Bridge implementations share a single unicast MAC between all ports
- On a shared media where it is possible to have duplicate MAC addresses we could receive a unicast transmission in multiple places

Shared Media, Extension Request PDU, Duplicate Switch MACs



- Here we have an Extension Request (XREQ) PDU addressed to bridge port with MAC A sourced from bridge B with source MAC B
- The bridge port MAC A is duplicated on ports 1 and 2 of the bridge
  - The bridge receives two copies of the XREQ
  - These two copies are directed to the bridge brain for processing
- The bridge brain uses the Destination ChassisID and PortID to determine what database the PDU is intended for
- The bridge brain uses the request number to filter out duplicates
- A single response is generated from the correct port
- If the XREQ was also delivered to some other bridge the destination chassisID would be used by the brain to filter it.
  - In the example switch C has a duplicate MAC A port.

Shared Media, Extension Response, Duplicate Switch MACs



- Here we have a Extension Response PDU addressed to bridge port with MAC B from bridge A
- The bridge port MAC B is shared by ports 1 and 2 of the bridge
  - The bridge receives two copies of the Response
  - These two copies are directed to the bridge brain for processing
- The bridge brain uses the source ChassisID and PortID to determine what databases the PDU is intended for
  - If no database matches the ChassisID+PortID the PDU is discarded
- The bridge brain checks to see if the PDU matches the manifest by comparing the PDU Number, the PDU Revision, and computing the checksum
  - If the PDU does not match the manifest it is discarded
- If the PDU matches the manifest and also matches the current extension PDU then it is discarded
  - Note: this discards any duplicate PDUs
- If the PDU matches the manifest, however does not match the current extension PDU, then the database is updated with the new PDU
  - Note: a single response PDU may update multiple port databases
- If another switch (i.e. switch C) also used a duplicate port MAC B and if switch C also stored the same ChassisID+PortID database with the same manifest TLV, then it also can update it's database as above.

Extending LLDP Agenda:

–Introduce New Definitions

–Motivations and Objectives

–Link Layer Discovery Extension Protocol Principles

–Shared CSMA/CD Ethernet Worst Case Operation

–Summary and New Definition Discussion

Summary

- A Manifest TLV in the foundation LLDPv1 PDU is used to specify the extensions
  - The Manifest TLV identifies the PDU number, revision, and a check code
  - Using an 7 bit PDU number, 8 bit PDU revision, and 32 bit check code allows up to 84 PDUs in the manifest
- A unicast protocol can operate in the face of duplicate MACs on shared media
  - The unicast protocol allows receiver pacing for PDU reception
  - The receiver may also control re-transmission for reliable delivery of extension PDUs
- Extension PDU updates are only required when the current revision is out of date
  - The foundation PDU which includes the Manifest TLV is updated periodically and whenever something changes
  - If any extension PDU changes then a change will occur in the Manifest TLV in the foundation PDU
  - LLDP determines if an extension PDU needs to be updated by comparing the manifest with the current database

