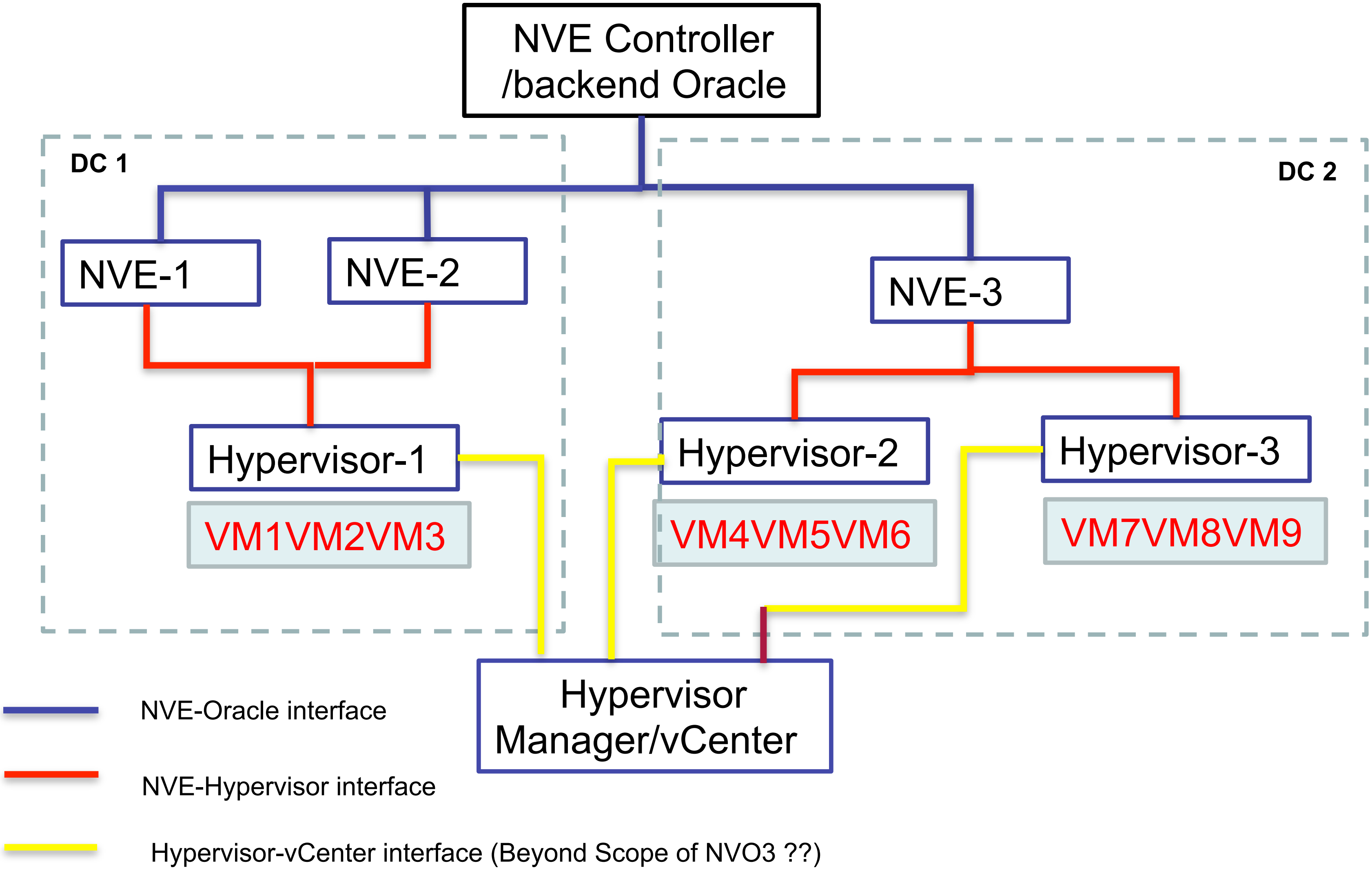
Network Virtualization Architecture Design and Control Plane Requirements

