University of Amsterdam
System and Network Engineering MSc
July 3, 2014

Software Defined VPNs

S Konstantaras & G Thessalonikefs

stavros.konstantaras@os3.nl george.thessalonikefs@os3.nl

Background

Software Defined Networking (SDN)

- A modern flexible networking concept separating the control plane from the data plane.
- A single entity governs the SDN topology and applies local policies.
- A standardized open interface (OpenFlow) allowing to combine hardware from different vendors.

Virtual Private Networks

- Logic separation of a physical infrastructure with complete traffic separation.
- Interconnects LANs which are located in different countries/continents.

A type of VPN technology is Virtual Private LAN Service (VPLS) which:

- Allows organizations to interconnect their local Ethernet networks in a scalable way .

Research Questions

The main research question is the following:

- *How can VPLS be implemented efficiently by using the OpenFlow 1.3 switch specification interface?*

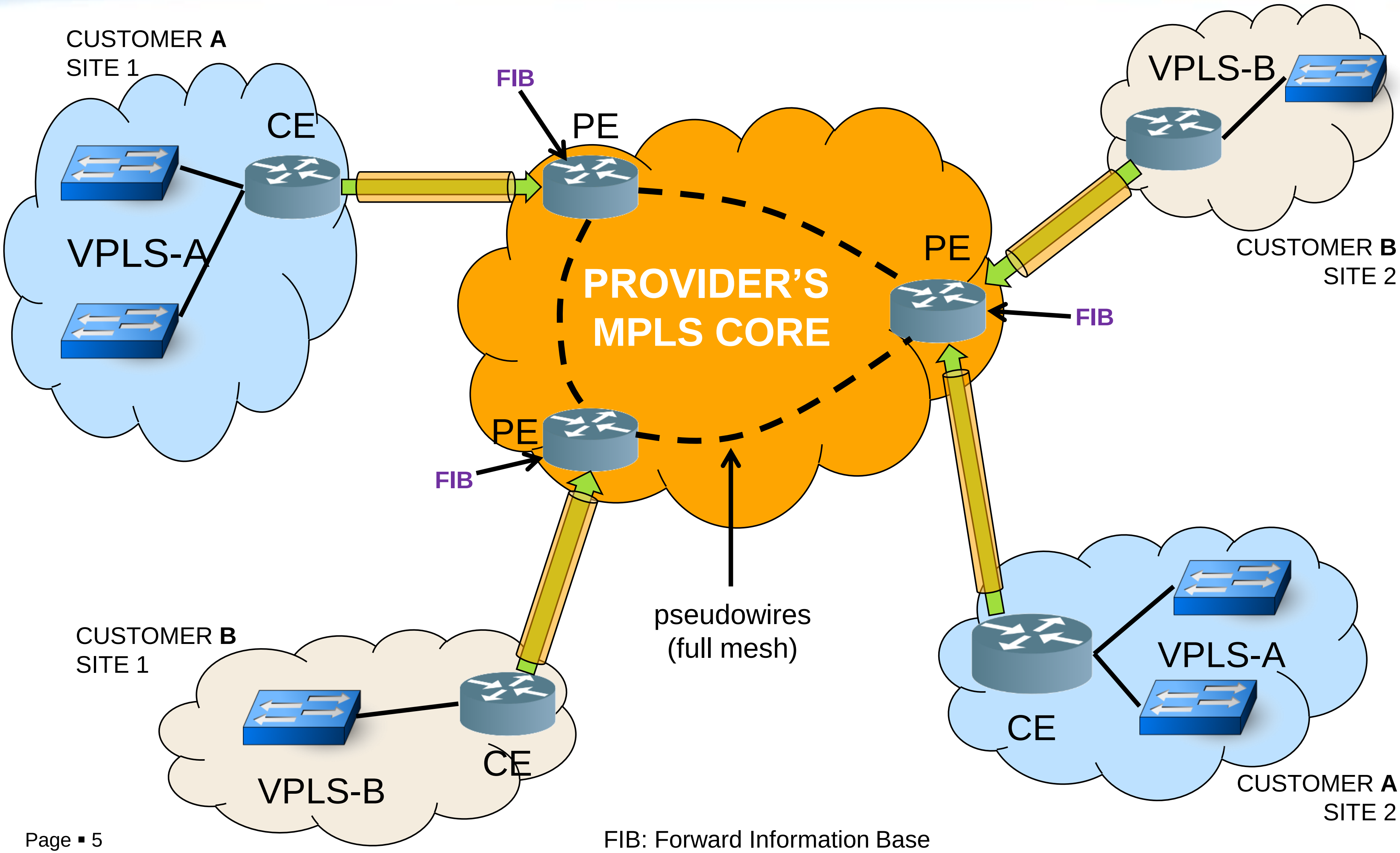
The main research question can be divided into the following sub-questions:

- *Can SDN be an underlay layer for building on-demand VPLS services?*
- *Is SDN flexible enough to support at least a scalable, efficient and effective implementation of VPLS as existing solutions?*

Outline

- Involved Technologies
- Design part
- Architecture analysis part
- Optimizations and ideas
- Conclusion

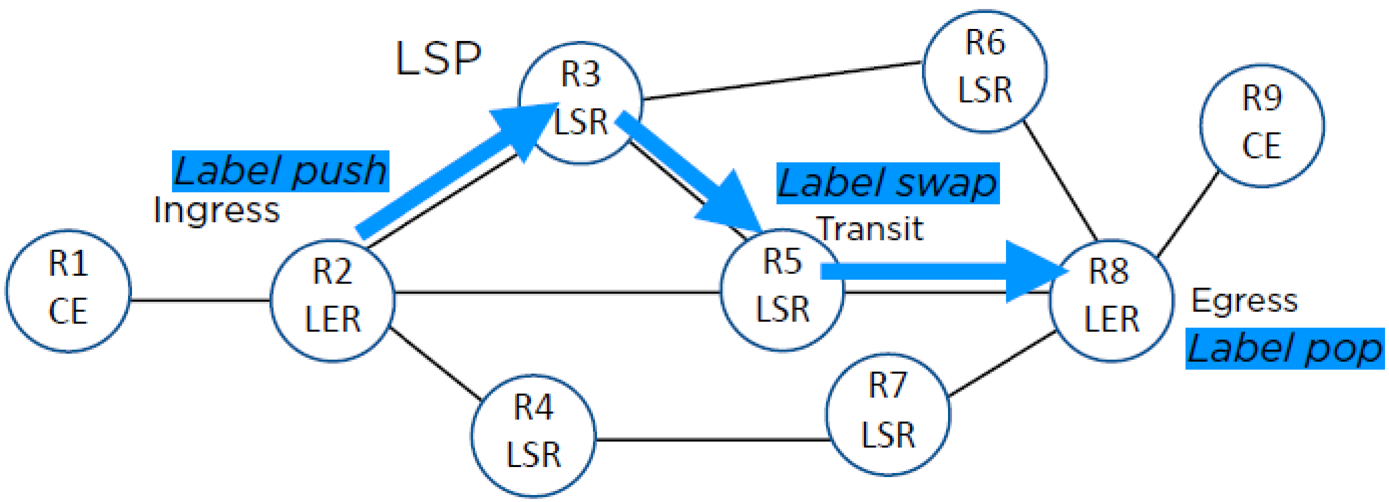
MPLS/VPLS Architecture



FIB: Forward Information Base

MPLS

- Protocol used in the core of networks
- Single domain (ISP)

