PFC Deadlock Free in Data Center Network

LLDP use case: Self-learning of switch type, level and port type in CLOS network

Xiang Yu

yolanda.yu@Huawei.com

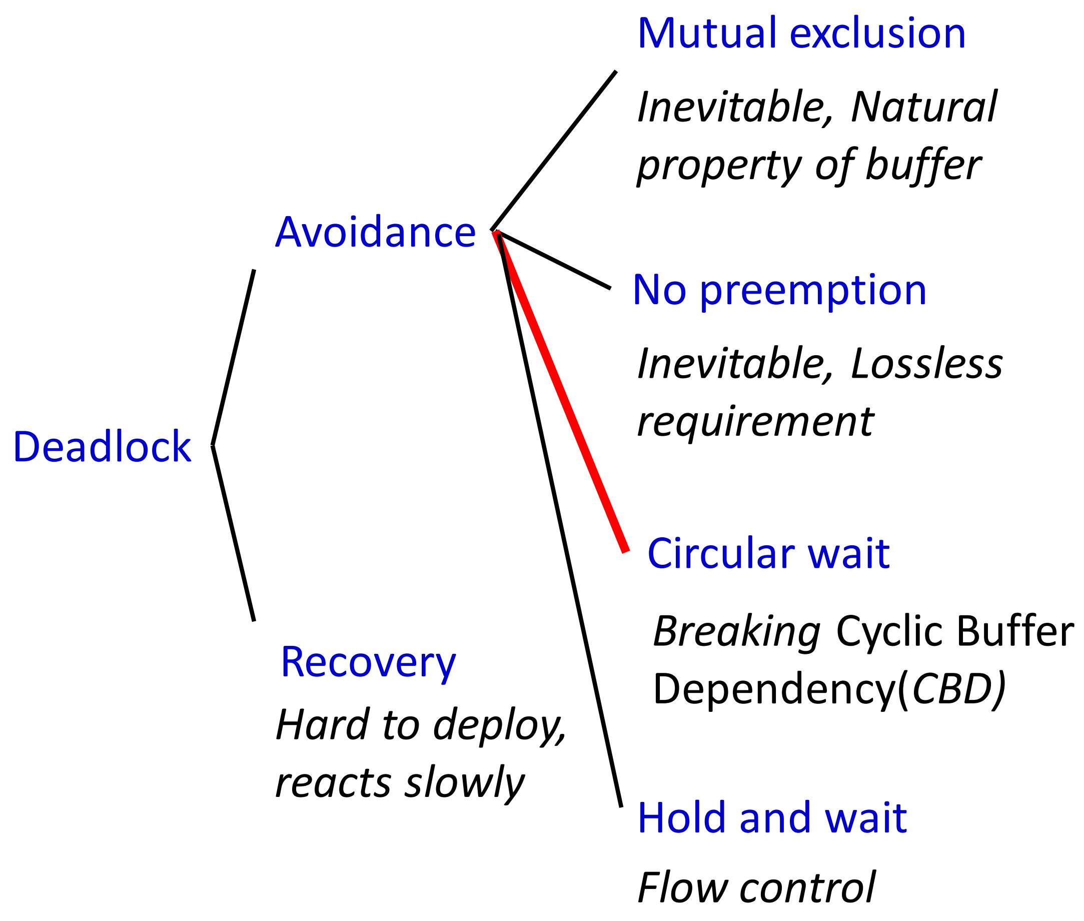
Edinburgh, UK, Sep. 2019

Purpose

- Define a new TLV to support more switch type, level and port information via LLDP protocol
- Understand the topology of the CLOS network. Self-learning of switch level and port type in CLOS network
- Application Use case: Avoid Cyclic Buffer Dependency (CBD) to prevent PFC deadlock problem in DCN

Background

- Remote Direct Memory Access (RDMA) over Converged Ethernet is used to pursue the required performance, such as ultra-low latency, high throughput and low CPU overhead in modern data centers.
- RDMA requires zero packet loss network where Priority-based Flow Control (PFC) must be used. PFC, however, makes Ethernet networks prone to deadlocks.
- There are four necessary conditions for deadlock occurrence[1]. To prevent deadlocks, we can ensure that at least one of the necessary conditions never holds.



