



FPMAs for VN_Port to VN_Port Virtual Links

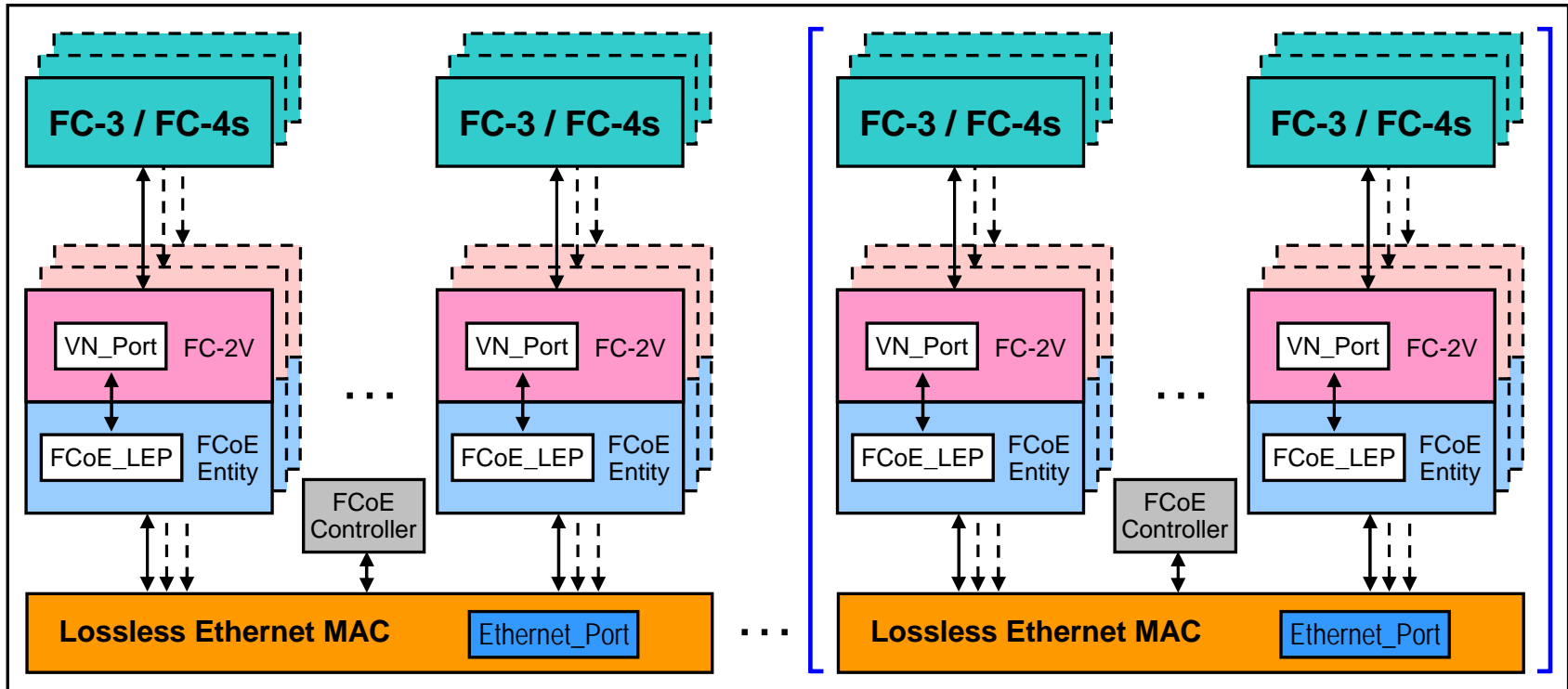
Claudio DeSanti

T11/09-663v0, December 2009

Image Credit:
Landon Curt Noll



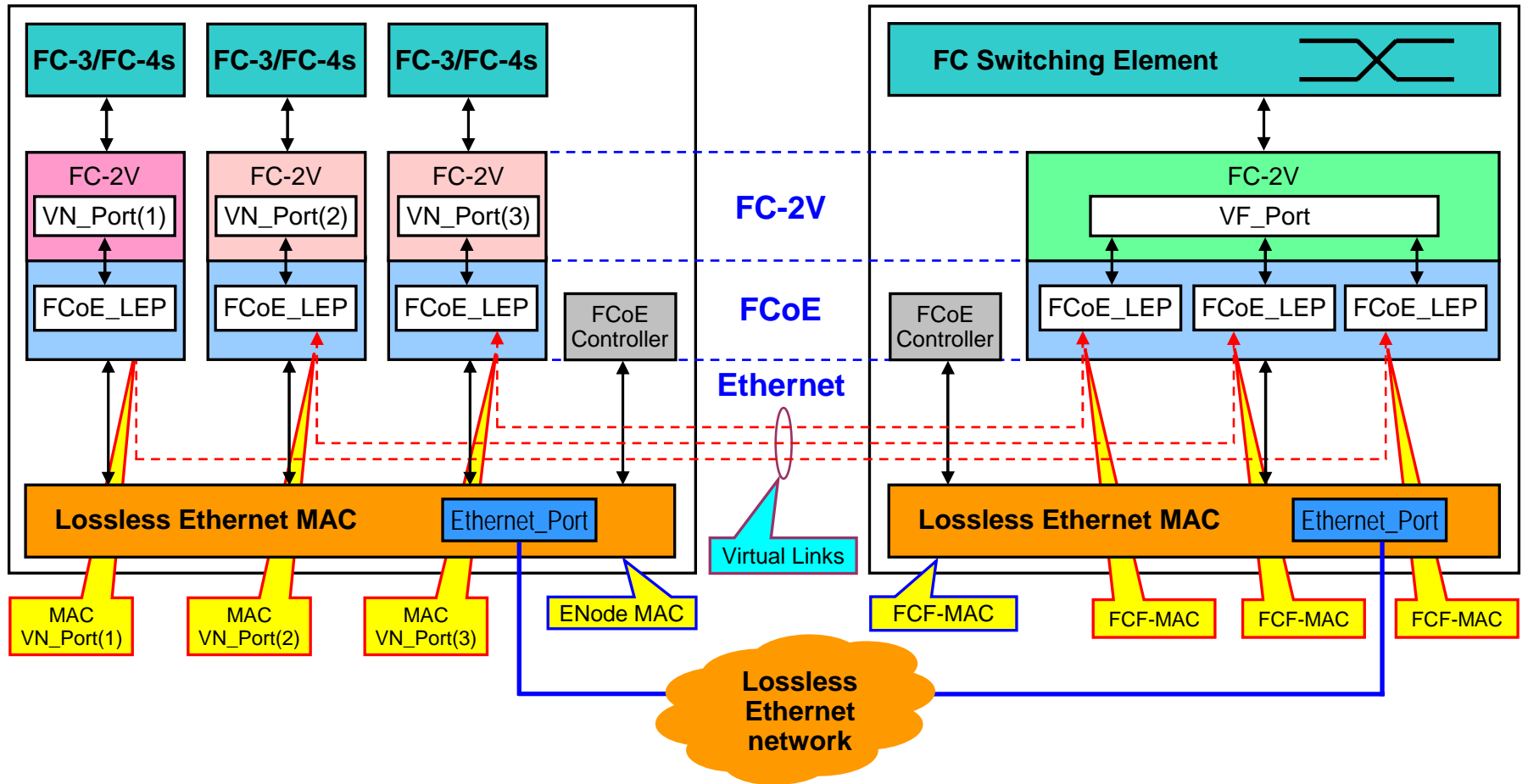
FCoE Node (ENode)