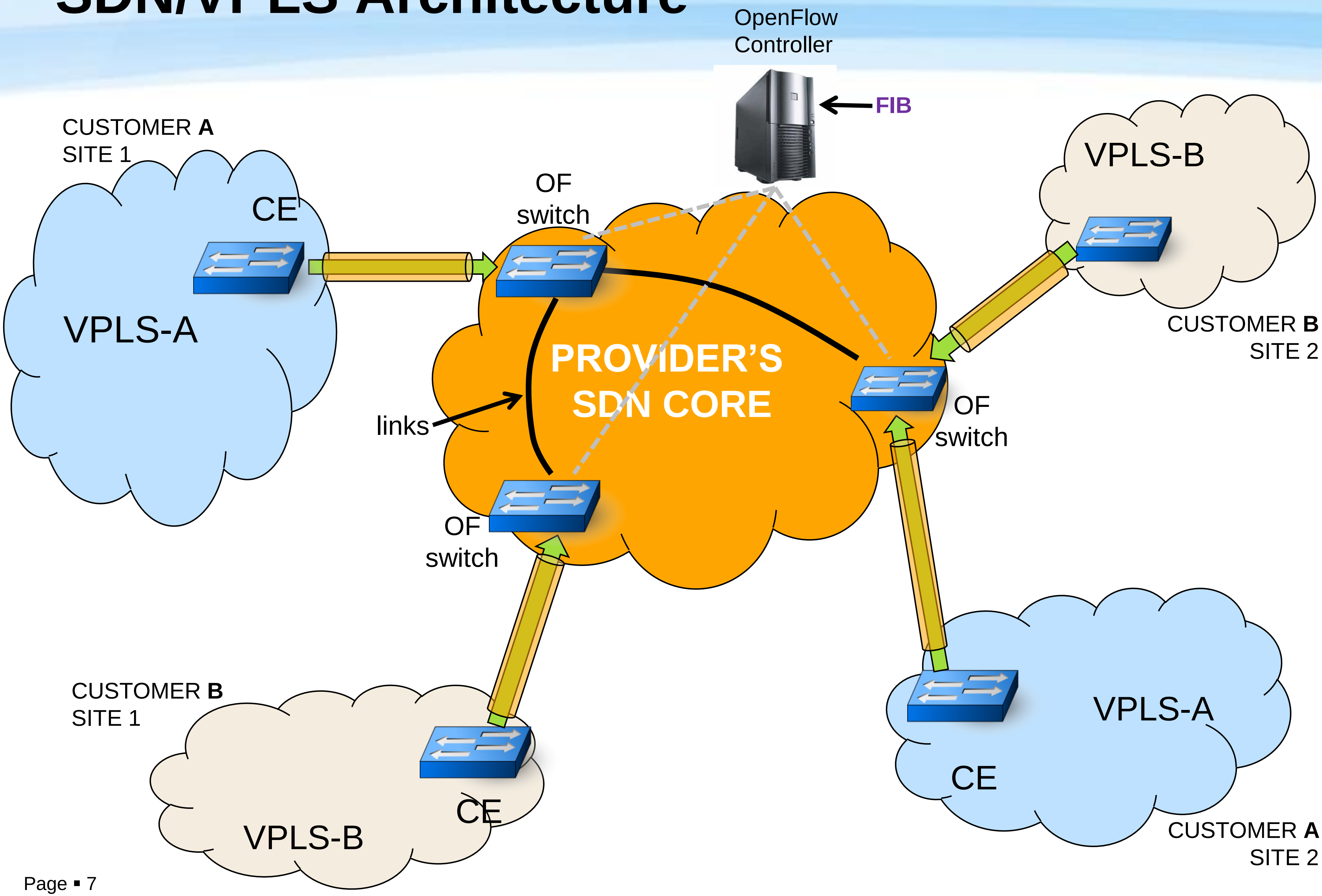


LSP = Label Switched Path: unidirectional path between LERs

LER = Label Edge Router (or PE = Provider Edge router)
LSR = Label Switching Router (or P = Provider router)

slide: Marijke Kaat

SDN/VPLS Architecture



OpenFlow 1.3

- Added support for MPLS
  - MPLS Label matching (ability to match more than one)
  - MPLS Label manipulation (push/pop/swap)

- Group tables allow multiple actions per flow.
  - e.g. for packet A send to port 10 AND change VLAN_ID and send to port 3.

SDN/VPLS vs MPLS/VPLS

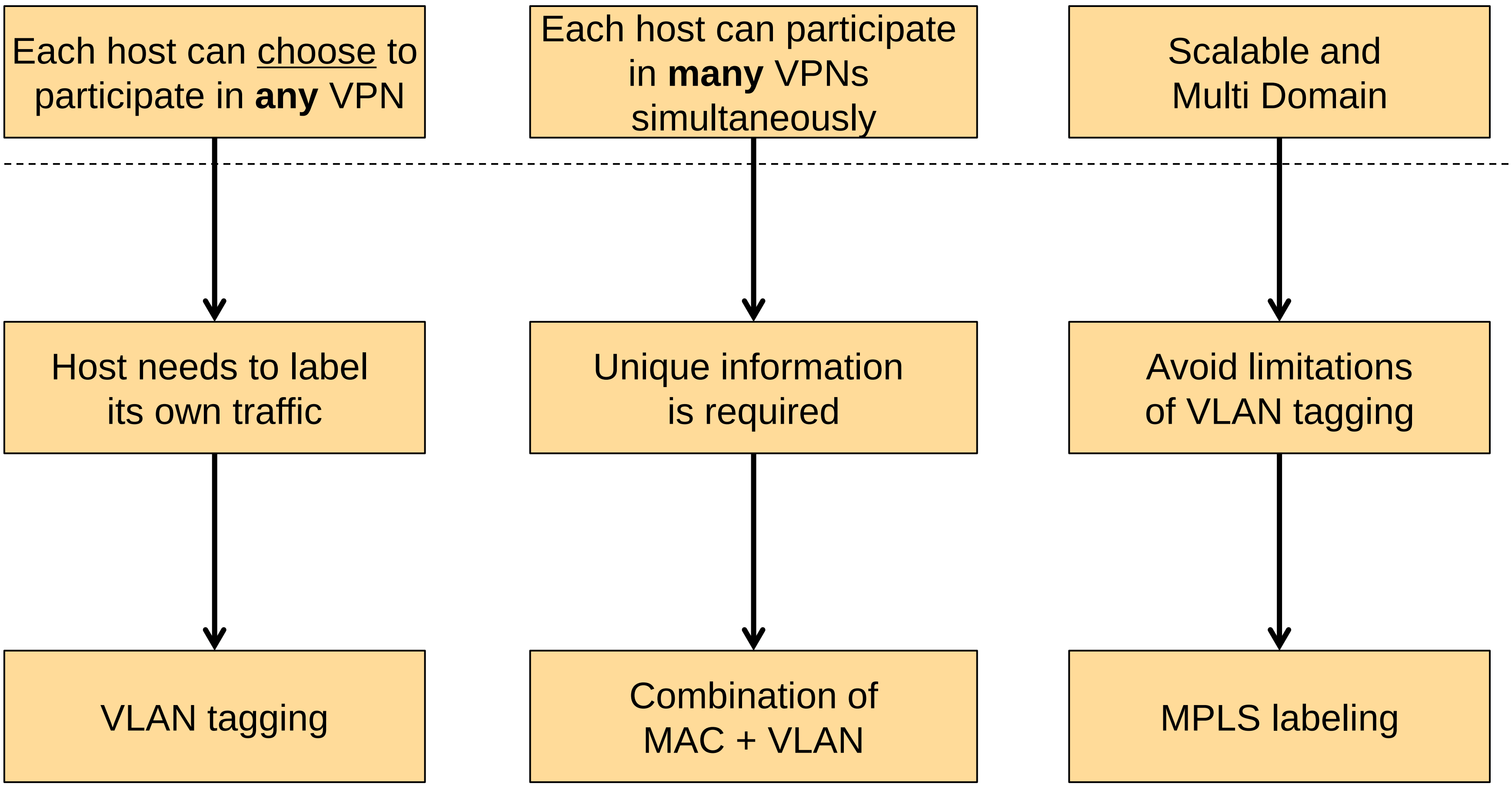
- Common OpenFlow switches replace PEs.

- No full mesh required.