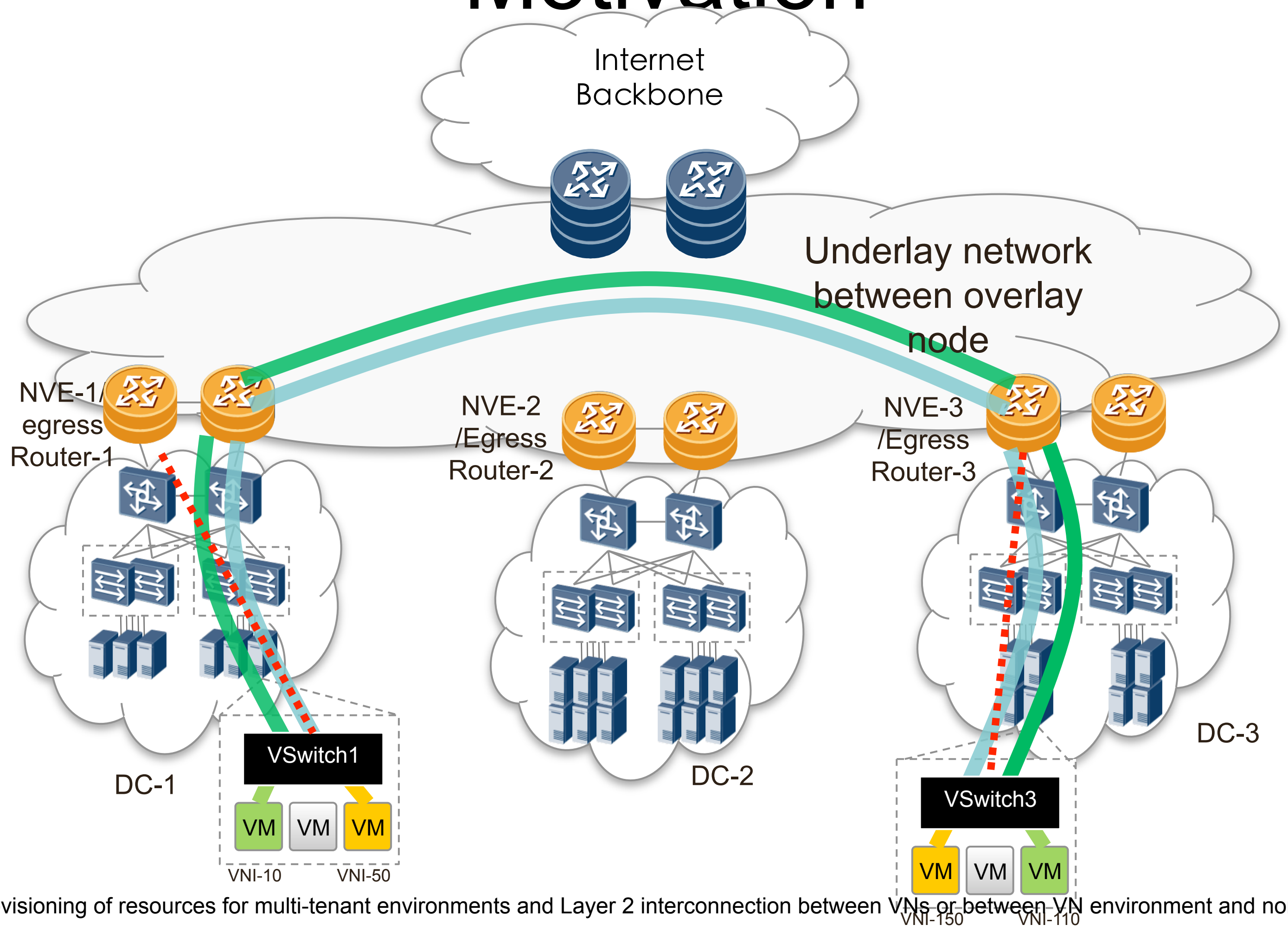
draft-fw-nvo3-server2vcenter-01
draft-wu-nvo3-nve2nve
draft-wu-nvo3-mac-learning-arp