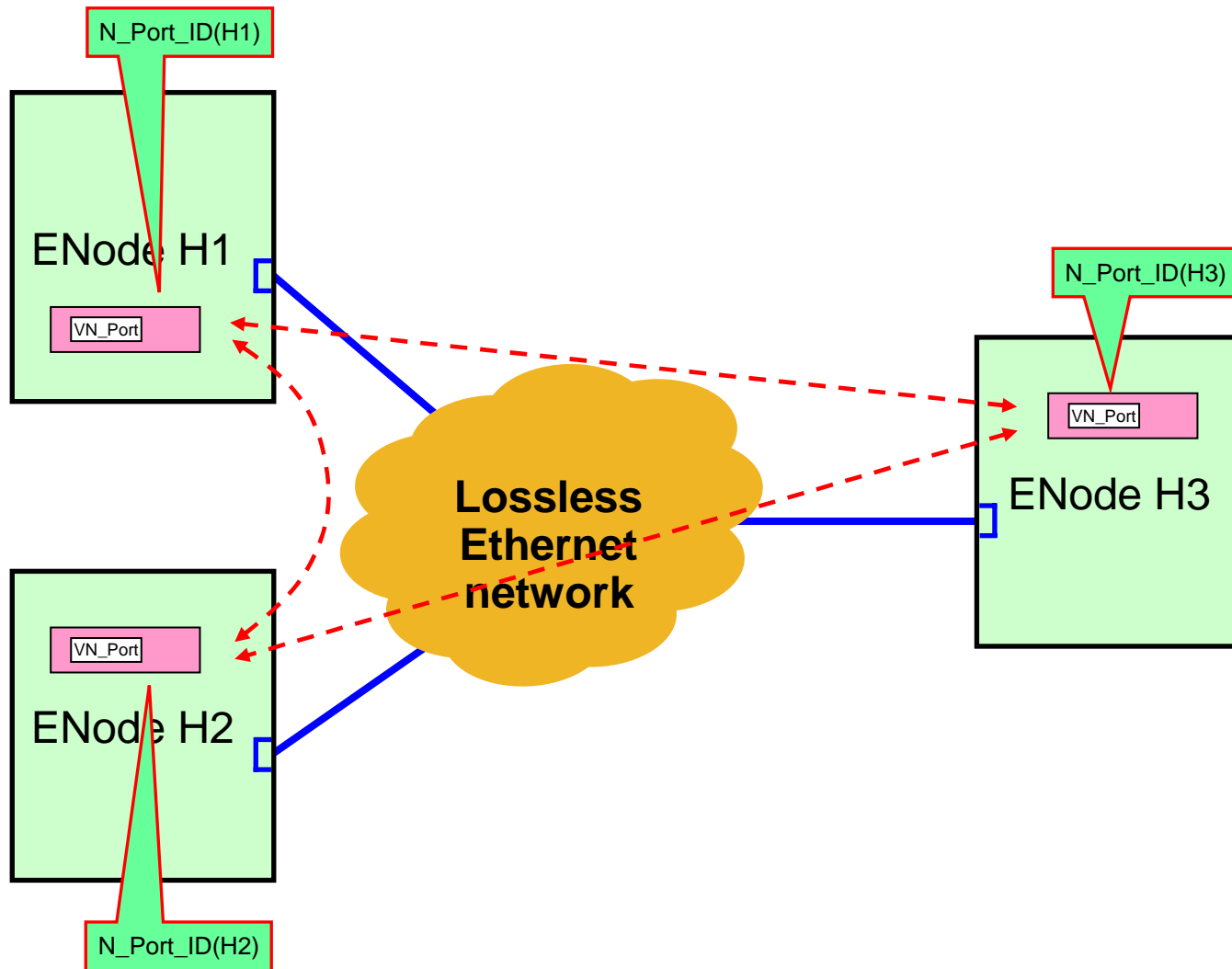
VN_Port to VF_Port Virtual Links

ENode

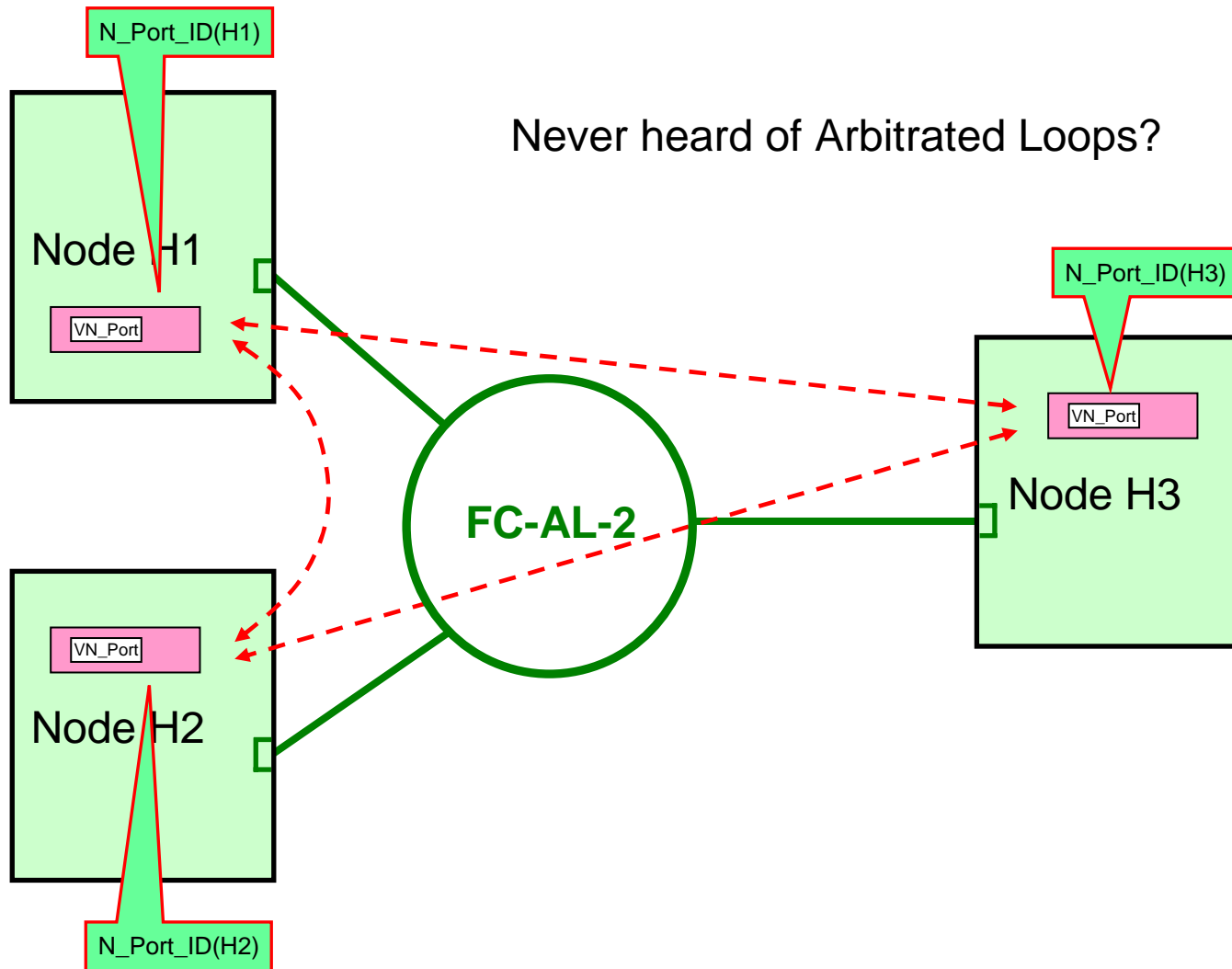
FCF



VN_Port to VN_Port Configuration (1)



Something New?