- No pseudowires
  - No Signaling
  - No Label exchange

- Centralized Controller in commodity server with:
  - Network topology knowledge
  - FIB

Architecture requirements



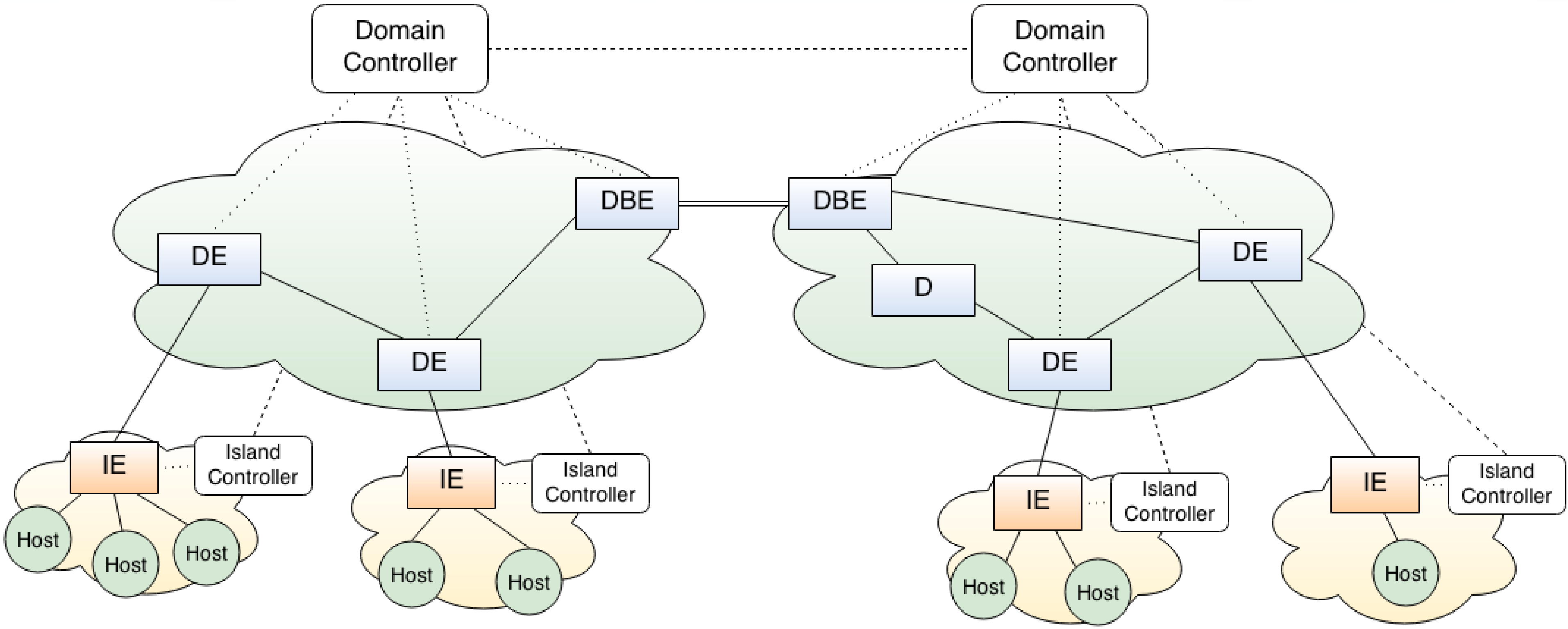
VPN representation

- Each VPN is represented by a **VPLS_ID** (MPLS label).
- Hosts define VPNs by VLAN (they are MPLS agnostic).
- Therefore, 4K VPNs can be represented in an island and 1M can be represented globally.
  - A mapping is required between local VLAN ID and global VPLS ID per island.

General Acknowledgements

- Inside an island, a **HOST** is defined as a unique combination of **MAC address + VLAN**
- Inside provider's domain, a **HOST** is defined as a unique combination of **MAC address + VPLS_ID**
- A **BROADCAST_MAC** is defined as a MAC address that is either the well-known Ethernet broadcast address or one of the easily recognizable Ethernet multicast addresses.
- **VPLS_ID** which is global and unique by representing VPN instances that can run simultaneously in the complete network.
- **ISLAND_ID** which is global and unique by representing the islands that participate in the complete network.

Architecture entities



- **DE :** Domain Edge device
- **D :** Domain device
- **DBE:** Domain Border Edge device
- **IE :** Island Edge device

2 Different solutions

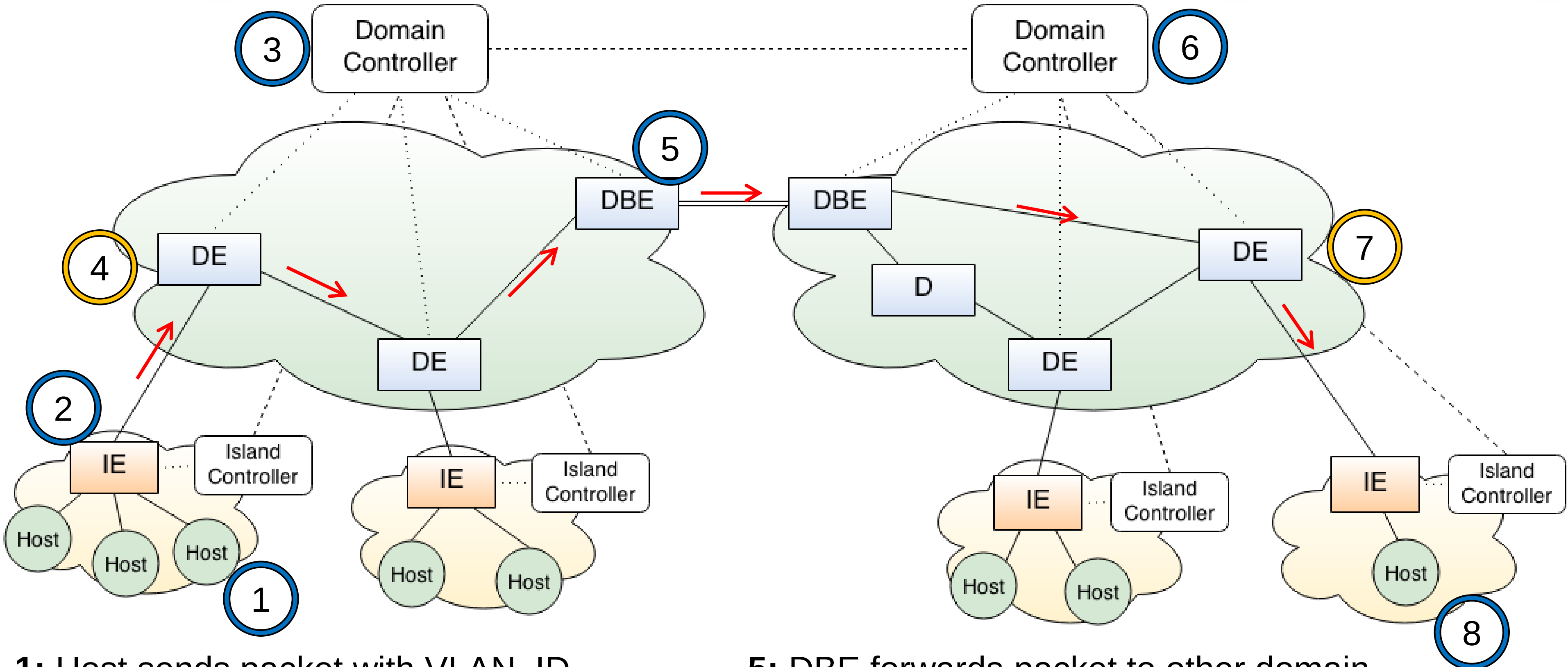
Core Labeling

- Islands are MPLS agnostic
- Uses 2 MPLS tags
  - Destination information
  - VPN information
- MAC Tables on both domain and island controllers

Island Labeling

- Core is MAC agnostic
- Uses 1 MPLS tag
  - Destination information (Unicast)
  - VPN information (Broadcast)
- All information is known to each island

