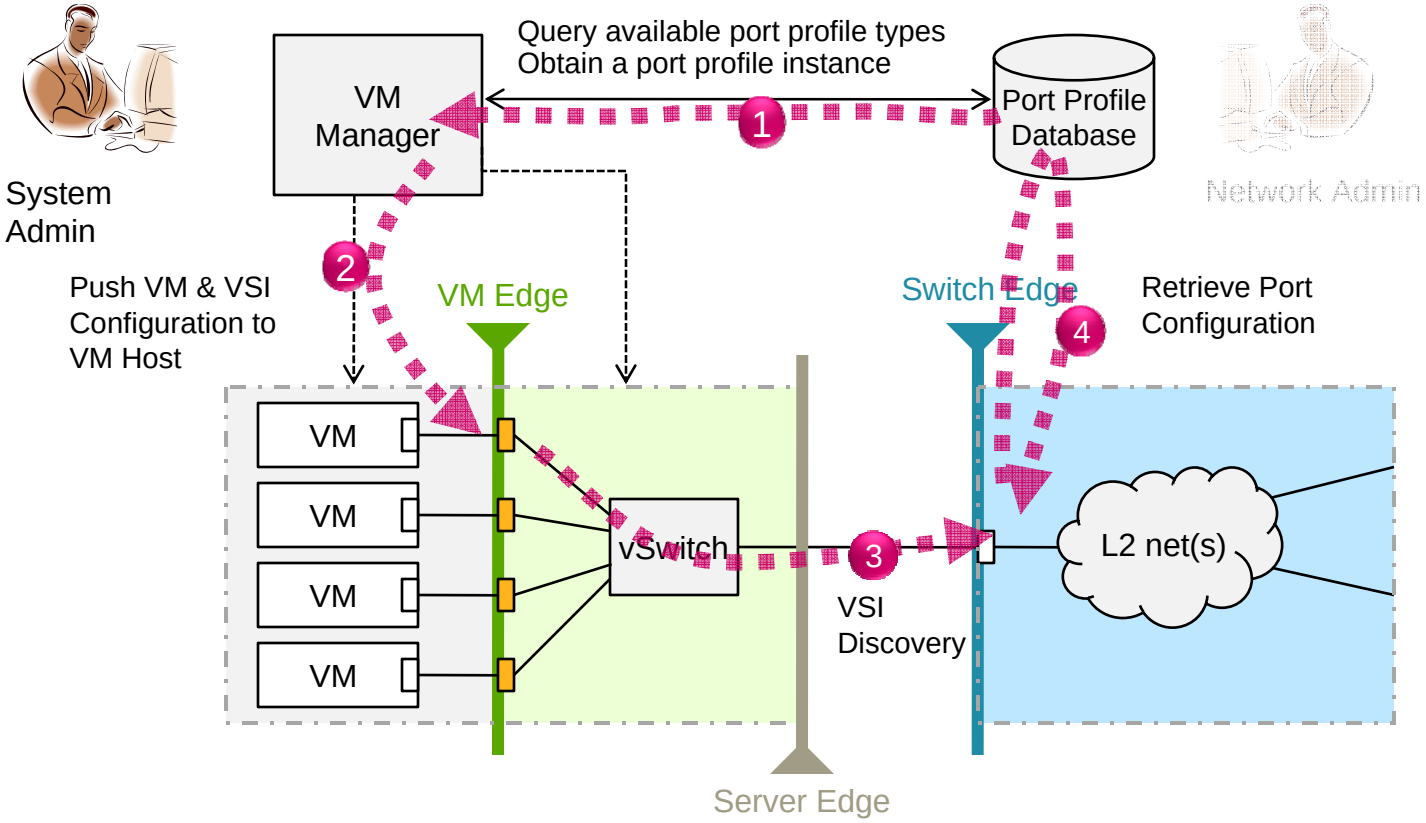
vPort (VSI) Discovery with LLDP

Uri Elzur, Broadcom

v31

12/1/09

Steps for Configuring Edge Connections (vPorts)



Legend: vPort (or VSI)