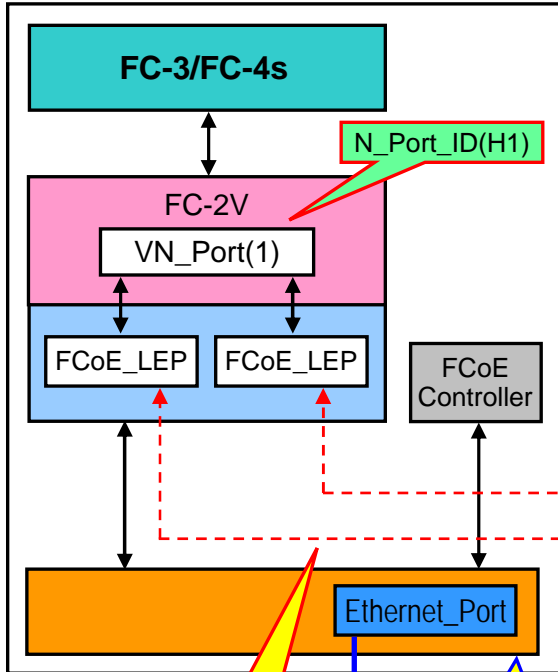


N_Port_ID Assignment (1)

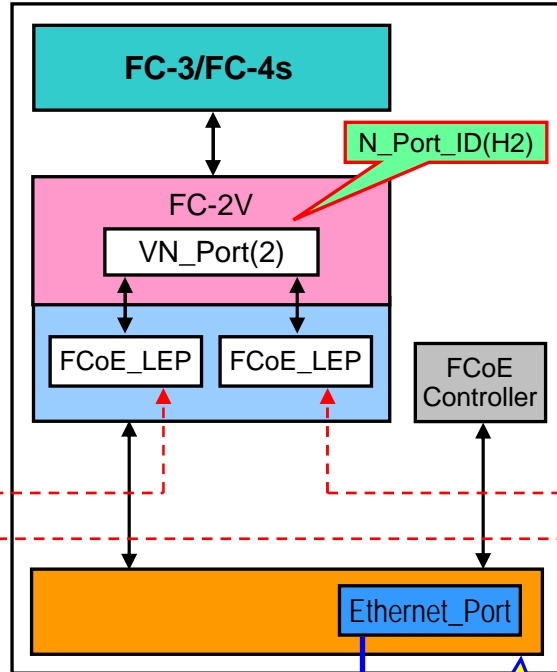
- **ENodes supporting VN_Port to VN_Port Virtual Links (VN2VN ENodes) have to perform N_Port_ID assignment by themselves**
 - There is no Fabric that can perform this function for them
- **A VN2VN ENode has to determine an N_Port_ID unique over the network to which it is connected**
 - A locally unique N_Port_ID
- **A locally unique N_Port_ID determines the FPMA to be used as VN_Port MAC Address**
 - FC-MAP || N_Port_ID

VN_Port to VN_Port Virtual Links (1)

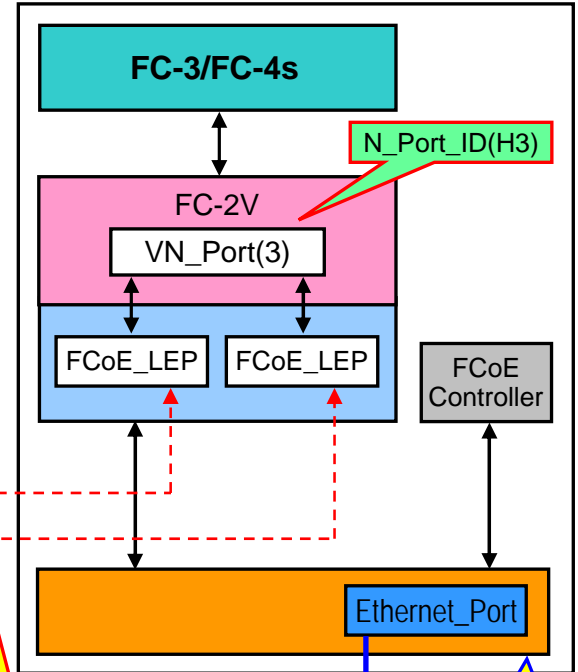
ENode H1



ENode H2



ENode H3



VL End-Points:
MAC VN_Port(1)
MAC VN_Port(3)

ENode MAC
H1

VL End-Points:
MAC VN_Port(1)
MAC VN_Port(2)