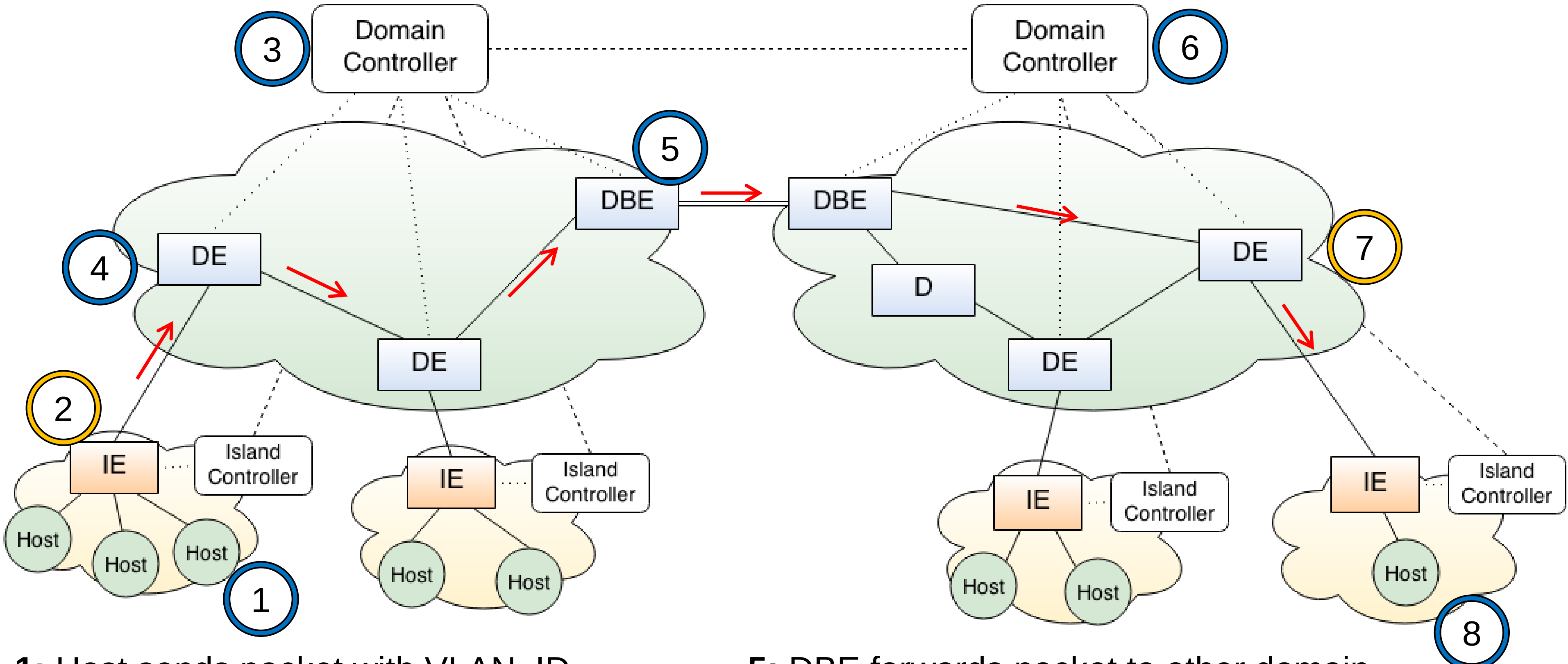
Core Labeling - Unicast



1: Host sends packet with VLAN_ID
2: IE forwards packet to Domain
3: Controller calculates shortest path to destination DBE and install flows
4: DE pushes ISLAND_ID + VPLS_ID

5: DBE forwards packet to other domain
6: Controller calculates shortest path to destination DE and install flows
7: DE pops MPLS tags and changes VLAN_ID
8: Host receives packet

Island Labeling - Unicast



1: Host sends packet with VLAN_ID

2: IE changes VLAN_ID, pushes ISLAND_ID and forwards to Domain

3: Controller calculates shortest path to destination DBE and install flows

4: DE forwards packets by ISLAND_ID

5: DBE forwards packet to other domain

6: Controller calculates shortest path to destination DE and install flows

7: DE pops MPLS tag and forwards to island

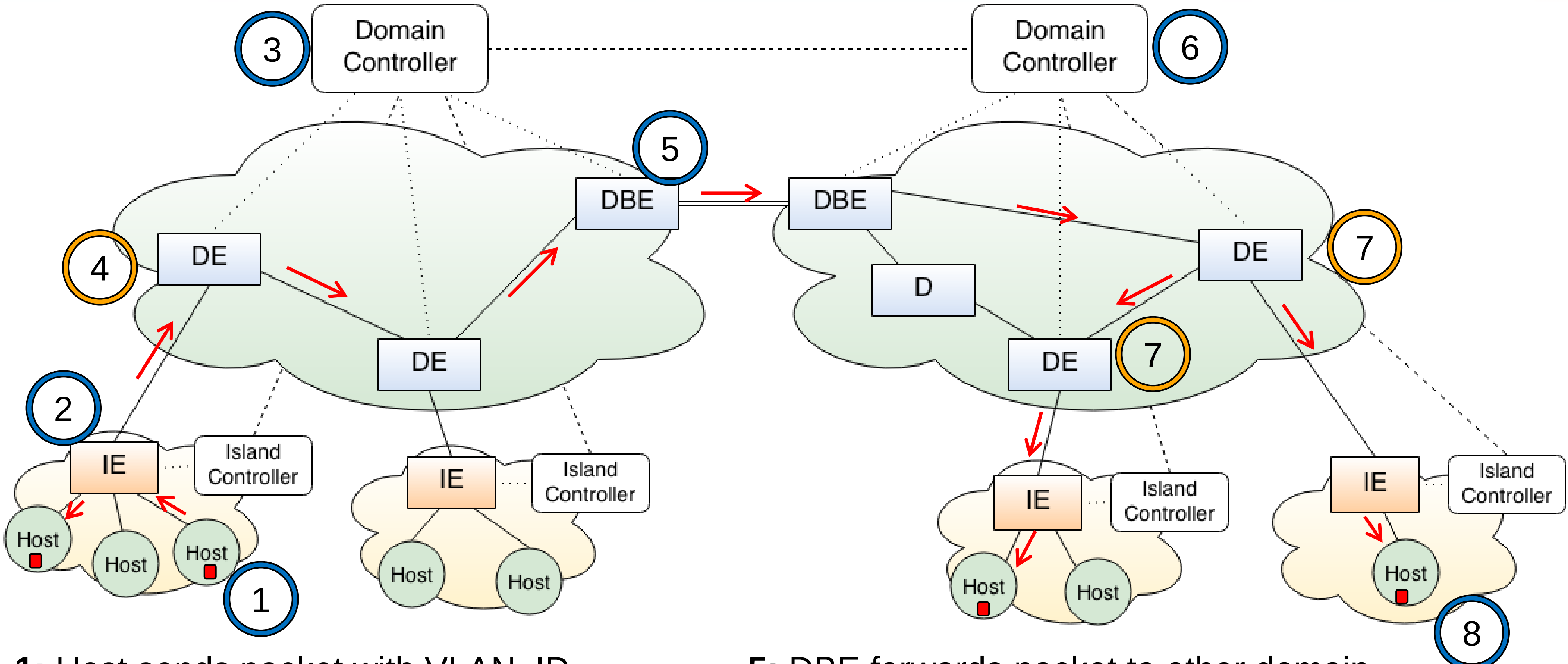
8: Host receives packet

Broadcast considerations

- Broadcast traffic can not be blindly flooded to all ports
  - Traffic isolation is ignored and privacy is violated !
    - Preconfiguration based on (PORT,VLAN) required
  - Split Horizon needed to avoid broadcast loops.

- Broadcast traffic must be as minimum as possible at core
  - Multicast trees are needed to forward traffic only to corresponding islands.