Qin Wu
Roland Scott

NVO3 architecture Overview

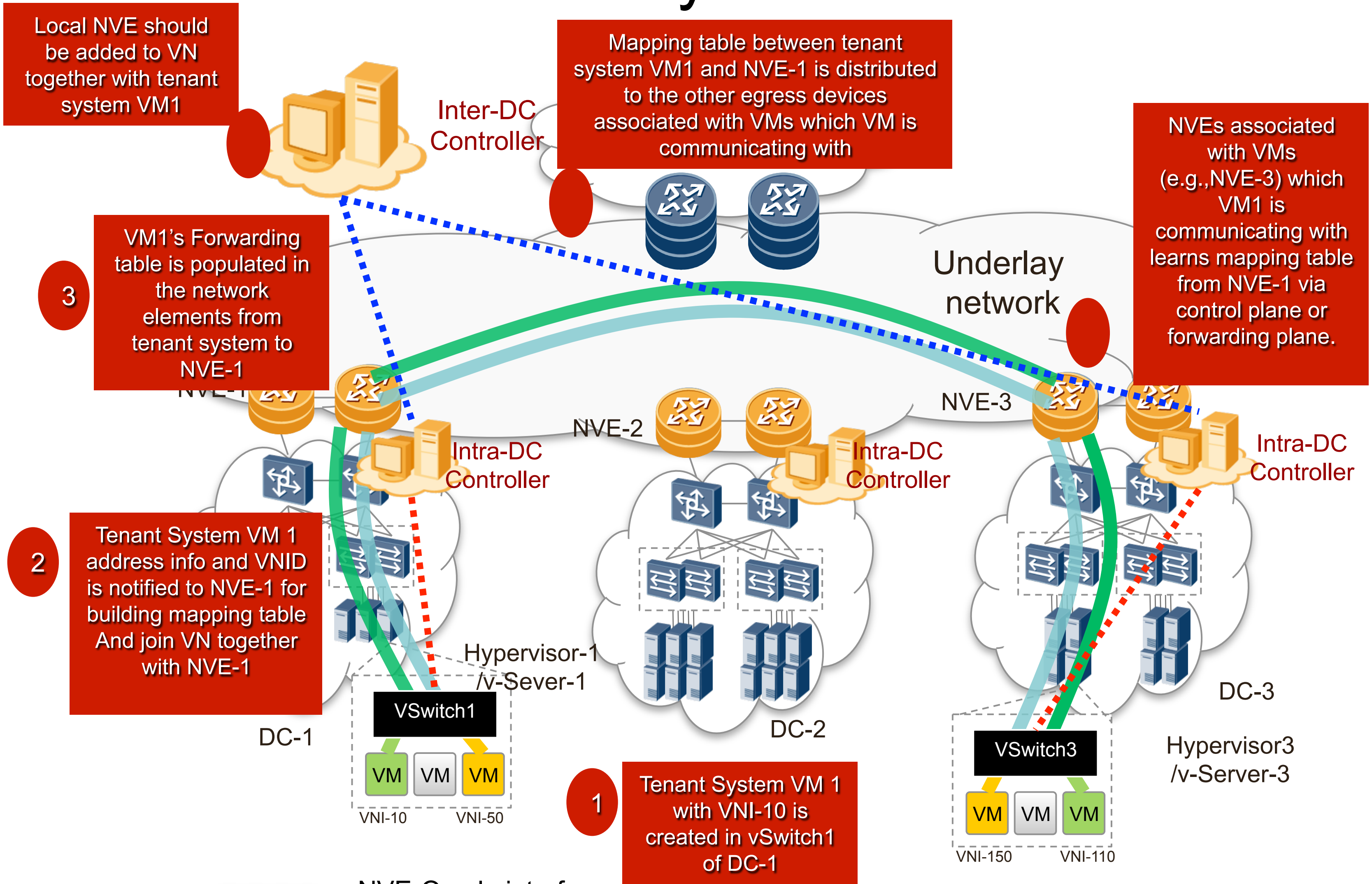


Motivation



- Provisioning of resources for multi-tenant environments and Layer 2 interconnection between VNs or between VN environment and non VN environment are two very important features for cloud computing
- Two challenging issues are
  - how to provision network connectivity in end to end mode, particular for a moving tenant
  - To enable two VM communication, overlay nodes should know which tunnel the packet needs to be sent to. The VM should know MAC address of VM which it communicate with.
- This slides go into details to discuss centralized approach and distributed approach for Auto provision and network connectivity setup.