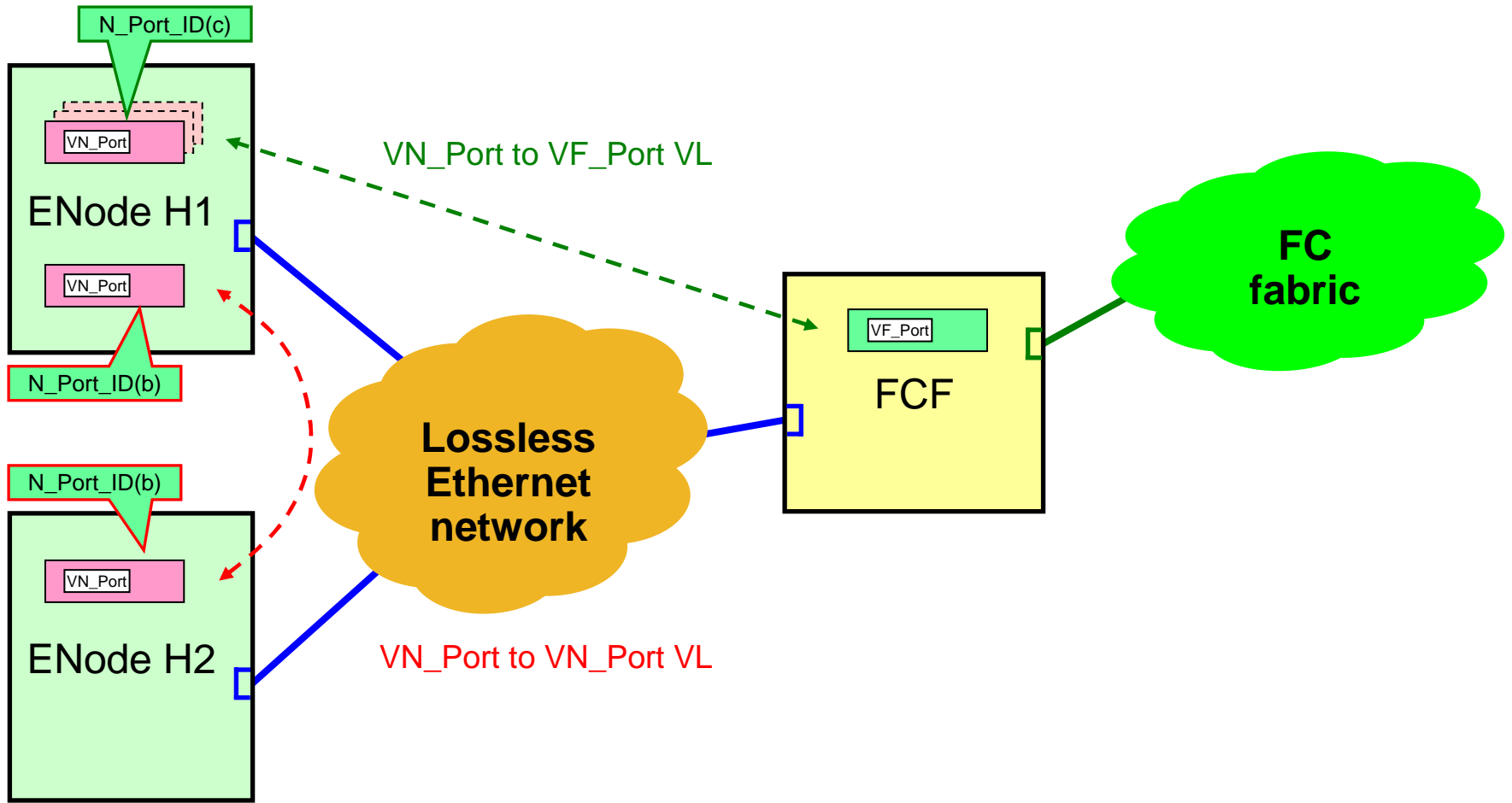
ENode MAC
H2

VL End-Points:
MAC VN_Port(2)
MAC VN_Port(3)

ENode MAC
H3



VN_Port to VN_Port Configuration (2)



N_Port_ID Assignment (2)

- **N_Port_IDs used for VN_Port to VN_Port Virtual Links shall not interfere with N_Port_IDs assigned by a Fabric**