Deadlock
Avoidance
Recovery
Hard to deploy, reacts slowly
Mutual exclusion
Inevitable, Natural property of buffer
No preemption
Inevitable, Lossless requirement
Circular wait
Breaking Cyclic Buffer Dependency(*CBD)*
Hold and wait
Flow control

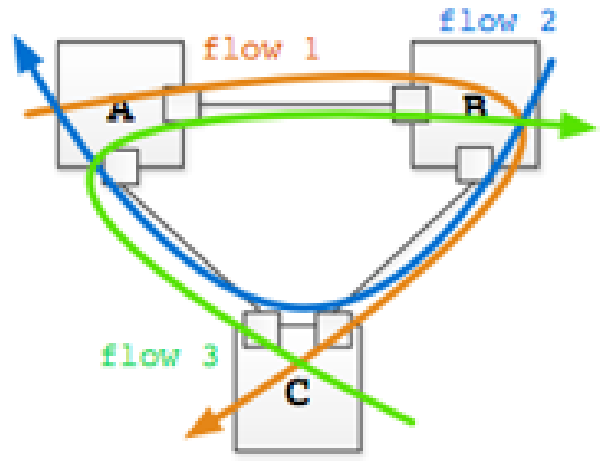
* The original figure is from [2]

[1] Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne. 2014. Operating system concepts essentials. John Wiley & Sons, Inc.
[2] Qian, Kun, et al. "Gentle flow control: avoiding deadlock in lossless networks." *Proceedings of the ACM Special Interest Group on Data Communication*. ACM, 2019.

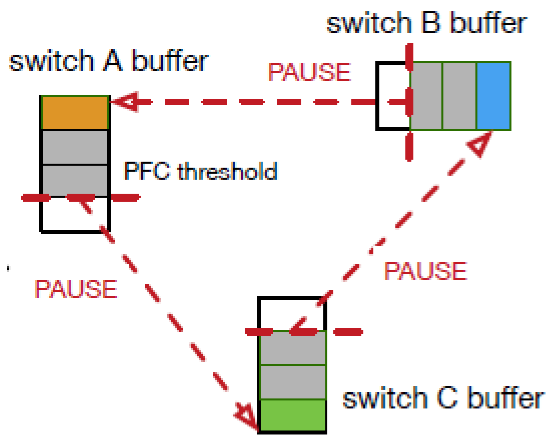
How does PFC deadlock form?

- CBD(Cyclic Buffer Dependency) is a necessary condition for deadlock formation
- Flow loop is a necessary condition for CBD