New Definitions: Discussion

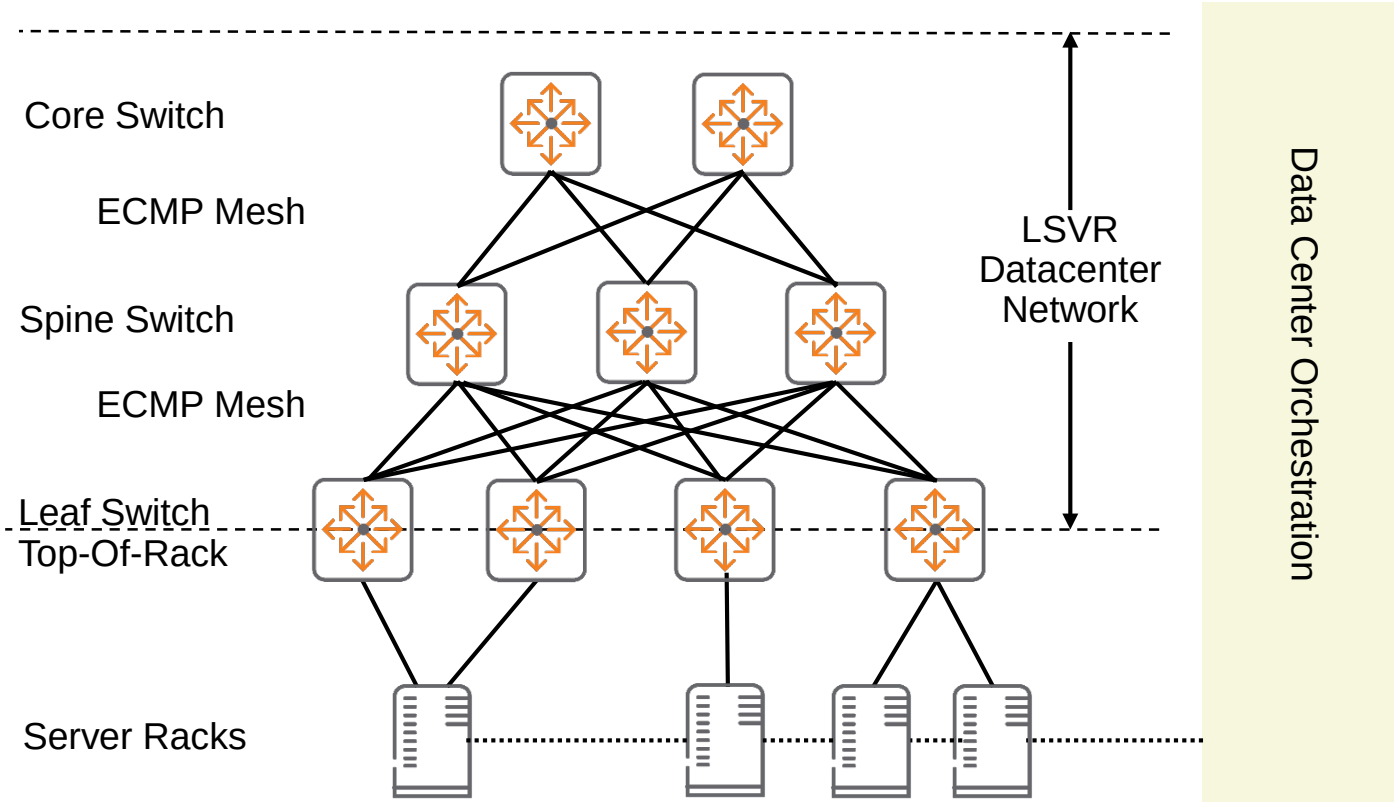
- **Link Layer Discovery Foundation PDU (LLDFPDU,F-PDU):** This is the single LLDPv1 PDU. In context this can be shortened to “foundation PDU” or F-PDU.
- **Link Layer Discovery Extension PDU (LLDXPDU,X-PDU):** This is an extension PDU for the LLDP database. In context this can be shortened to “extension PDU” or X-PDU.
- **Link Layer Discovery Extension Request PDU (LLDXREQPDU,XREQ-PDU):** This PDU is a request for transmission of one or more X-PDUs. In context this can be shortened to “request PDU” or XREQ-PDU.
- **Link Layer Discovery Extension Protocol (LLDXP):** This is the protocol used to exchange the extension PDUs of a multi-frame database.
- **Manifest TLV:** This is an LLDP TLV which describes each X-PDU of an Extended LLDP Database.
- **Extension PDU Identifier TLV (XID TLV):** This is an LLDP TLV carried in an X-PDU used to help identify the PDU.
- **Extension Request TLV (XREQ TLV):** This is an LLDP TLV carried in a RFXPDU and used to identify the requested X-PDUs.

Thank You

aruba
a Hewlett Packard Enterprise company

Backup Slides

Datacenter Network Using LSVR



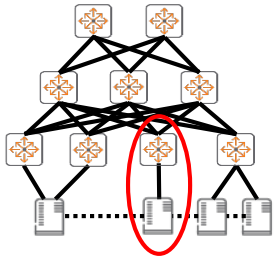
– Most datacenters are configured as 2-3 layer Clos networks using ECMP for distribution over the mesh and LAGs/M-LAGs for server attachment