This ensure each VN_Port has a unique N_Port_ID

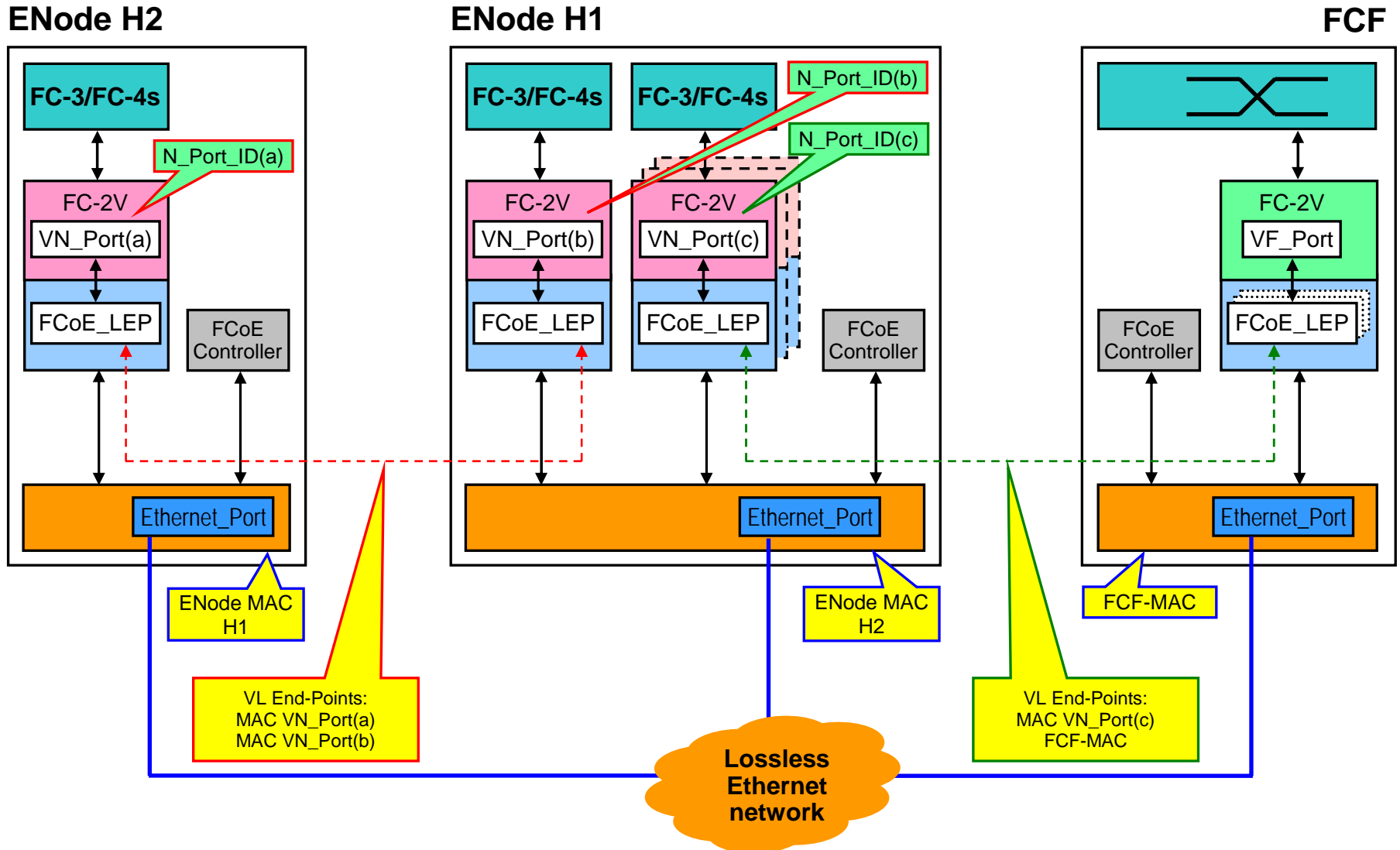
- **This is easy to achieve:**

A Fabric does not assign N_Port_IDs with Domain_ID in the range F0h .. FEh (see FC-SW-5)

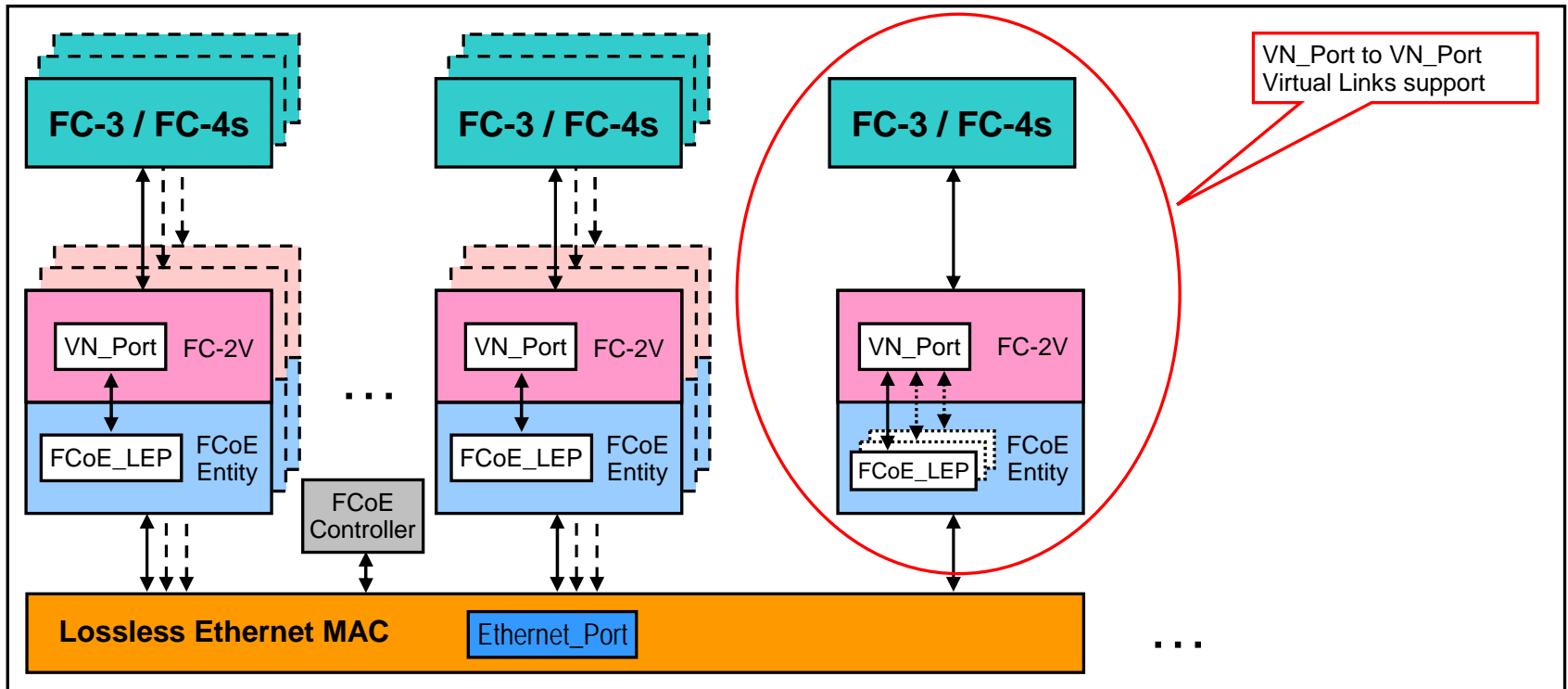
An N_Port_ID with a Domain_ID in that range is valid per FC-FS-x and therefore suitable to be used for VN_Port to VN_Port configurations

Let's use locally unique N_Port_ID of the form F0.xx.yy

VN_Port to VN_Port Virtual Links (2)



ENode with VN_Port to VN_Port Support



Why FPMAs?