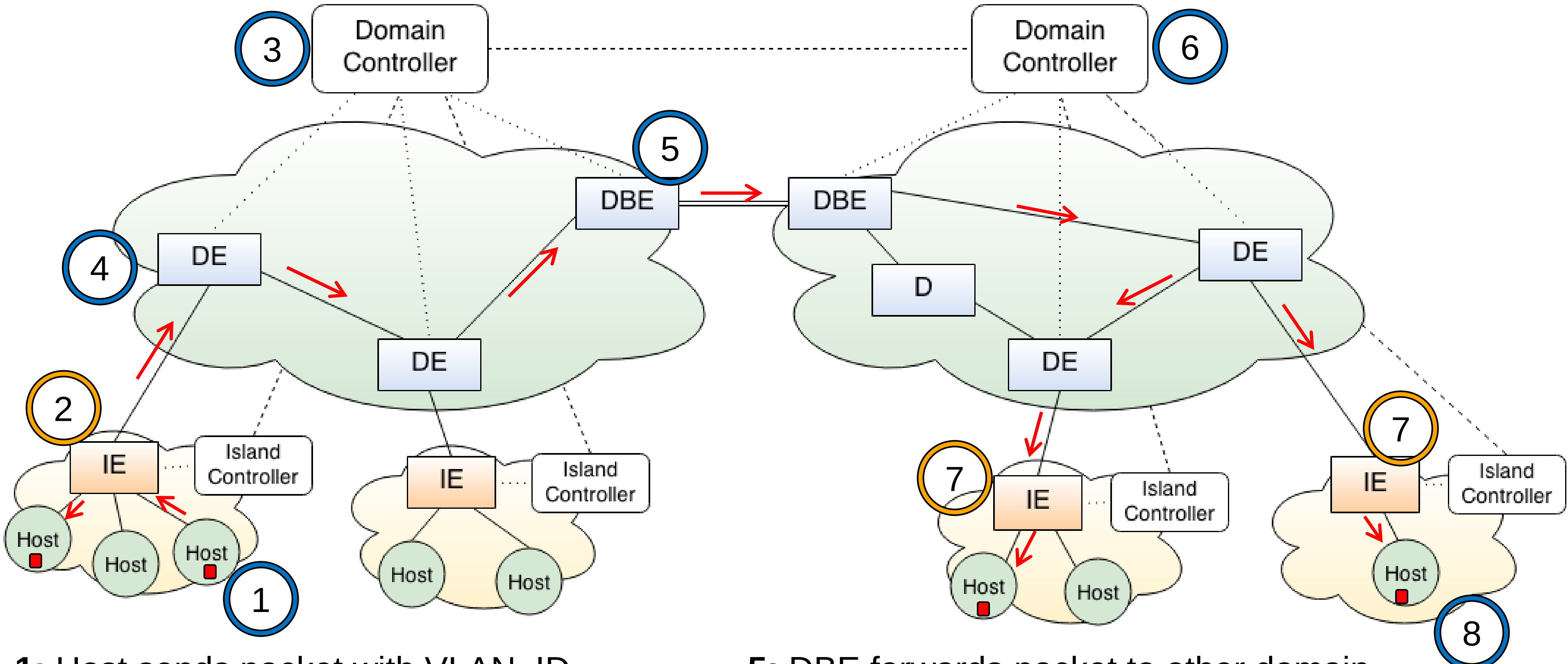
Core Labeling - Broadcast



1: Host sends packet with VLAN_ID
2: IE forwards packet to VPN ports
3: Controller creates multicast tree to VPN destination islands and install flows
4: DE pushes BRCAST_TAG + VPLS_ID

5: DBE forwards packet to other domain
6: Controller creates multicast tree to VPN destination islands and install flows
7: DE pops MPLS tags and changes VLAN_ID
8: Host receives packet

Island Labeling - Broadcast



1: Host sends packet with VLAN_ID

2: IE forwards packet to VPN host ports, AND pushes VPLS_ID + send to domain

3: Controller creates multicast tree to VPN destination islands and install flows

4: DE forwards packets by VPLS_ID

5: DBE forwards packet to other domain

6: Controller creates multicast tree to VPN destination islands and install flows

7: IE pops MPLS tag and changes VLAN_ID

8: Host receives packet

MAC Learning

- Based on OpenFlow Packet In events in order to combine (source) MAC addresses with PORT + VLAN

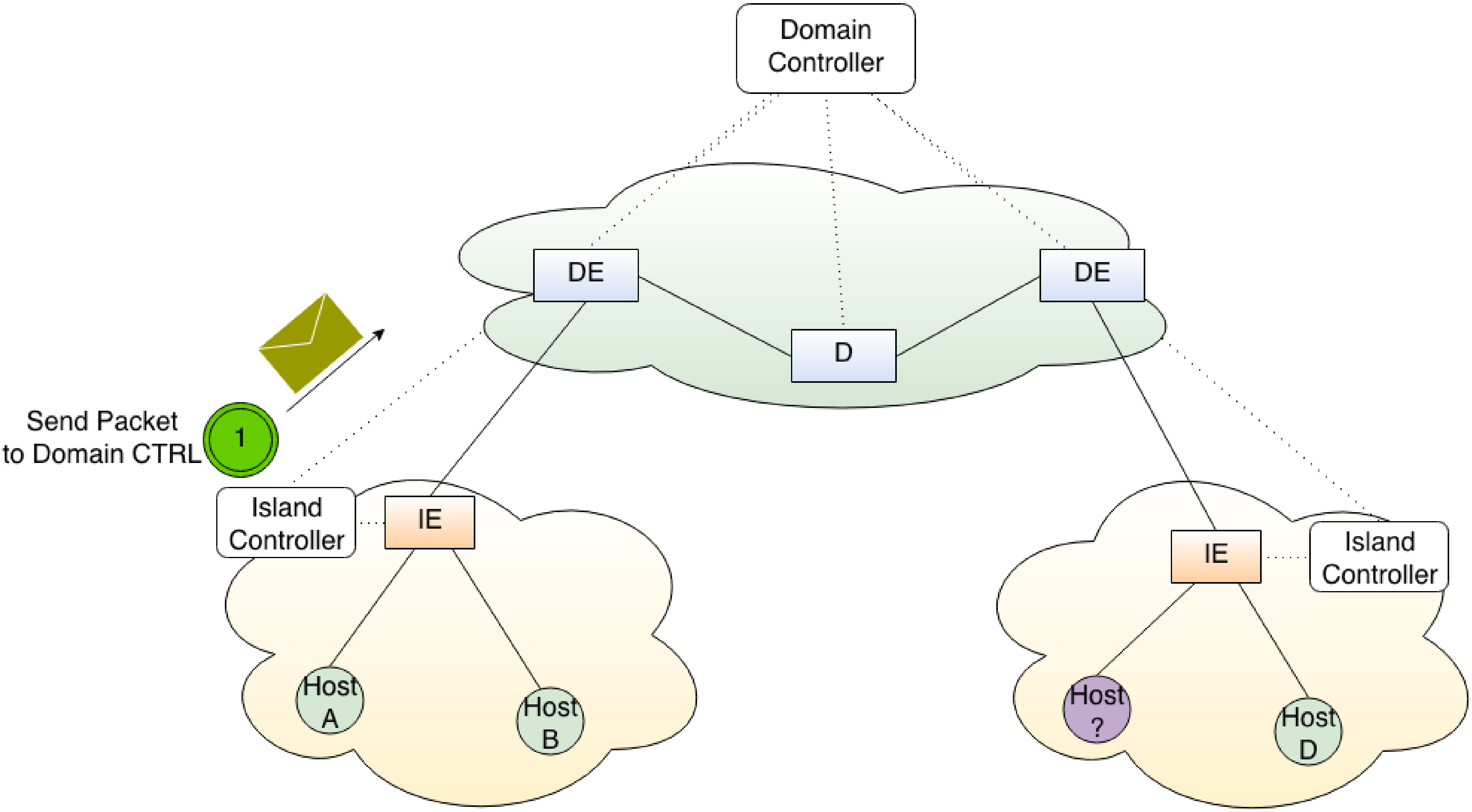
- Nevertheless, the ‘Unknown unicast’ **problem** exists:
“*Response traffic from unknown hosts may match existing flows and the MAC learning mechanism is skipped.*”