Network Connectivity Auto Provision Overview



Local NVE should be added to VN together with tenant system VM1

Mapping table between tenant system VM1 and NVE-1 is distributed to the other egress devices associated with VMs which VM is communicating with

NVEs associated with VMs (e.g.,NVE-3) which VM1 is communicating with learns mapping table from NVE-1 via control plane or forwarding plane.

3 VM1's Forwarding table is populated in the network elements from tenant system to NVE-1

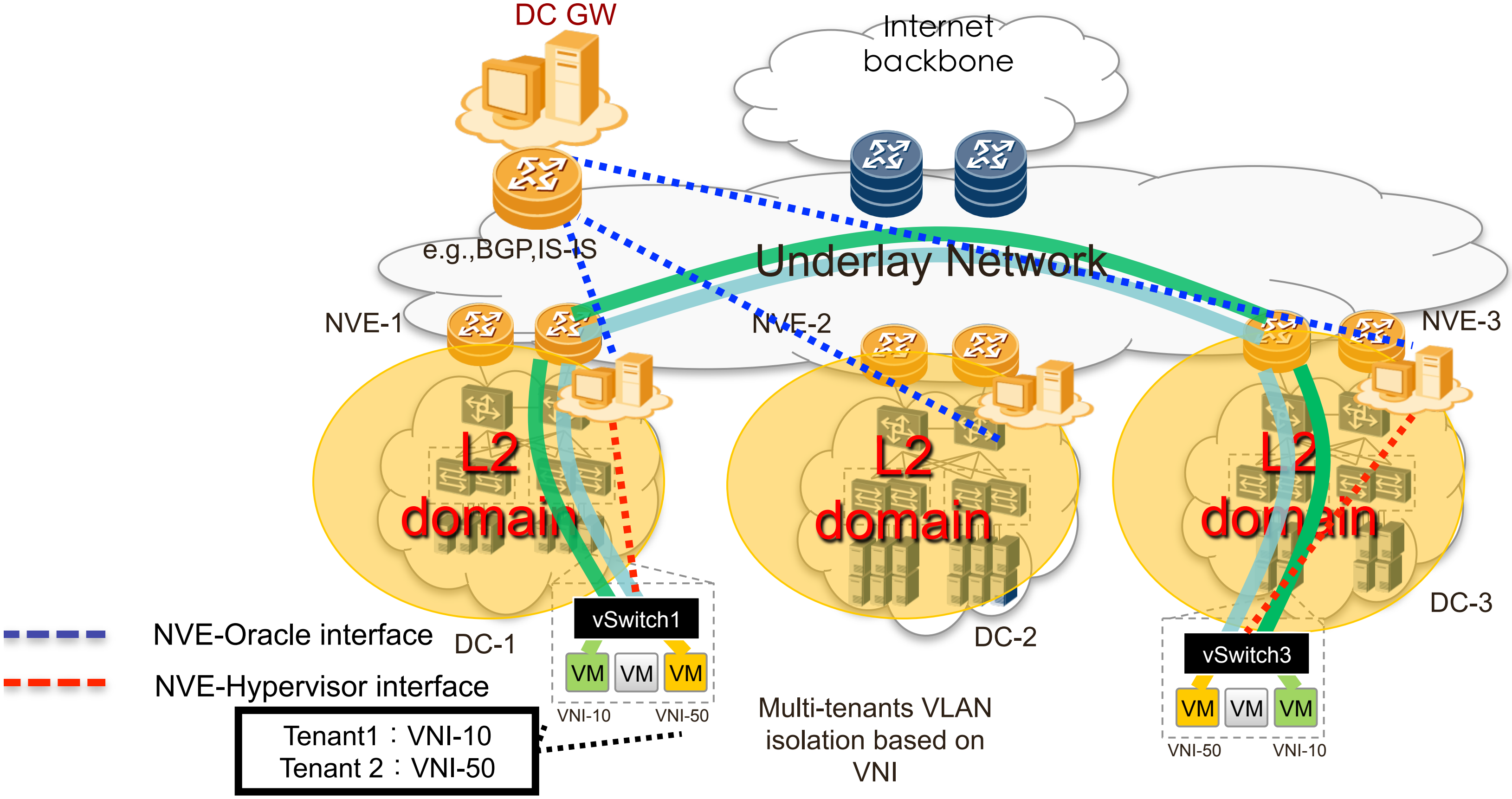
2 Tenant System VM 1 address info and VNID is notified to NVE-1 for building mapping table And join VN together with NVE-1