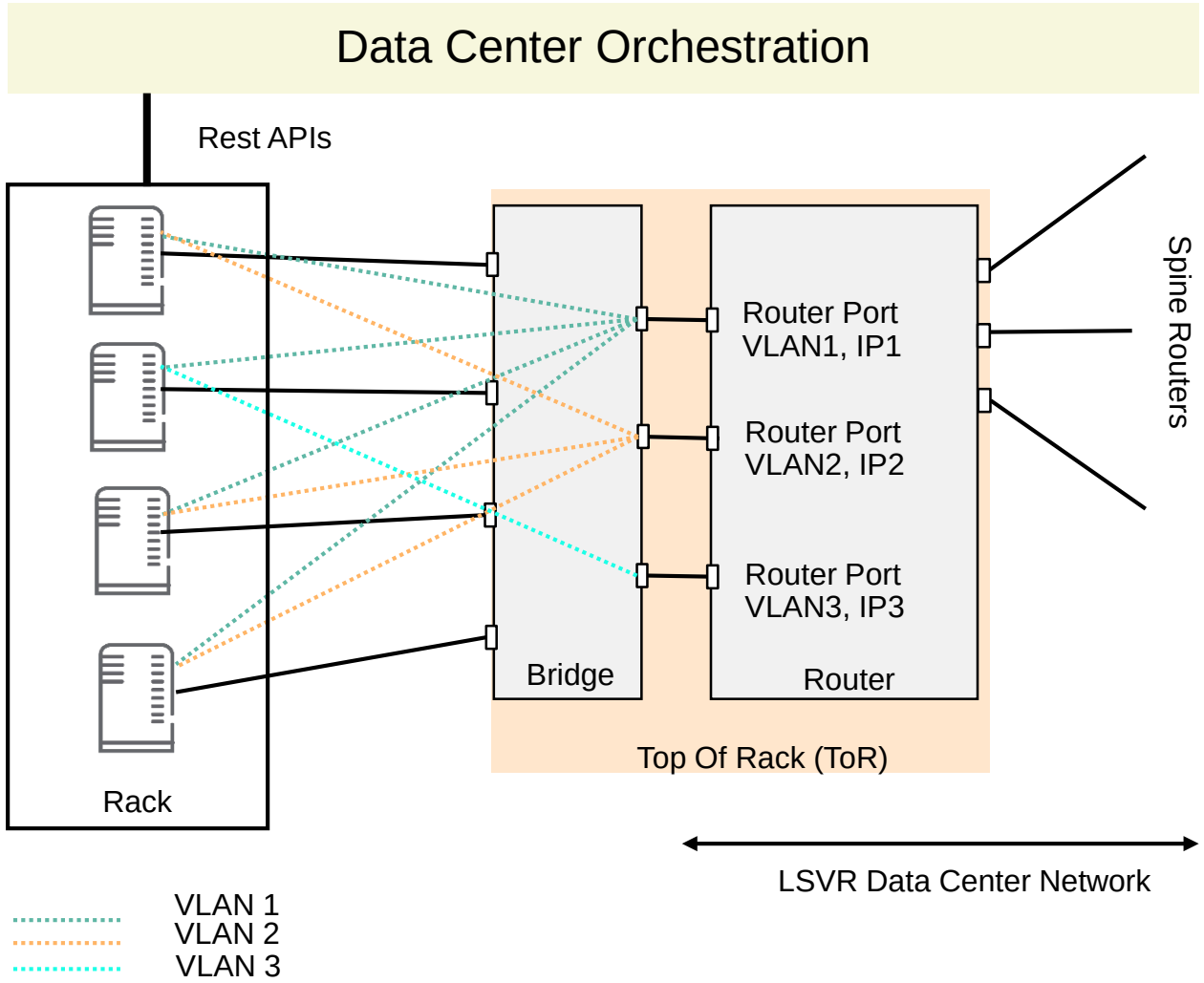
– Typically these networks provide an IPv4/IPv6 topology organized with ToR and Spine switches within Pods (around 8-128 racks)

– Servers at the network edge manage virtual and tenant networks which are encapsulated into the IP packets for transmission over the data center

– The orchestrator controls the creation of the virtual and tenant networks along with coupling to services