Solution

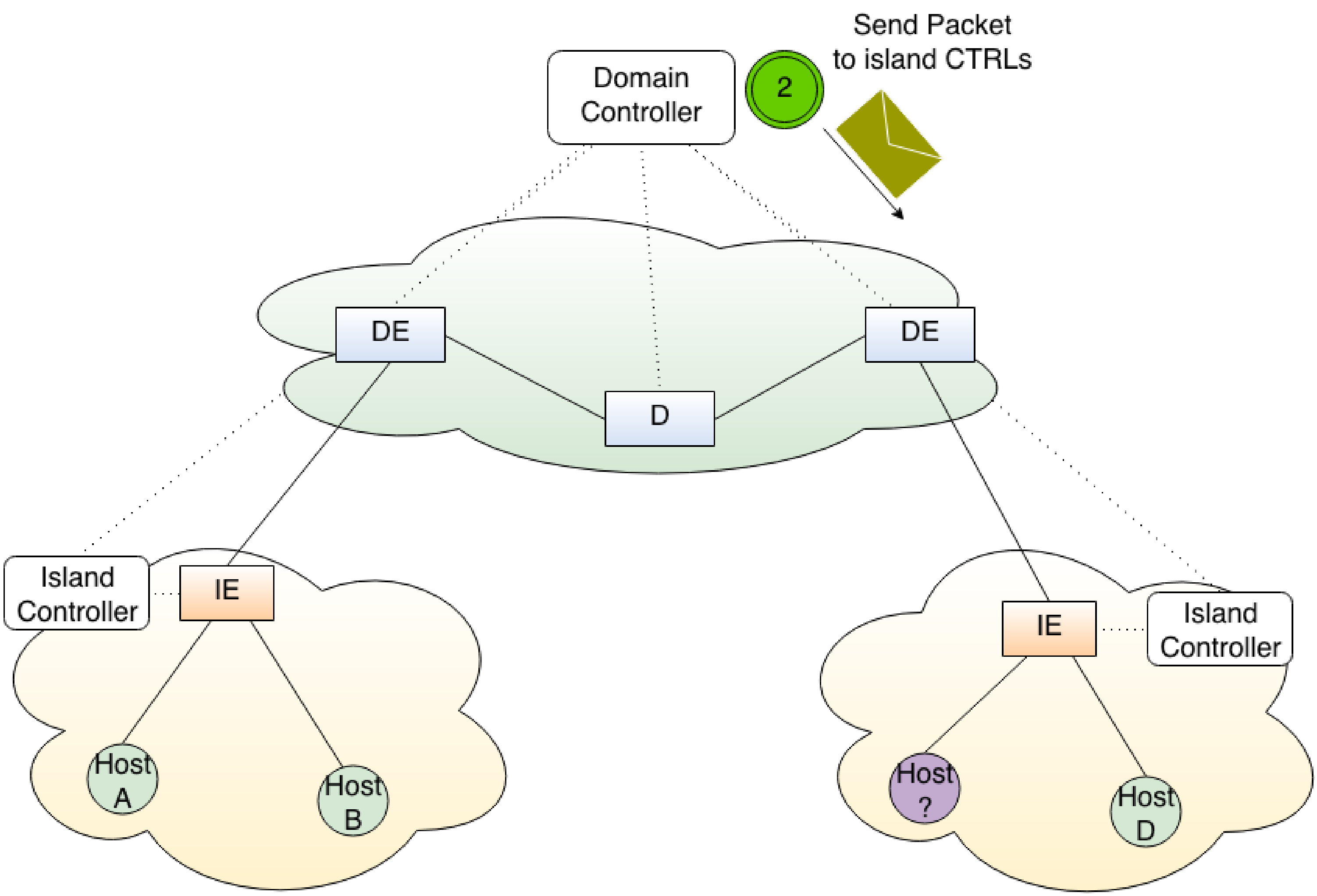
“*Skip global flooding and introduce a new host discovery mechanism*”*

*(Based on ForceMacLearning mechanism)