Flows Loop create buffer dependencies

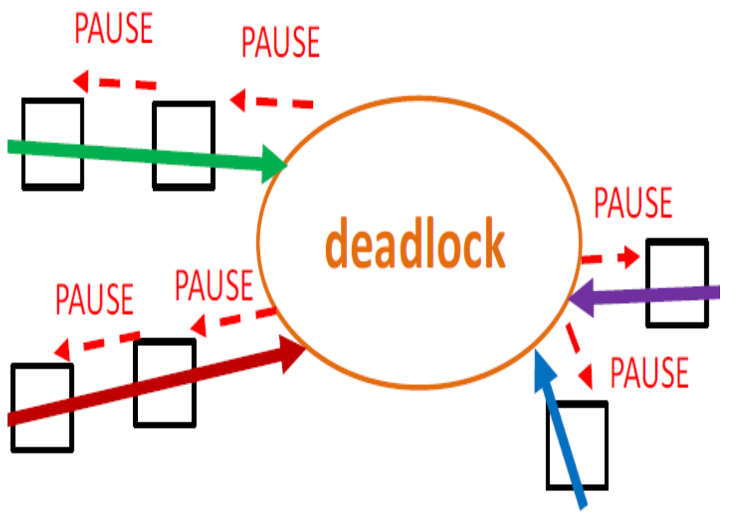


Traffic CBD



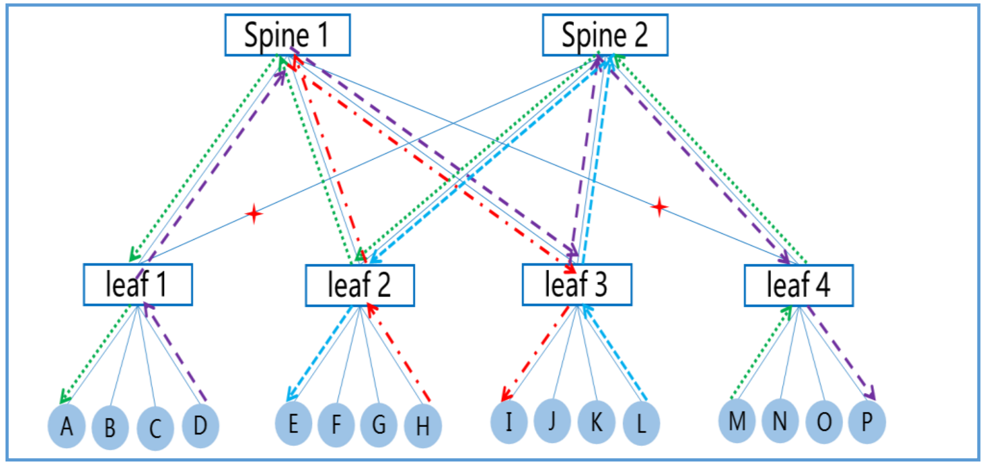
Cyclic Buffer Dependency:
A -> B -> C -> A

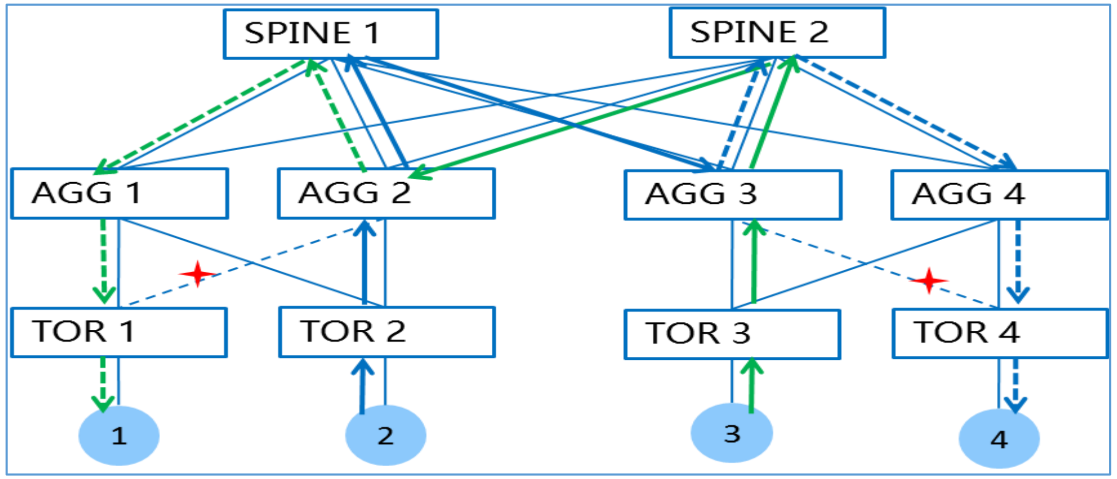
PFC Deadlock



Hu, Shuihai, et al. "Tagger: Practical PFC Deadlock Prevention in Data Center Networks." *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*. ACM, 2017.