1 Tenant System VM 1 with VNI-10 is created in vSwitch1 of DC-1

- - - - NVE-Oracle interface

- - - - NVE-Hypervisor interface

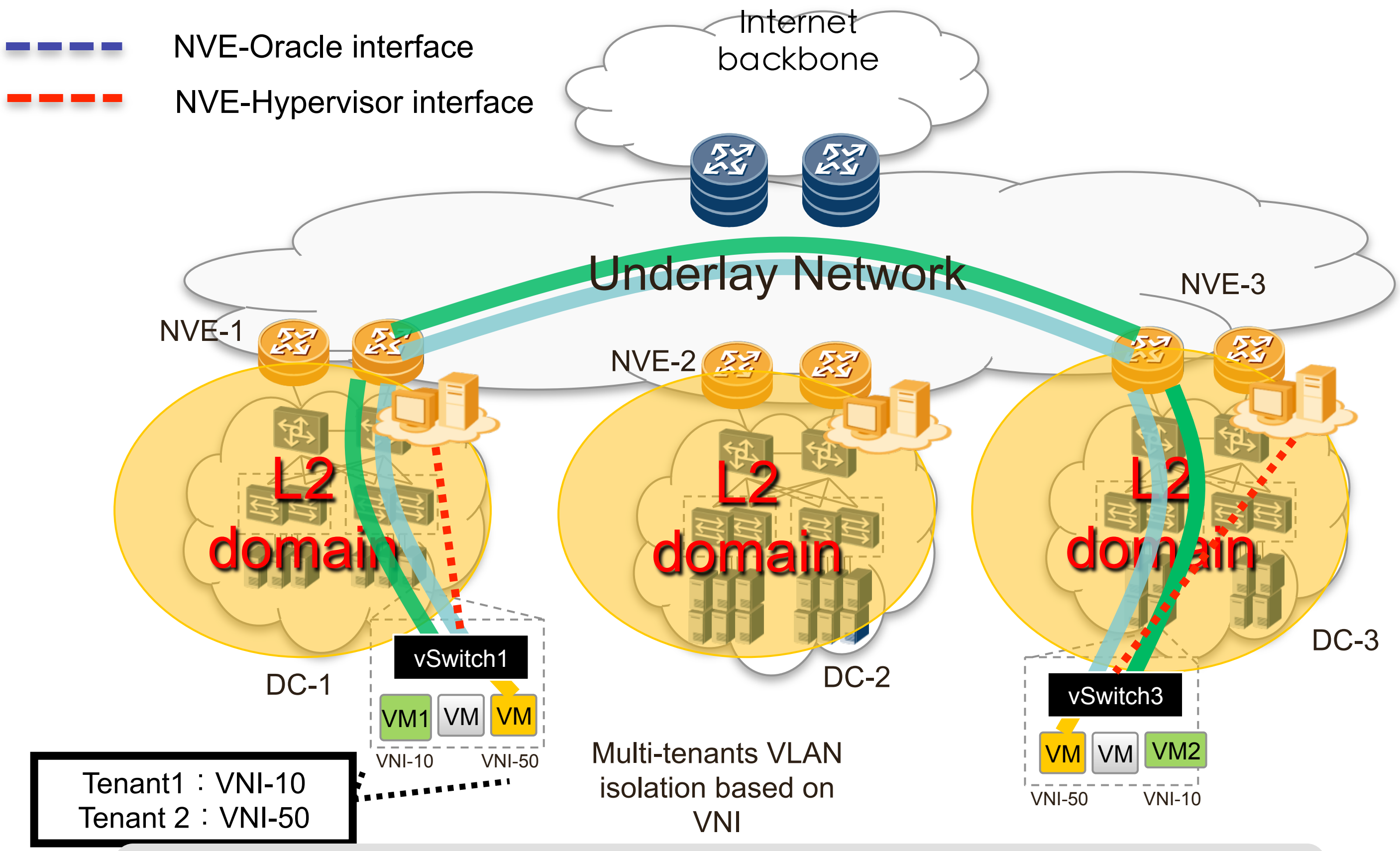
Mapping table creation/distribution/update