- **FPMAs allow to identify all FCoE addresses in a practical manner**
 - i.e., upper 24 bits = FC-MAP
 - This allows easy construction of FCoE ACLs (see Annex C of FC-BB-5)
- **FPMAs are computed and used by FCoE only**
 - A requirement from the FCoE threat model analysis (see section 7.6 of FC-BB-5)
- **FPMAs are easily movable with Virtualization**
 - Because are not tied to the hardware
- **FPMA definition:**
 - A MAC address of the form FC-MAP || N_Port_ID
 - One FPMA per VN_Port



Image Credit: Wikipedia
Creative Commons License

Additional Considerations

- It is nice to have the VN_Port to VN_Port operations independent from the VN_Port to VF_Port operations
- When a Fabric is present the FC-MAP is provided by the Fabric
In the range 0EFC00h .. 0EFCFFh
- Let's use a different FC-MAP for VN_Port to VN_Port operations
e.g., 0EFD00h
Easy ACLs to control VN_Port to VN_Port operations
- Let's define a new multicast address to identify ENodes supporting VN_Port to VN_Port operations
e.g., All-VN2VN-ENode-MACs

Locally Unique N_Port_IDs



Image Credit:
Flickr user TheMM
Creative Commons
License

- **How to compute locally unique N_Port_IDs?**
- **Let's re-use a well proven standard technique:
RFC 3927**

'Dynamic Configuration of IPv4 Link-Local Addresses'

**A protocol to compute locally unique IPv4 addresses
of the form 169.254.xx.yy**

16 bits of address space

As for F0.xx.yy N_Port_IDs

Implemented and used on every PC since 10 years

Supported by Windows 98 and Mac OS 8.5

- **Adapt the protocol and algorithm to FCoE**

Locally Unique N_Port_ID Selection Algorithm

- 1. Select a tentative locally unique N_Port_ID and enable reception of frames sent to All-VN2VN-ENode-MACs**
- 2. Wait a random time between 0 and 100 ms**
- 3. Send two FIP Address Check Requests, with C=0, to All-VN2VN-ENode-MACs, spaced 100ms**
- 4. Wait up to 400 ms to collect FIP Address Check Responses indicating N_Port_ID conflict; if a conflict is detected, select another tentative locally unique N_Port_ID and go back to step 3**
- 5. Send a FIP Address Check Request, with C=1, to All-VN2VN-ENode-MACs**
- 6. Wait up to 400 ms to collect padded FIP Address Check Responses; if a conflict is detected, select another tentative locally unique N_Port_ID and go back to step 3**
- 7. Send a FIP VN2VN Advertisement to All-VN2VN-ENode-MACs**



Image Credit:
Flickr user mr.beaver
Creative Commons
License

Locally Unique N_Port_ID Selection Properties



Public domain photo by Mark Prince

- **A tentative locally unique N_Port_ID is selected through a pseudo-random number generator with a uniform distribution in the range 0000h .. FFFFh**
- **The pseudo-random number generator should be seeded with the N_Port_Name of the VN_Port for which a locally unique N_Port_ID is being selected**
 - So different ENodes generate different numbers
 - So a VN_Port will usually be associated with the same N_Port_ID each time it is enabled
- **An ENode connecting to a network with already 1300 ENodes has a 98% chance of selecting an unused locally unique N_Port_ID on the first try and a 99.96% chance of selecting an unused locally unique N_Port_ID within two tries**
 - The probability that it will have to try more than ten times is about 10^{-17}
 - The algorithm cycles in 1 second

FIP Operation Codes and SubCodes

Operation Code	SubCode	Operation
0001h	01h	Discovery Solicitation
	02h	Discovery Advertisement
0002h	01h	Virtual Link Instantiation Request
	02h	Virtual Link Instantiation Reply
0003h	01h	FIP Keep Alive
	02h	FIP Clear Virtual Link
0004h	01h	FIP VLAN Request
	02h	FIP VLAN Notification
0005h	01h	Address Check Request
	02h	Address Check Response
	03h	VN2VN Advertisement
FFF8h .. FFFEh	00h .. FFh	Vendor Specific
All others	Others	Reserved

New operation codes and subcodes to not interfere with existing 'code'