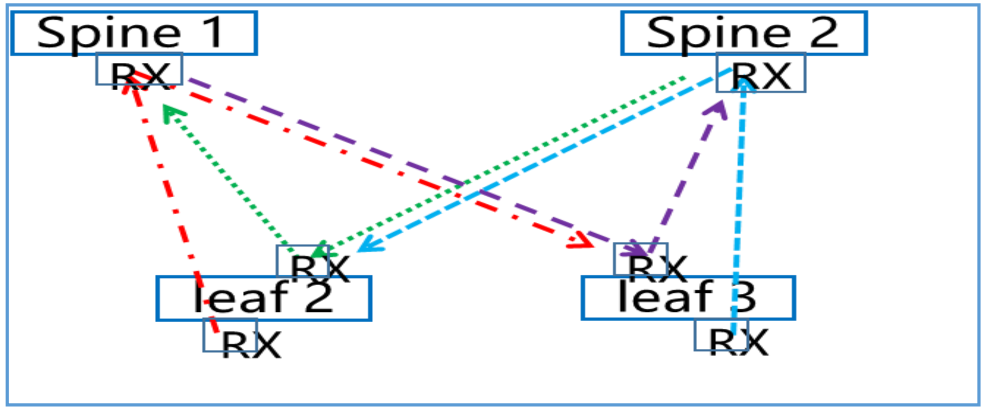
Reproduce the PFC deadlock problem

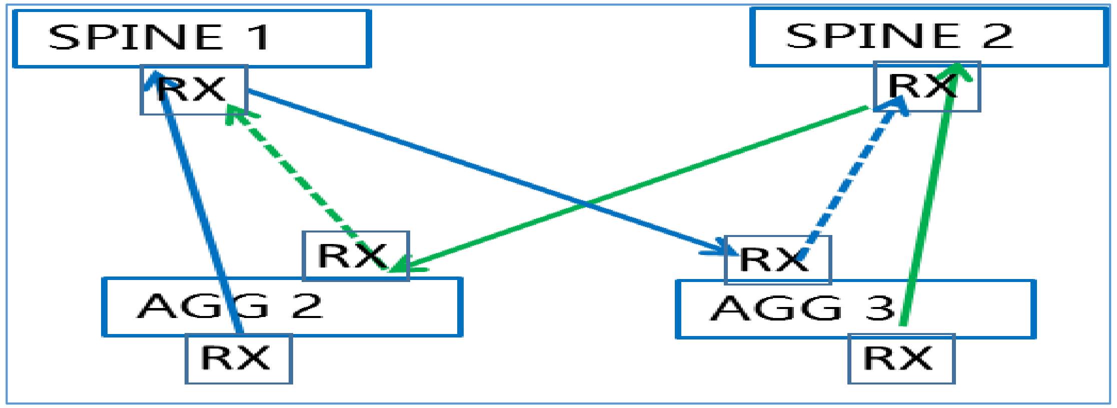




CASE 1 traffic flow: H→I L→E M→A D→P

CASE 2 traffic flow: 3→1 2→4





CASE 1 CBD: Spine 1→Leaf 3→Spine 2→Leaf 2→Spine 1

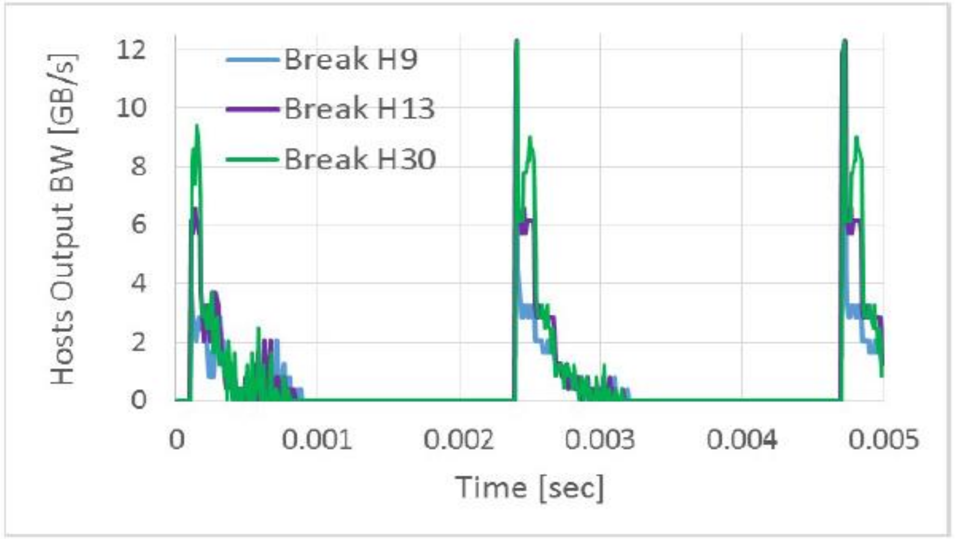
CASE 2 CBD: AGG 2→Spine 1→AGG 3→Spine 2 → AGG 2

- Reproduce the PFC deadlock in both level 2 CLOS and level 3 CLOS network.
- Although CLOS network does not have route loops, when link fails, flow loop happens and CBD appears. PFC deadlocks may happen.

Current mechanism

Two broad categories:

- **Reactive:** mechanisms/systems detect that a deadlock has formed, and then try to break it by resetting links/ports/hosts etc. May cause the network performance seriously.