Multi-tenants VLAN isolation based on VNI

- When one tenant system is attached to local NVE, tenant system(i.e.,VM) should be assigned with MAC address, IP address and Virtualization Network Identifier (supporting multi-tenant environment)
- Tenant system should tell local NVE it attached about its own MAC address and VNID.
- The local NVE as overlay node establish mapping table and associate VM ID with overlay node ID using VNID.
- DC GW (e.g., BGP GW) should know which overlay nodes belong to the same virtualization network and which of VMs are in communication (Centralized approach).
- The local NVE should distribute such mapping table via DC GW to all the other remote NVEs that belong to the same virtualization network (Distributed approach).
- The mapping table should be updated when VM moves or connection to VN fails.
- When VM moves, VN context and VN Instance including access and tunnel policies, forwarding function should also be moved.

Destination MAC address learning :MAC address translation



Tenant1：VNI-10
Tenant 2：VNI-50

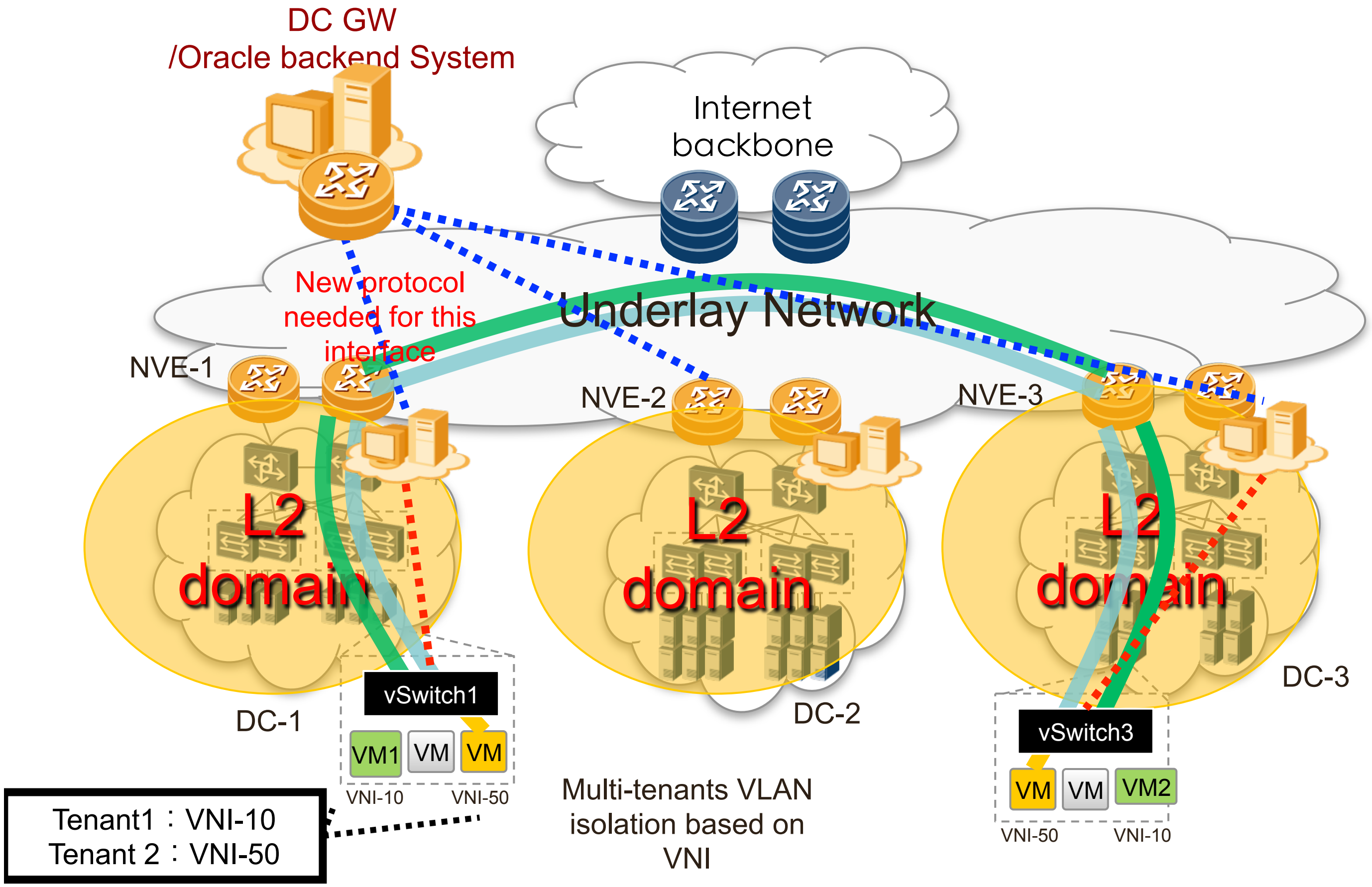
Multi-tenants VLAN isolation based on VNI