Address Check Request

Response required if C=1
 Response only when a conflict is detected if C=0

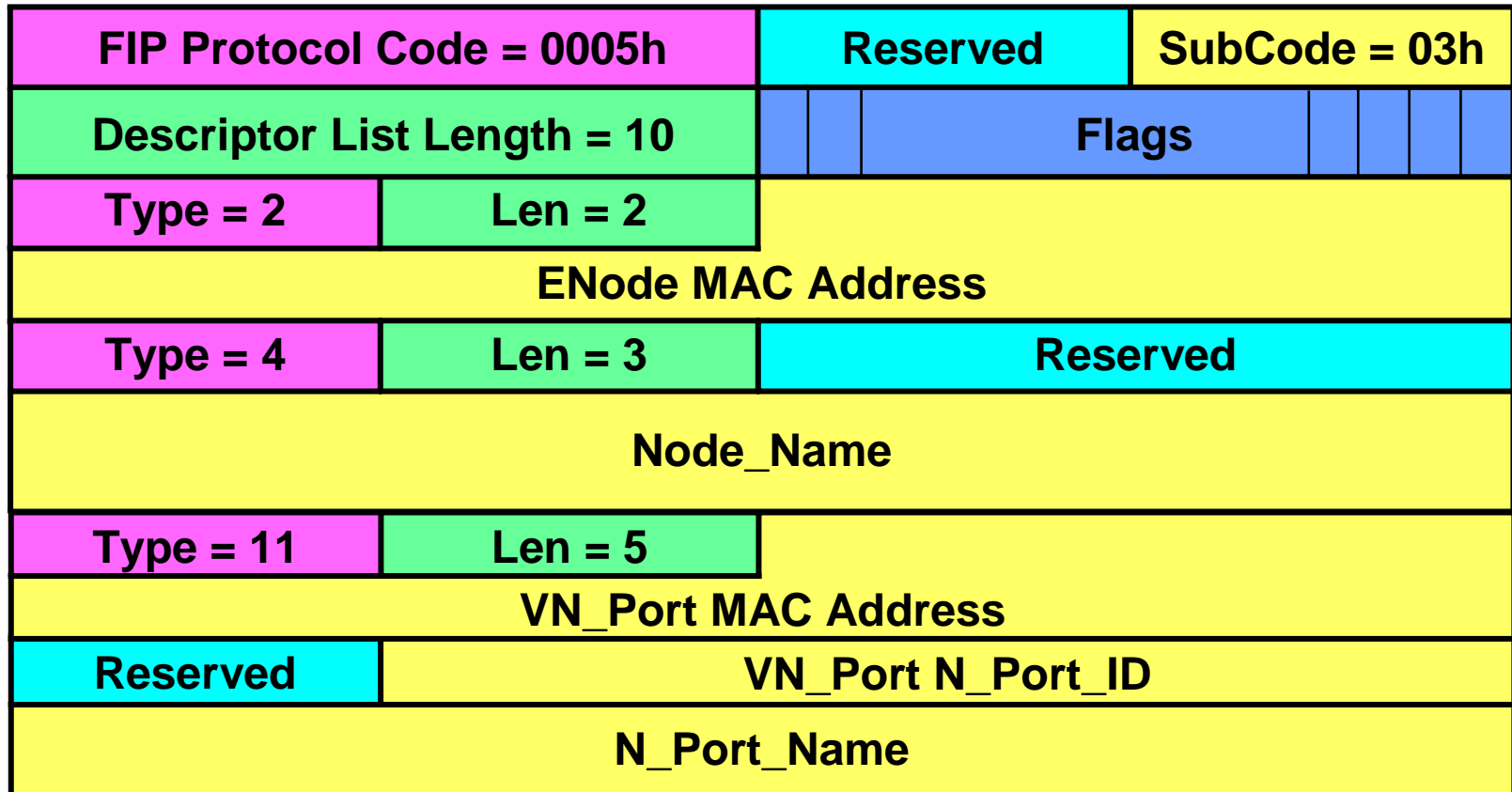
FIP Protocol Code = 0005h		Reserved	SubCode = 01h
Descriptor List Length = 11		Flags	
Type = 2	Len = 2	C	
ENode MAC Address			
Type = 4	Len = 3	Reserved	
Node_Name			
Type = 11	Len = 5		
Tentative MAC Address			
Reserved	Tentative N_Port_ID		
N_Port_Name			
Type = 6	Len = 1	Max FCoE Size	

Address Check Response

C=1 if conflict is detected
C=0 if conflict is not detected

FIP Protocol Code = 0005h		Reserved	SubCode = 02h	
Descriptor List Length = 10		Flags		C
Type = 2	Len = 2	ENode MAC Address		
Type = 4	Len = 3	Reserved		
Node_Name				
Type = 11	Len = 5	VN_Port MAC Address		
Reserved	VN_Port N_Port_ID			
N_Port_Name				
FIP Pad to Max FCoE Size if C=1 in the Address Check Request, otherwise no FIP Pad				

VN2VN Advertisement



- SA = VN_Port MAC Address
- DA = All-VN2VN-ENode-MACs



Additional Steps

- **A VN2VN ENode collects the FIP Address Check Responses received in step 6 in a VN2VN ENode list**
- **A VN2VN ENode can use the information in the VN2VN ENode list as a “poor man name server”**
 - i.e., to associate a Port_Name to an N_Port_ID**
- **A VN2VN ENode performs FIP FLOGIs to selected remote VN2VN ENodes**
 - Exchanging the associated FPMAs**
- **Then an FCoE encapsulated PLOGI is performed**
 - Announcing to the Fibre Channel stack the N_Port_ID choice made through FIP**
- **A VN2VN ENode sends periodically a VN2VN Advertisement to All-VN2VN-ENode-MACs to verify the it is operational**
 - Keep-alive function**

Summary

- **Direct connectivity needs N_Port_IDs**
- **Let's allocate them locally unique from a reserved "Domain_ID"**
 - Using a well proven standard technique
 - Implemented since 10 years by virtually all IP hosts
- **The associated FPMA can be derived from the N_Port_ID**
 - Using a dedicated FC-MAP
- **This allows to handle locally unique N_Port_IDs with a single ACL**
 - Greatly reducing ACL scaling and management issues
- **The approach solves both point-to-point and small network configurations**
 - Compatible with independent/concurrent Fabric operations



Public domain photo by Mark Prince

Thank You



Image Credit
Flickr user Kaeru
Creative Commons License