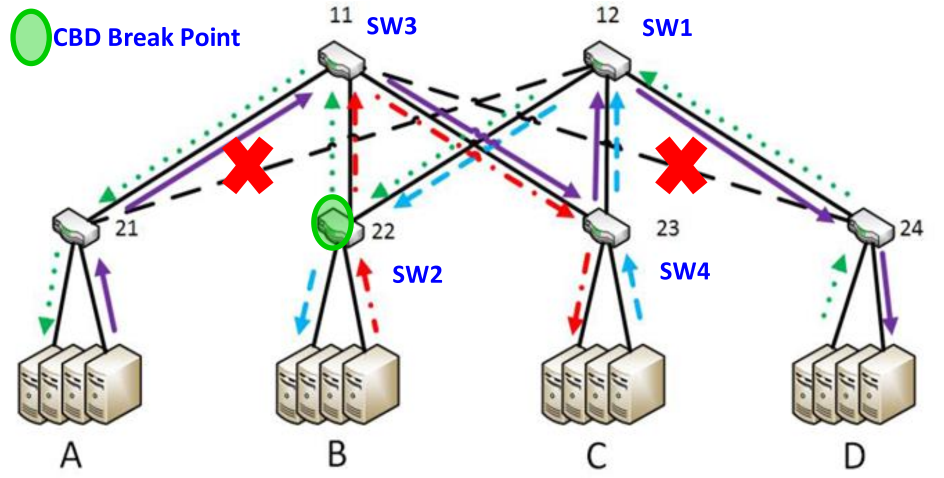
Shpiner, Alex, et al. "Unlocking credit loop deadlocks." *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*. ACM, 2016.

- When a queue continues to be in the PFC-XOFF state for a period of time, it is considered that a deadlock has occurred. Software will trigger to interrupt notification to perform deadlock recovery.
- The software allows the scheduler to ignore PFC-XOFF state of the deadlock queue for some time (configurable) and continue scheduling (send packet to the peer/direct drop the packet)
- If the CBD persists, then it will invoke deadlock immediately after recovery, and the throughput will be greatly affected.

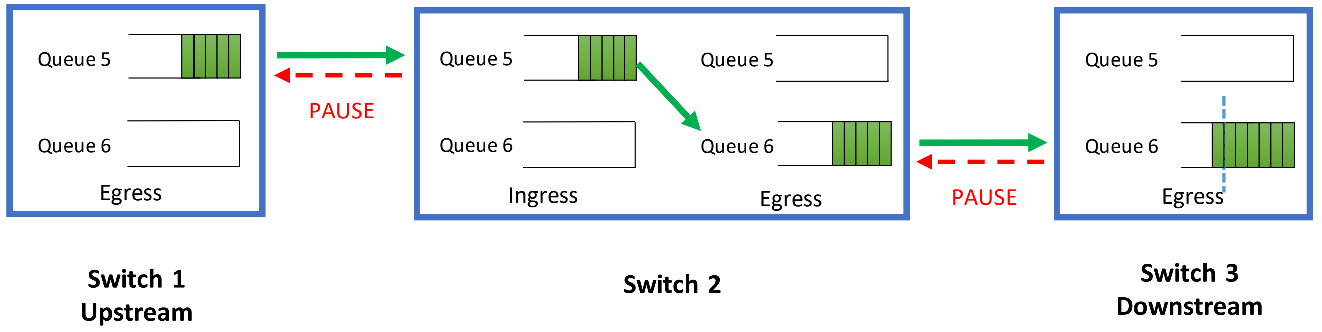
- **<u>Proactive</u>:** deadlock prevention is a more principled approach to this problem.

Deadlock free mechanism (Proactive)

- Identify CBD break point and prevent the PFC Deadlock
- Mindset:
  - Although the traffic in CLOS network itself has no loop, when the link fails or jitters, it may cause the reroute which may form CBD.
  - Use some attributes of the switch or server (Device type, Device Level, Port type) to design a method to judge if packet reroute happens.
  - Use some certain mechanisms to prevent CBD, then PFC deadlock. For example switch the priority queue.