Local source NVE translate destination MAC address of ARP from source Tenant system to its own MAC address, forward it to destination tenant system and populate mapping table corresponding to destination NVE with received ARP reply.

Pro: Each local source NVE only need to learn MAC address of tenant system in its own local network and MAC address of all the destination NVEs. MAC address table size reduced greatly.

- If DC GW or source overlay node want to distribute mapping table only to the destination overlay node which belongs to the same virtualization network and is attached by destination VM who is communicating with source VM, VM learning mechanism can be used.
- ARP resolution is one typical method for VM address learning however ARP flooding should be tackled.
- In order to learn MAC address without ARP flooding, we can choose
    - a. Carry both IP address and MAC address in the control plane.
    - b. Restrict ARP message within layer 2 network behind NVE and use control protocol to distribute mapping table between NVEs

Destination MAC address learning by interaction between NVE and Oracle



🟦 NVE-Oracle interface

🟥 NVE-Hypervisor interface

① Source tenant system sends a broadcast ARP message to discover the MAC address of Destination tenant system. The message contains IP_B of Destination VM2 in the ARP message payload.

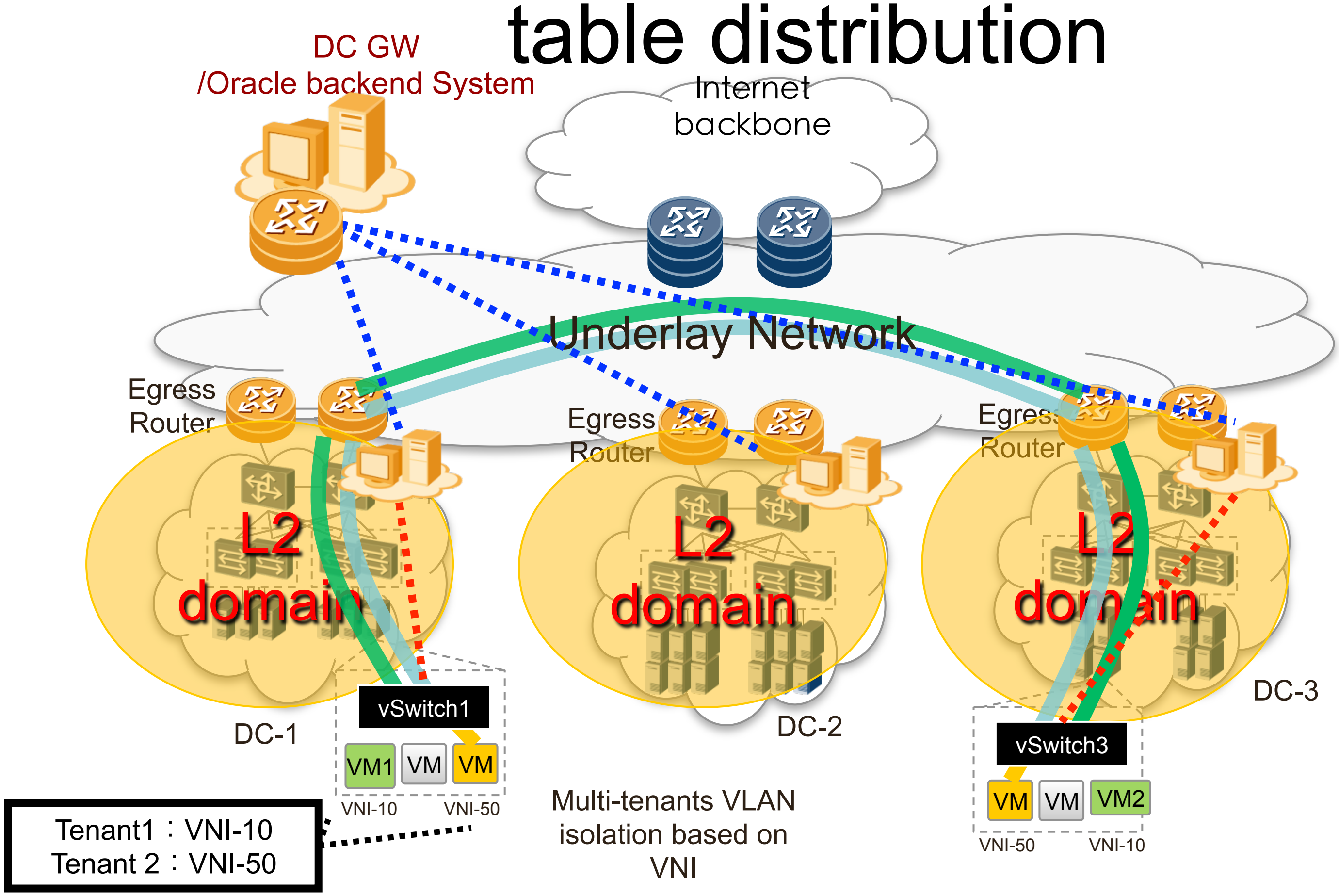
② Source NVE-1, receiving the ARP message, but rather than flooding it on the overlay network sends a Map-Request to the DC GW that maintains mapping information for entire overlay network for TEID
= <VNID,IP_B,*>.