Assumptions

- A given VM Host (vSwitch, VEB, VEPA and combinations, etc) may feature
  - Multiple vPorts (e.g. 64 but may be up to few hundreds. But allow scalability to a larger number)
  - A small number of S-Component a.k.a. “channels” or none
  - A small number of uplinks i.e. physical NIC ports (e.g. 1, 2, 4, 8)
- A given physical server may have a small number of channels / NIC port
  - Each uplink (physical NIC ports) is discovered separately and runs its own LLDP and vPort Discovery session/s
- vPort state changes when a VM is re-deployed or its operational state changes (?), a relatively rare event
  - Ensure vPort Discovery is not the limiting factor for Power-up, Disaster Recovery

- LLDP
  - LLDP has frame size limitations (1500B)
  - LLDP frame contains the complete state of the sending party
  - More, on slide #8

- Physical Switch and NIC
  - Prefer to manage limited set of timers, state variables per LLDP session
  - Prefer to run one protocol (LLDP is ratified and std, used by DCBx) vs. two protocols (complexity, coordination, more work required to define and address all issues)

Requirements (Partial list)

- One protocol for Host to switch negotiation of all DCB parameters
  - PFC, ETS, QCN and evb/vPort (VSI)
- One LLDP exchange for the discovery and configuration of n vPorts per Channel
- Follow the DCBx approach
  - Bidirectional Symmetric and Asymmetric exchanges
  - Information in LLDP exchange triggers events for external State Machines dedicated to DCB
  - Use one way protocol. Delivery Confirmation by frequent retransmission
  - Use of Optional TLVs, designed to allow for flexibility and implementations variety

vPort Discovery Proposal – Key components

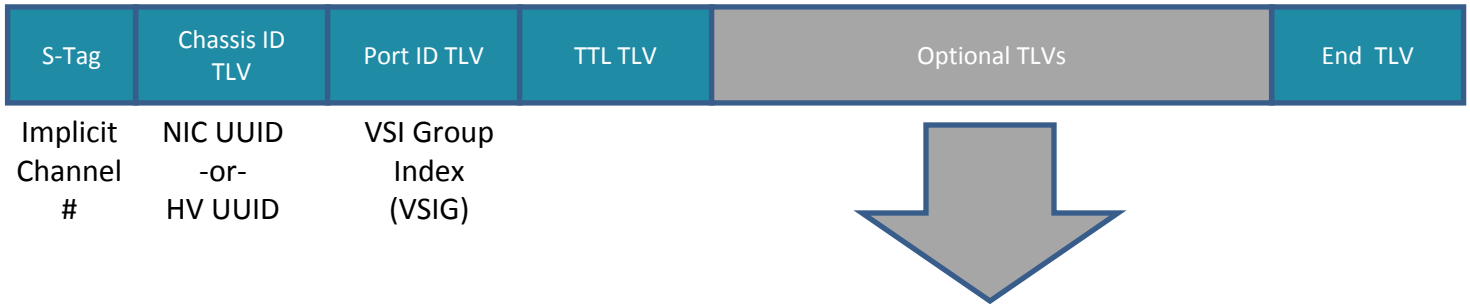
- One vPort Discovery state machine per “n” vPorts associated with the physical Port/S-Component channel vs. one per vPort
- Most of the vPorts are likely to be in a stable state
  - Either ASSOCIATED or DEASSOCIATED
- Communicate the State of all vPorts in every exchange
- Allow only a subset of the vPorts, on a given Port / S-Comp channel, to be in state changing mode
  - 1500B Frame may allow up to 64 vPorts to change state simultaneously
  - Can scale to a large number (e.g. 512) by using multiple S-Component channels (which increases LLDP frame sending rate as well)
- Quick vPorts/Port reset: Allow taking down the whole Port / S-Component channel in one exchange
  - Switch can de-associate any number of vPorts by flipping the State bit to DEASSOCIATED, no need for per vPort TLV for that