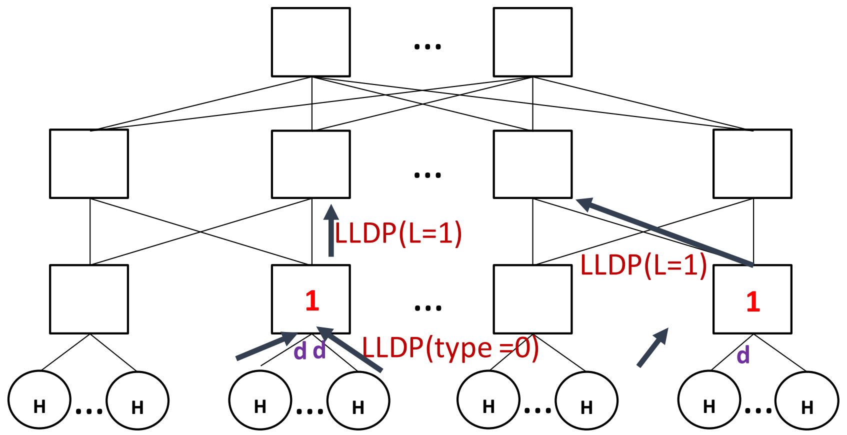


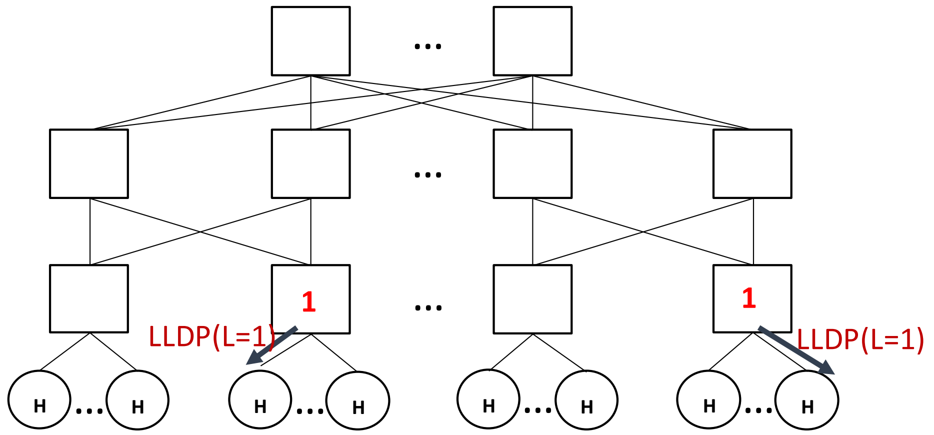
Recognize down-up change, identify the CBD break point



- ➢ Both Queue5&6 are lossless queues(Enable PFC)
- ➢ Switch 2 judge the packet and enqueue to Queue6, modify the DSCP
- ➢ When downstream triggered the PFC on Queue 6 in switch 3, PFC will map to Queue 5 in switch 2.

LLDP carry necessary information -1





Step 1:

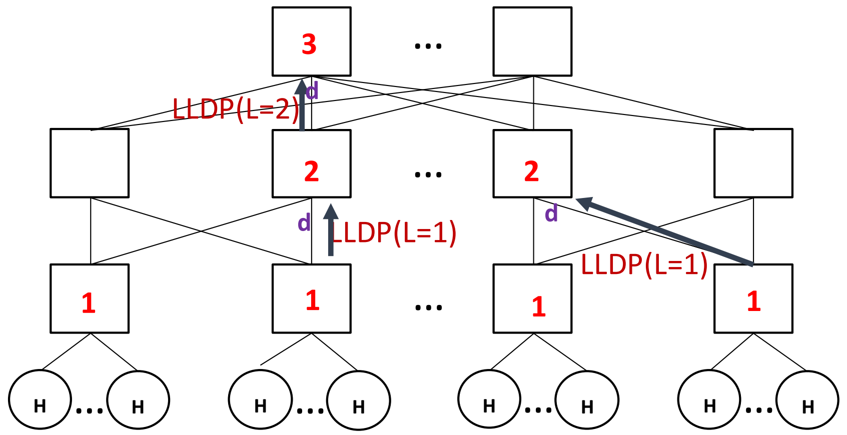
- When the switch receives a LLDP packet, if the **Device Type = 0** (host) in the packet, the switch knows that it is the switch closest to the server. If the switch does not have **Device Level** information or has level ≠1, set its own Device Level to 1, the corresponding **port is set to downlink**, and the other ports are unmarked.
- If you already have level=1, just set the corresponding **Port Type to downlink**.

Step 2:

- When the Server (Device Type = 0) itself receives an LLDP packet containing the level information , the level information is ignored.