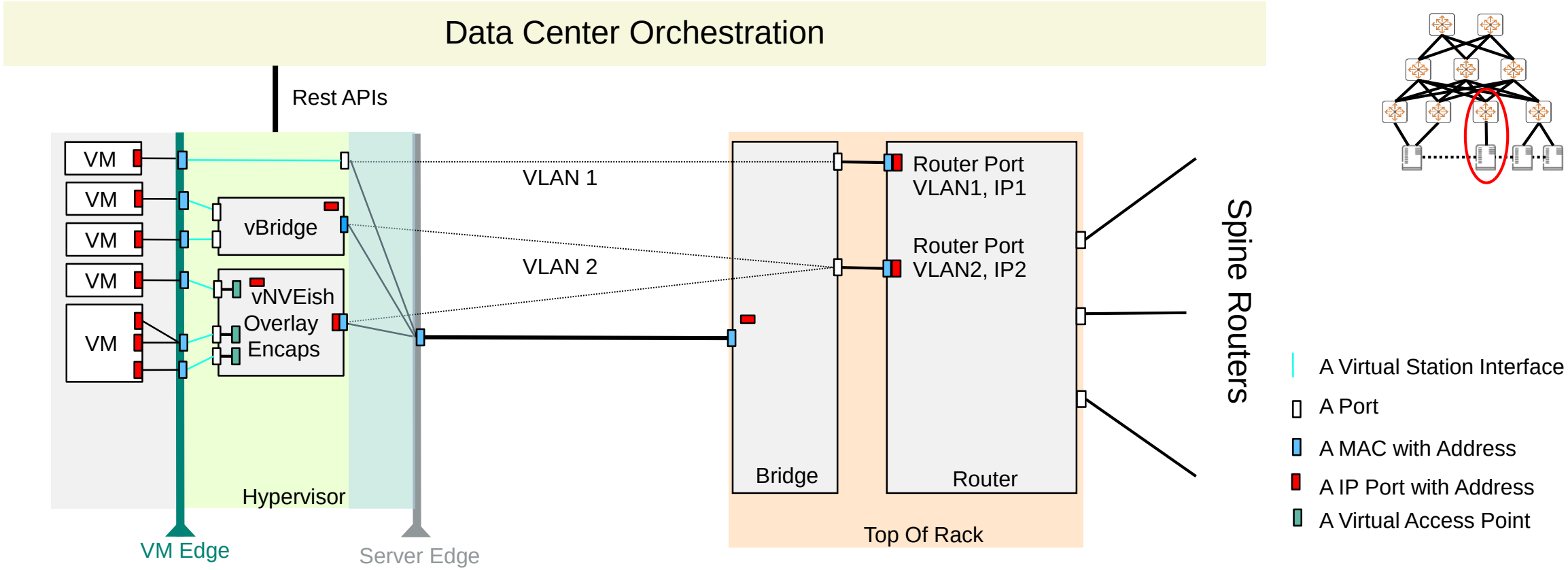
Typical Server and Switch Rack Configuration





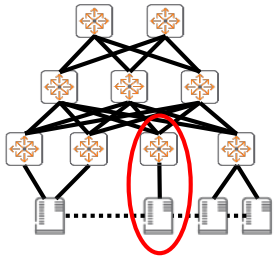
- Here the Bridge portion of the Top Of Rack Switch couples physical ports to each server in the rack
- Over the Bridge Ports VLANs are distributed to each server
- For each VLAN within the rack an IP subnet is assigned
- Each router port in the Top Of Rack is coupled to a single VLAN which is mapped onto an IP subnet
- Protocols within the switch (in this case LSVR) advertise the subnets available within the rack to the rest of the network

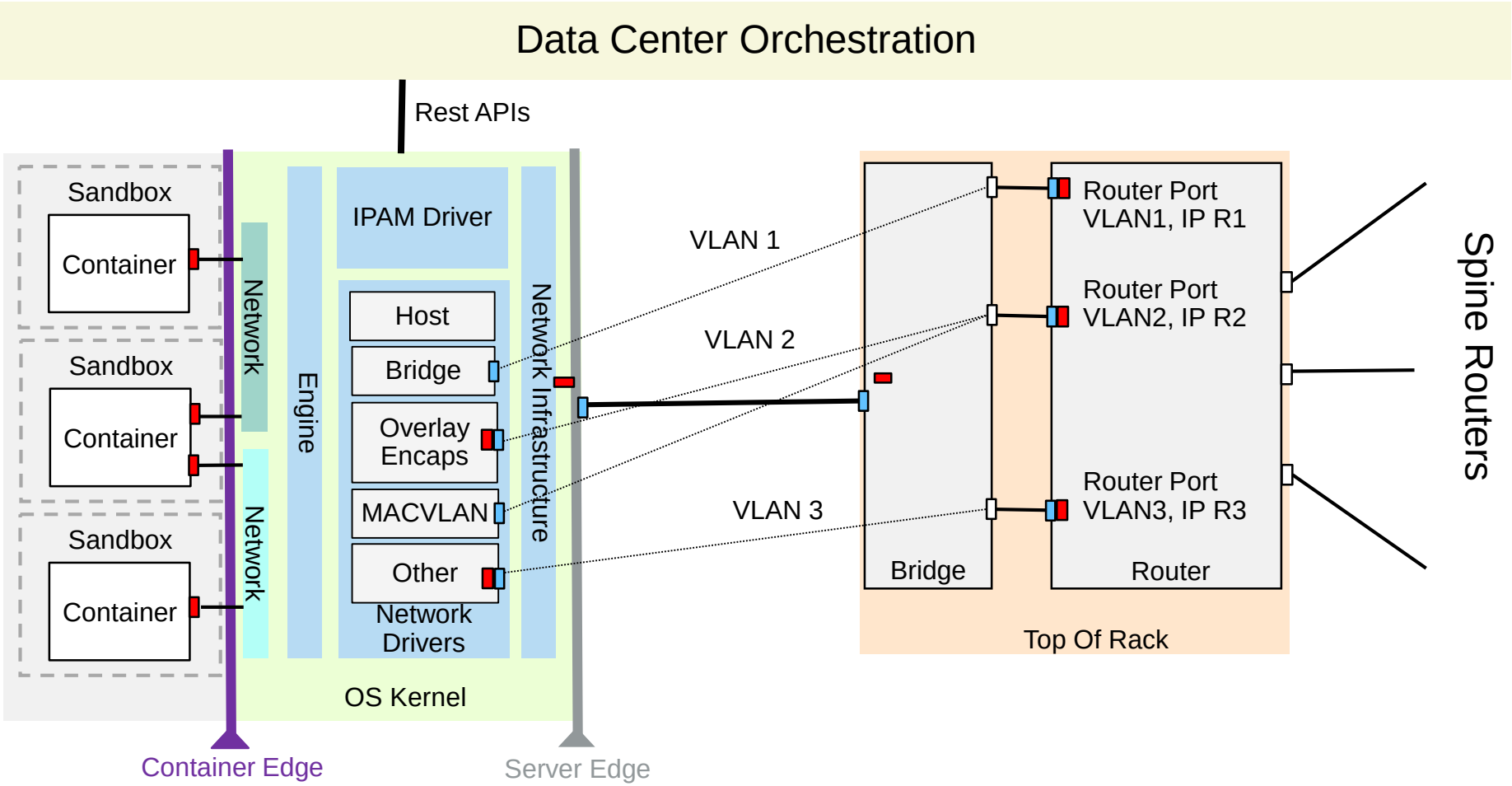
Server Network Interfaces – Virtual Machines (i.e. VMWare)