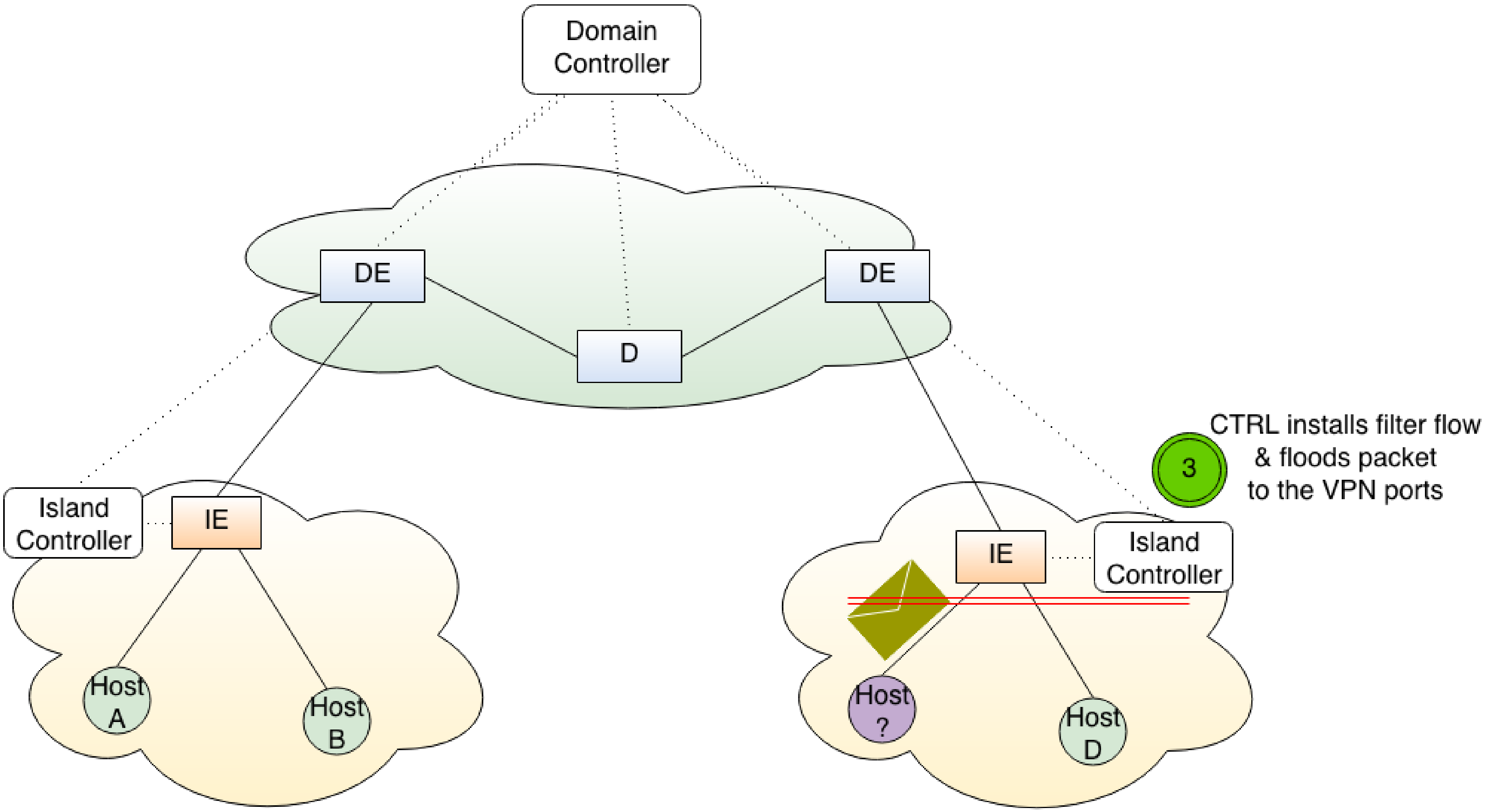
Solving Unknown Unicast



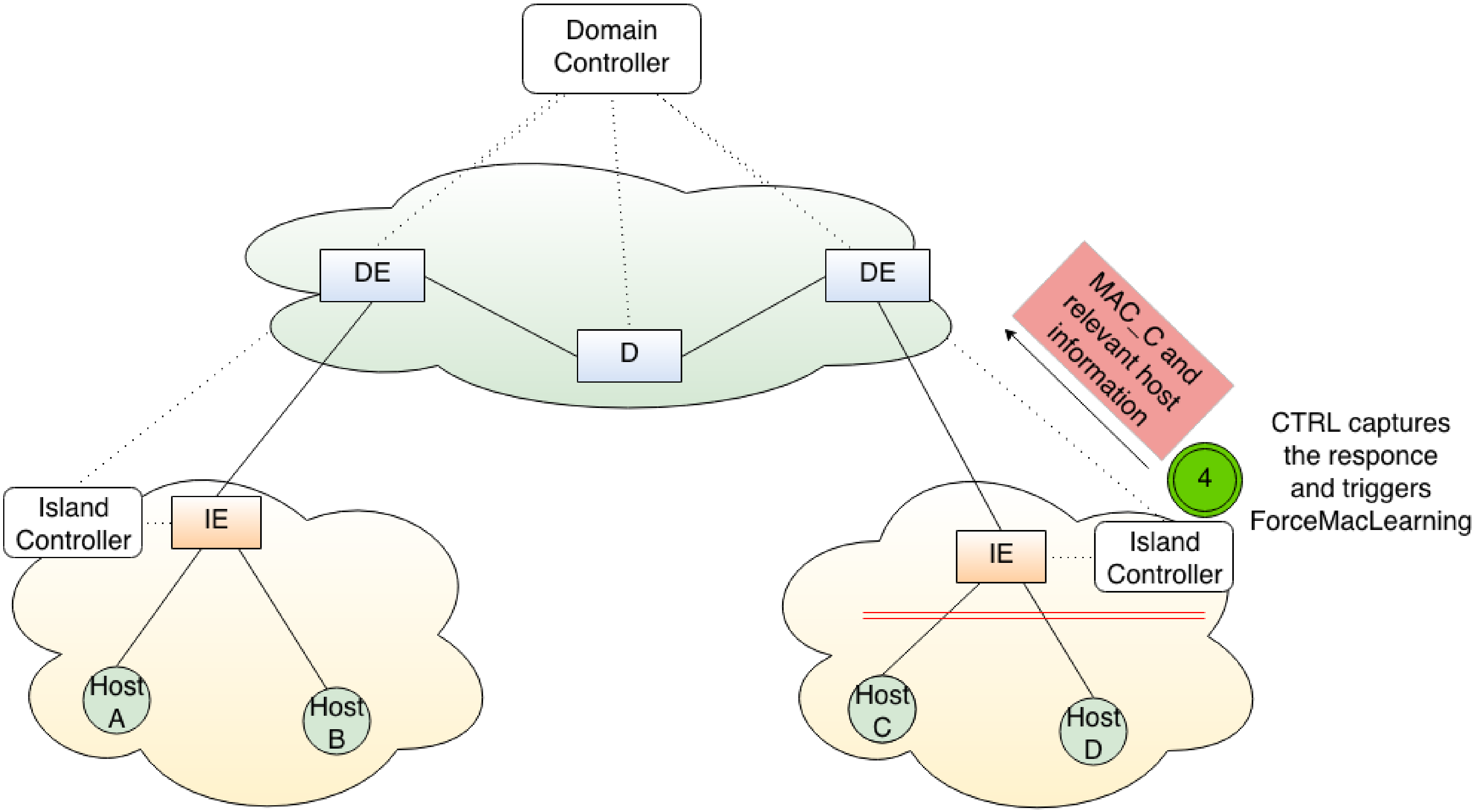
Solving Unknown Unicast



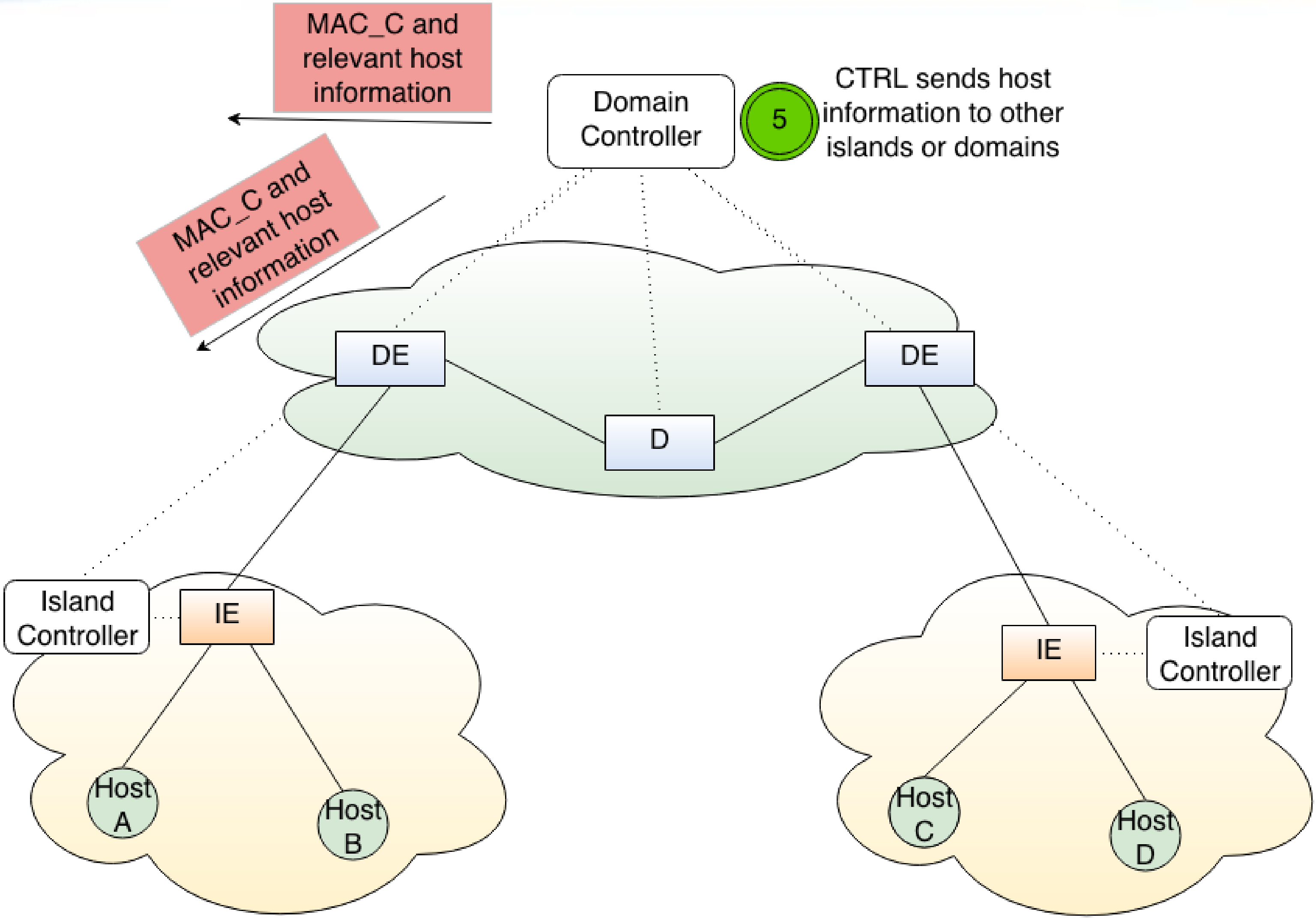
Solving Unknown Unicast



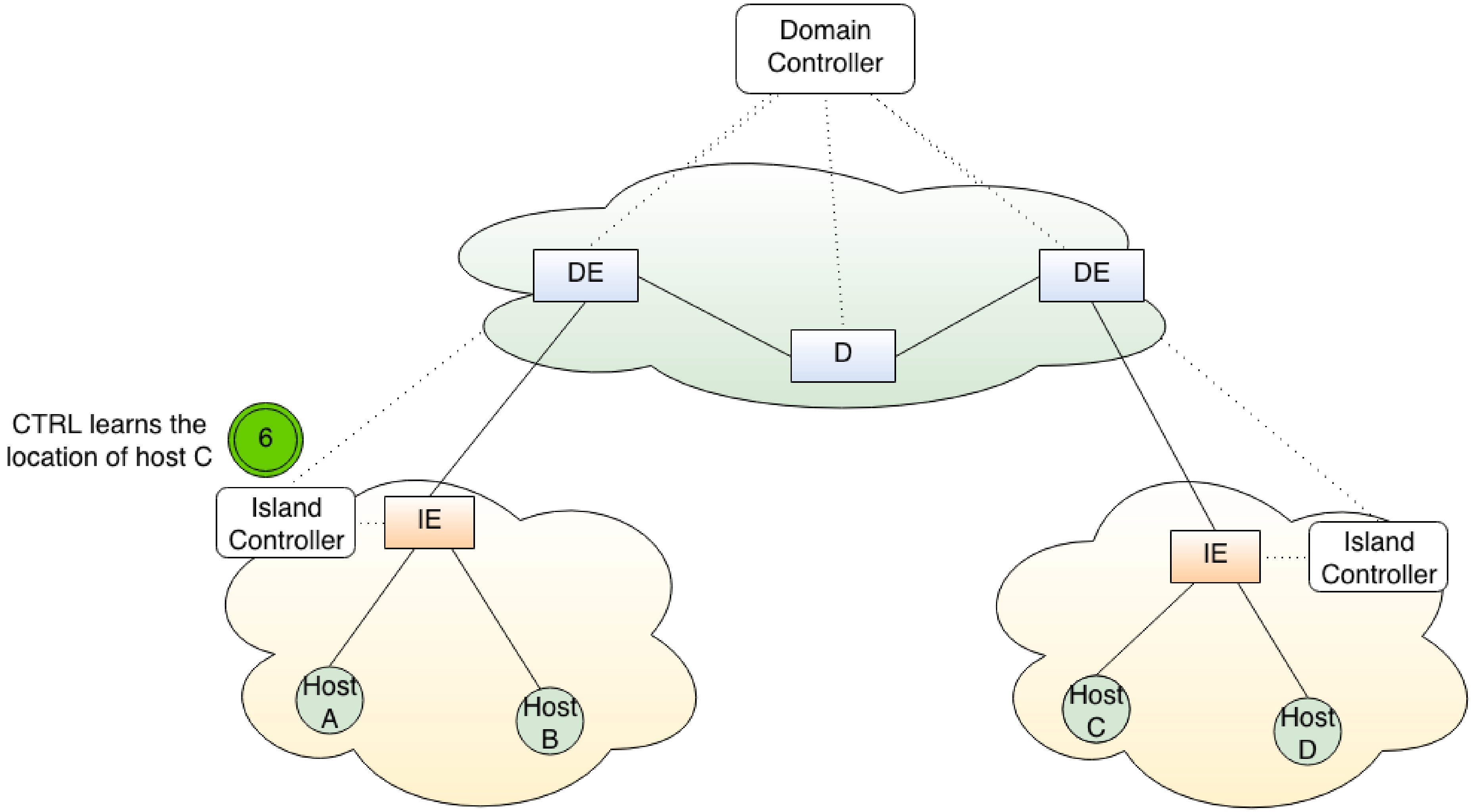
Solving Unknown Unicast - ForceMacLearning