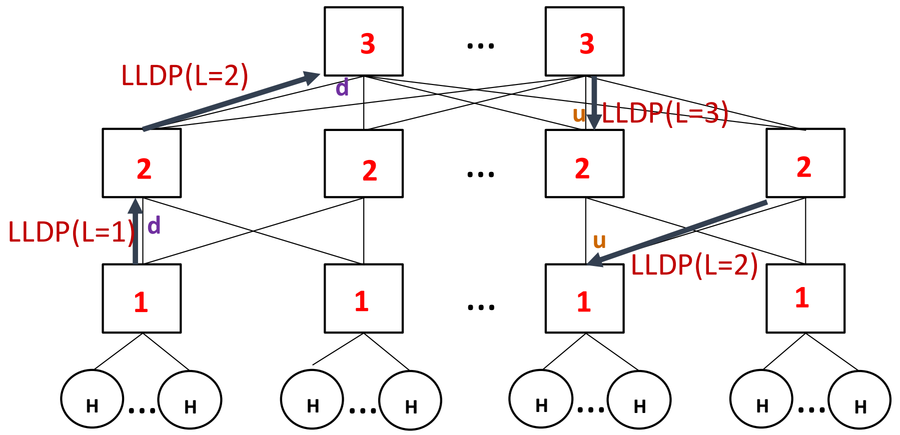
▹ The LLDP function are enabled both on the servers and switchs

LLDP carry necessary information -2



Step 3:

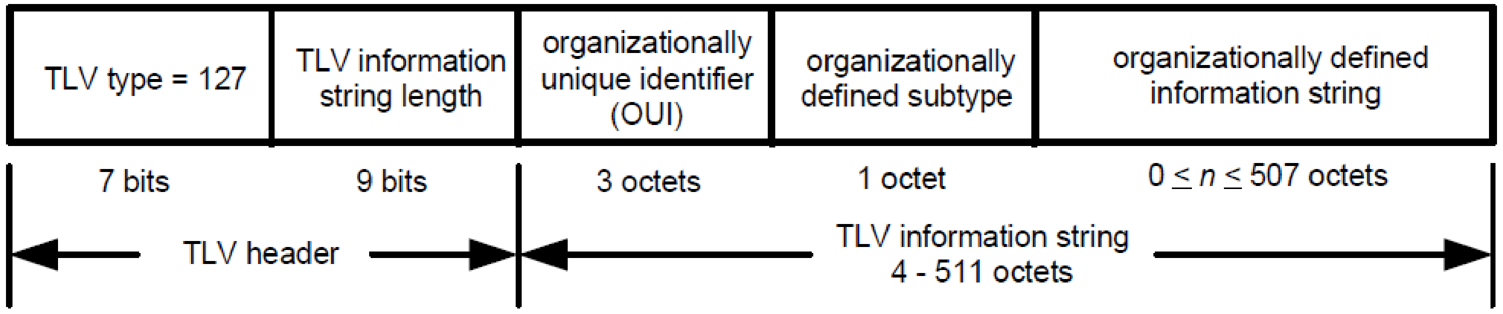
- When the switch (Device Type = 1) received a LLDP packet containing the **Device Level=Lx** information,
- At this time, If the switch does not have Device Level information and the minimum of the level from LLDP messages received from all the ports is Lm. Set the switch level to Lm+1.
- The corresponding port is set to downlink and the other ports are set to unmarked.



Step 4：

- When the switch (Device Type = 1) has its own Device level = N value and received a LLDP packet containing the Device Level = M information.
- If N = M + 1, the corresponding Port Type is downlink. If N = M - 1, the corresponding Device port is uplink.

Support Level/Port type/Device type in Organizationally Specific TLVs



TLV type = 127
TLV information string length
organizationally unique identifier (OUI)
organizationally defined subtype
organizationally defined information string
7 bits
9 bits
3 octets
1 octet
0 ≤ n ≤ 507 octets
TLV header
TLV information string
4 - 511 octets

Figure 9-12—Basic format for Organizationally Specific TLVs

LLDP TLV:

TLV Type(7 bits)	TLV Length (9 bits)	OUI (3 octets)	Subtype (1 octet)	Device Type (1 octet)	Device Level (1 octet)	Port uplink/downlink(1 octet)
127		LLDP OUI	0x1	0~0xFF 0： host 1： switch …	0~0xFF 0： Server 1： Level 1 2: Level 2 …	0~0xFF 0： unmark 1： uplink 2: downlink …reserved

Next step

- Propose to define the new Organizationally Specific TLVs in the project.
- Support **Device type/Device level/Port type** in Organizationally Specific TLVs.

Thank you