- **Virtual Station Interface (VSI, defined in IEEE Std 802.1Q-2018): is an internal LAN which connects between a virtual NIC and a virtual Bridge Port**
- **Virtual Access Point (VAP): A logical connection point on the Network Virtualization Edge (NVE) for connecting a Tenant System to a virtual network**
- **DC network is a simple IP underlay network. For scaling L3 encapsulations are supported using "NVE like" procedures within the server controlled by Data Center Orchestration**

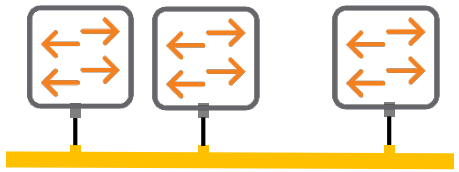
Server Network Interfaces – Containers (i.e. Docker)



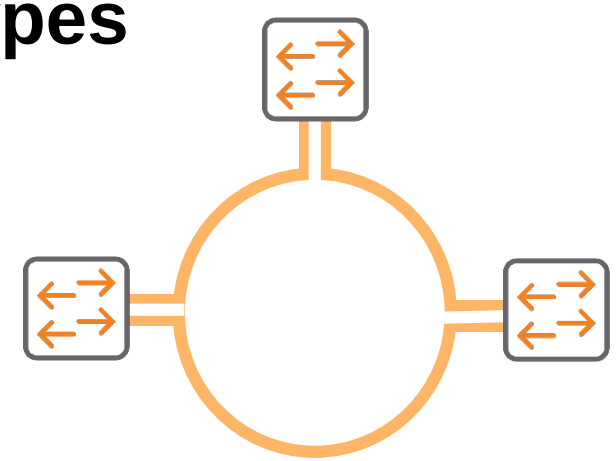


- **Container Solutions use Linux Namespaces and Groups to isolate containers**
- **These solutions provide a variety of network connections, though use an overlay for large scale datacenters**
- **DC network is a simple IP network. For scaling L3 encapsulations are supported using “NVE like” procedures within the server controlled by Data Center Orchestration**

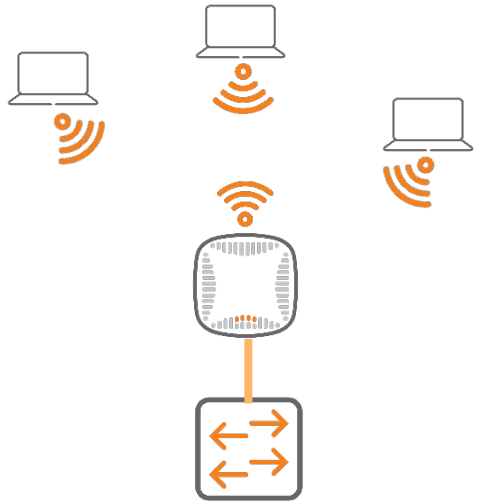
Shared Media Types



CSMA/CD



Token/Insertion Ring



Wireless