Conceptual Operational Description

- Each LLDP message, carries the vPort state of all vPorts on Port/Channel
  - Single bit per vPort if ASSOCIATED or DEASSOCIATED only, 2-3 bits for all states
- Host's evb entity (HV?) sends vPort Configuration TLV for up to "n" vPorts to be configured (n < SC_max)
  - Host continues to send the same vPort Configuration TLV/s with same vPort list, until it receives a change in vPort state response from the switch
- Switch may reply with different states for some of the vPorts above
  - Some vPorts will go to ASSOCIATED, some may take longer or no resources
- Host may mirror Switch message by sending the original vPort Discovery TLV updating state for vPort/s, where Switch's message indicates a state change [OPT]
- As the state for some of the n vPorts, requesting state change, has changed to ASSOCIATED or DEASSOCIATED, Host may replace some of the vPort/s in the vPort Configuration TLVs

EVB-vPort Discovery LLDP Frame format

Frame format for per channel exchanges [B]



EVB Status –

- CONFIG REQUEST
- CONFIGURED
- CONFIG REQUEST NACK
- RESET REQUEST
- RESET

vPort State*** –

- ASSOCIATE REQUEST
- ASSOCIATED
- ASSOCIATE REQUEST NACK
- DEASSOCIATE REQUEST
- DEASSOCIATED

** In some cases SC-max can be equal to V-max*

*** for 1024 vPorts*

**** Can be extended to incorporate additional states*

	Bytes
Channel Configuration TLV –	
•S-Tag number *(optional, NULL in case multi-channels mode is not used)*	2
•EVB Type (VEPA | VEB | VNIC | …)	1
•EVB Type Status	1
•Max number of vPorts supported *(V-max)*	2
•Current number of vPorts for this channel *(V-n)*	2
•Max number of State changing vPorts supported *(SC-max)**	2
•Current number of TLVs for State changing vPorts for this channel *(SC-n)*	2
vPort Configuration TLV 0 –	
•vPort index *(0 … V-n)*	2
•PPID	4
•MAC address	6
•VLAN list	4
•vPort State	1
vPort Configuration TLV 1 –	17
:	
vPort Configuration TLV X – where X < *SC-n of* State Changing vPorts	17
vPort Stable State TLV *(0, V-n)* – $A/!D_0$, $A/!D_1$, …, $A/!D_{V_n}$	< 128**

EVB-vPort Discovery LLDPDUs per Port/Channel

vPort Created

1. The Host, initiates configuration of up to SC-max new vPorts with the vPort configuration TLVs
   - Each contains: vPort ID, PPID, MAC address, VLAN list and status set to ASSOCIATE REQUEST
2. The Host also sends a vPort Stable State TLV with a 1 bit state per vPort (ports in the change state show as previous-state till new-state is confirmed)
   - State represented as 1 bit per vPort, consume 128 Bytes only for 1024 vPort on a single channel
3. The Switch responds with vPort configuration TLVs including the ASSOCIATED state for all vPorts it had resources for and had completed configuration for
   - It can optionally fetch the PPID properties from DB at this point (or pre-fetch all at boot-up)
   - Switch may reply with PRE-ASSOCAITE or ASSOCIATE REQUEST NACK in case it has no resources
4. The Host sets local status to ASSOCIATED for all vPorts that got acknowledged by the Switch
   - Reflected in next vPort Stable State TLV sent
5. GO TO #1: The Host initiates configuration of up to SC-max new vPorts with a new vPort configuration TLV
   - Resend for those vPort/s where an ASSOCIATE has not yet been received from the Switch and for any new vPorts up to SC-max

vPort Deleted

- If any party (?) wishes to delete a vPort, it uses one of the vPort Configuration TLV (n < SC-max) and sets its state to DEASSOCIATE REQUEST
- The other party can respond with status set to DEASSOCIATED and both sides can delete the vPort
- First party resends the whole vPort Discovery LLDPU. For all vPort where Other Party changed state:
  - Removes these port's vPort Configuration TLV from the vPort Discovery LLPDU
  - Resets the A/!D bit for the relevant vPort in the vPort Stable State TLV
- Both parties update setting, e.g. VLAN; if the VLAN has no port membership, the VLAN is deleted

Notes

- If no vPorts are going through state change, the value of SC-n in the Channel Configuration TLV is 0.
- Prior to discovery, each party sets all A/!D bits for all vPorts to 0 – i.e. DEASSOCIATED.