③ The Map-Request is routed by the DC GW to Destination Overlay node, that will send a Map-Reply back to source NVE-1 containing the mapping TEID=<VNID,IP_B,MAC_B> where MAC_B is MAC address of destination VM2(Distributed approach). Alternatively, depending on the DC GW configuration, the DC GW may send directly a Map- Reply to Source NVE-1 (Centralized approach).

④ Source NVE-1populates the map-table with the received entry, and sends an ARP-Agent Reply to Source tenant system that includes MAC_B and IP_B of destination tenant system.

⑤ Source tenant system learns MAC_B from the ARP message and can now send a packet to destination tenant system by including MAC_B, and IP_B, as destination addresses.

MAC address learning by relying on mapping table distribution



Multi-tenants VLAN isolation based on VNI

a. First Mapping table established in local NVE is distributed to all NVEs in the VN (See page 5)

b. Secondly , source tenant system send an ARP to local source NVE ,if there is no mapping table corresponding to destination tenant system, local source NVE respond to tenant system with its own MAC

if there is mapping table corresponding to destination tenant system, local source NVE respond to source tenant system with MAC address of destination tenant system.

c. Thirdly, source Tenant system send a packet to destination Tenant System, the local source NVE intercept this packet and look up mapping table, if there is mapping table corresponding to destination tenant system, the local source NVE will tunnel this packet to destination NVE based on this mapping table.

- - - - NVE-Oracle interface

- - - - NVE-Hypervisor interface

Next Step

- Do WG think these work are fitted into Control plane requirements and data plane requirements?
- Do WG think some of these work can serve as the input to NVO3 architecture?
- Any other comments and suggestions?