Solving Unknown Unicast - ForceMacLearning



Solving Unknown Unicast - ForceMacLearning



Architecture analysis – Scalability (1/2)

Core labeling

- Able to support up to **1 048 575** islands in total.
- Requires **two** MPLS labels to operate

Customer Island

- Up to **4096** VPNs running simultaneously
- Unicast Flows at the OF Switch increase linearly by the number of hosts
- Broadcast Flows at the OF Switch increase by the combination of IN_PORT+VLAN ID

Provider's Domain

- Up to **1 048 575** VPNs running simultaneously. All islands can participate in any **4096** VPNs
- Unicast Flows at the DE switches increase linearly by the number of hosts
- Broadcast Flows at the OF Switches increase by the combination of VPLS_ID + INPORT

Architecture analysis – Scalability (2/2)

Island labeling

- Able to support up to **1 048 575** islands in total.
- Requires **one** MPLS label to operate

Customer Island

- Up to **4096** VPNs running simultaneously
- Unicast Flows at the OF Switch increase linearly by the number of hosts
- Broadcast Flows at the OF Switch increase by the combination of IN_PORT+VLAN ID

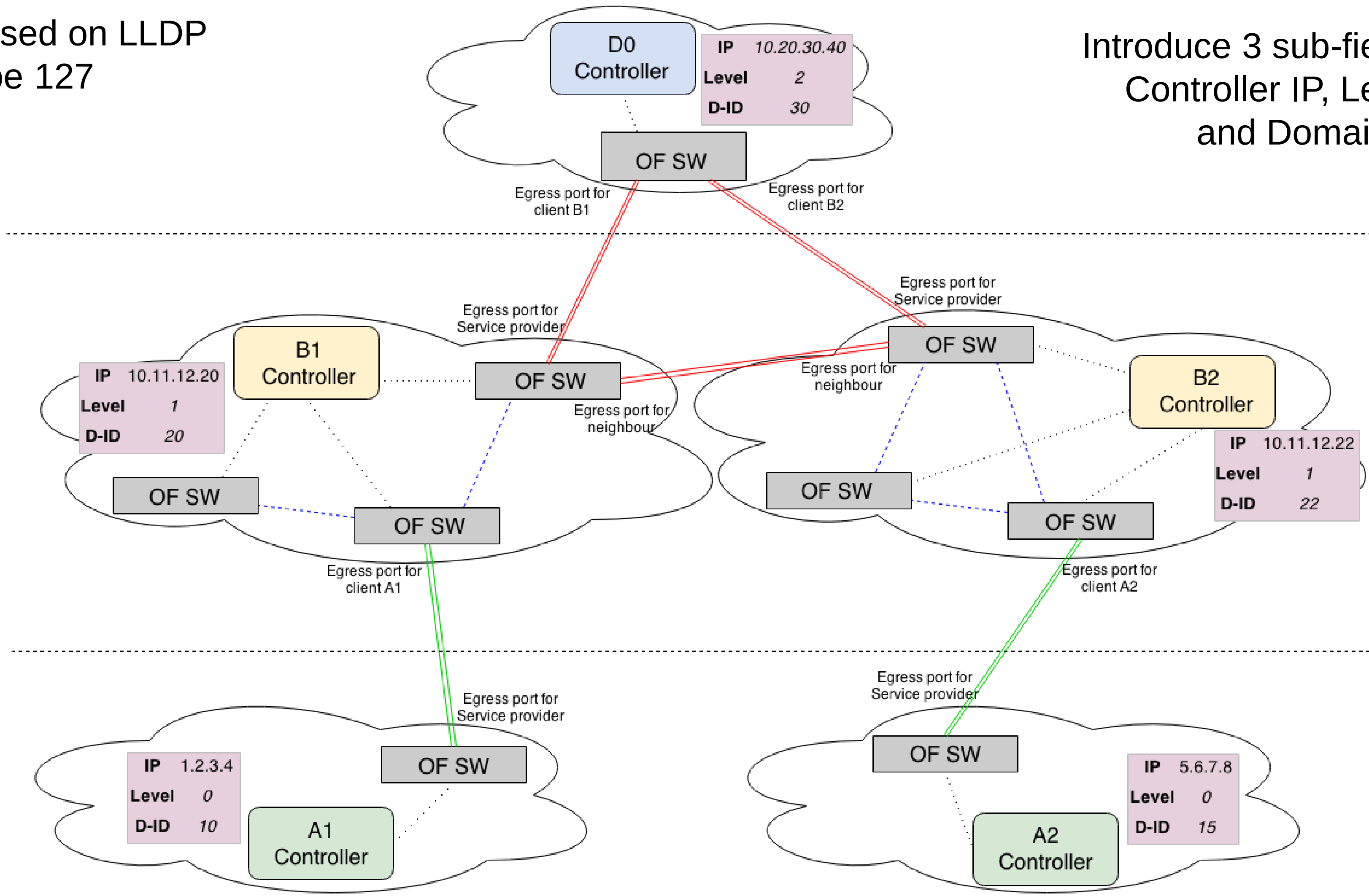
Provider's Domain

- Up to **1 048 575** VPNs running simultaneously. All islands can participate in any **4096** VPNs
- Unicast Flows at the OF Switches increase linearly by the number of islands
- Broadcast Flows at the OF Switches increase by the combination of VPLS_ID + INPORT

Optimizations/Ideas – M-Domain Discovery

Based on LLDP type 127

Introduce 3 sub-fields: Controller IP, Level and Domain ID



Optimizations/Ideas – Aggregation at core (Unicast Multi Domain traffic)

New MPLS tag

- Introduce Domain ID (20 bits) and let each provider choose its own unique identifier.
- Insert the Domain ID as an additional MPLS label at every packet needing to exit Provider's domain.
- Install flows at the core pointing to other provider domains. It will aggregate all the traffic from any VPN/Island.

Splitting MPLS TAG

- Introduce Domain ID (8 bits) and let each provider choose its own island identifiers.
- Separate the MPLS Label at Domain and Island Part:
  150 40 = LABEL
  10010110 000000101000 = 614440
- MAX 256 Domains and 4096 islands per Domain.
- Flows matching one MPLS label.

Discussion

Positives

- Efficient and flexible on demand VPLS services able to interconnect millions of hosts in thousands of customer sites.
- Scalable and easily extendable architecture which is able to work in a Multi Domain environment.
- Network programmability allows automation and design freedom.
- Architecture can be implemented in the near future based on OpenFlow 1.3

Negatives

- A Controller-to-Controller communication channel needs to be defined
- A modified OpenFlow Controller needs to be implemented covering the requirements of our architecture
- Scalability ends where combined protocols stop scale (e.g MPLS Label, VLAN ID).

Conclusion

- SDN technology provides the flexibility to design a complete network architecture for VPLS.

- Capabilities of OpenFlow expand through different combinations with other protocols.

- Network designers can build an abstract underlay SDN network and deploy multiple services on top of it.

Future work

- Extend architecture to handle Multicast traffic (possible at layer 2 via RFC 1112).
- Extend Architecture for QoS Considerations (based on OF 1.3 Meter table)
- Implementation of the architecture to verify that SDN is a flexible, production ready technology.
- Practical Performance evaluation (based on OF 1.3)

Thank you

Software Defined VPNs