LLDP Disclaimer

- Proposal honors the IEEE802.1AB (LLDP), implicit implementation assumption, that each link partner needs to store one Ethernet Frame and use one timer
- Optional TLVs do not affect switches that are not supporting Virtualization / evb
- Similar to the way DCBx uses LLDP, additional state may be stored/impacted using other state machine/s that DO NOT AFFECT the LLDP resources
- Proposal assumes all the Port/Channel state is present in each vPort Discovery LLDPU
  - The Stable State for vPorts i.e. A/!D is avail in each vPort Stable State PDU
  - The state changing vPorts in the vPort Configuration TLV may change over time
  - The LLDP state machine is NOT where the parties store the config info for each vPort e.g.
    - MAC
    - VLAN/s
    - PPID
    - Etc.
- Proposed approach for vPort Discovery LLPDU to contain both State of all vPorts and SC_max vPort/s with state change, provides a good trade off for bytes on the wire and flexibility.
  - Recommended regardless of vPort Discovery implemented over LLDP or another protocol

VLAN list per vPort

- Options:
  - Explicit – limited number
  - Bit map – expensive
  - VLAN Profile pointer – flex enough?
  - Extensible – one (two) explicit and a pointer to extension
- VLAN Profile
  - Assume another database has combinations of VLANs used by different vPort
  - A given VM serviced by a PP DB profile, may need a combination of VLANs
  - Different VMs with same PPID may need different VLANs, so VLAN may be outside of the PP
- Extensible
  - Have one or two explicit VLAN IDs and allow for extension into another VLAN DB if needed
  - Can use one bit in the two bytes to indicate extension or second VLAN
  - 4 bytes contains two VLAN IDs or one VLAN ID and one extension ID
  - Trades off cost and complexity

	Bits / VID	Max VIDs	Total [B]
Explicit VID	16 (12)	4	8
Bit map	1	4k	512
VP Pointer	16	4k	2
Extensible	16	4k	4

How many bytes avail for Optional TLV in LLDP ?

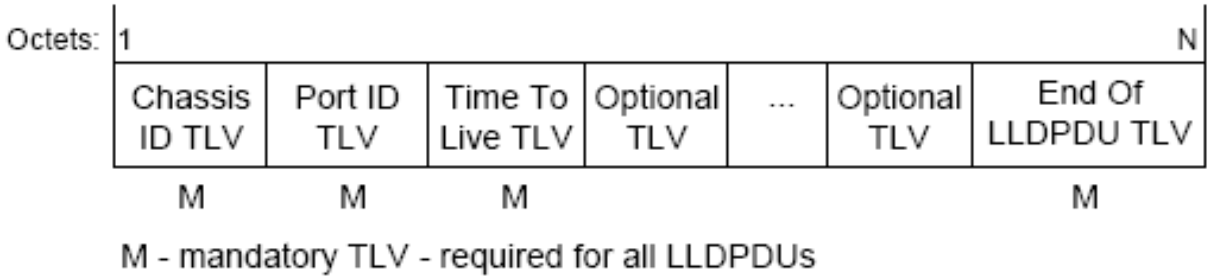


Figure 8-1—LLDPDU format

- Headers
- Mandatory TLV – 48B
  - Chassis ID: 3 Octets + ID (< 255 Octets). Assume 32B (similar to IPv6 address)
  - Port ID: 3 Octets + ID (< 255 Octets). Assume 6B (Ethernet MAC)
  - TTL: 4 Octets
- Optional TLV
- DCBX = 7 + 19 + 15 + 8 + (7 + 3x3*) + 9 = 64B
- With above assumptions ~1400B are available for vPort Discovery
- LLDP based vPort TLV for 1024 vPorts with 32 changing : 11 + SC-Max * 17B + 128B = 11 + 32 *17 + 128 = 11 + 534 + 128 = 673B