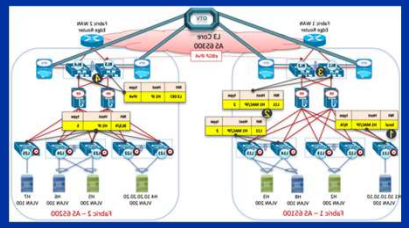
School of Advanced Networking by NAMEX

Reiss Romoli

Networking dei *Data Center*

Dai DC L2 ai DC per l'AI



Tiziano Tofoni

Copyright Tiziano Tofoni

Firmato digitalmente da: TOFONI TIZIANO
Data: 21/05/2025 13:17:16

Note di *copyright*

REISS ROMOLI

- Questo insieme di diapositive, incluse le pagine note, è protetto dalle leggi sul *copyright* e dalle disposizioni dei trattati internazionali. Il titolo ed i *copyright* relativi alle diapositive (ivi inclusi, ma non limitatamente, ogni immagine, fotografia, animazione, video, audio, musica e testo), in accordo con gli artt. 12 e seguenti della Legge 633/1941, sono di proprietà dell'autore Tiziano Tofoni (di seguito 'l'autore').

- Le diapositive possono essere utilizzate esclusivamente per scopi di studio nell'ambito dei corsi tenuti dall'autore.

- Ogni altra utilizzazione o riproduzione (ivi incluse, ma non limitatamente, le riproduzioni su supporti ottici/magnetici, su reti di calcolatori o stampate) in toto o in parte è vietata, se non esplicitamente autorizzata per iscritto, a priori, da parte dell'autore.

- L'informazione contenuta in queste diapositive è ritenuta essere accurata alla data della pubblicazione. Essa è fornita per scopi meramente didattici e non per essere utilizzata in progetti di impianti, prodotti, reti, ecc. In ogni caso essa è soggetta a cambiamenti senza preavviso. L'autore non si assume alcuna responsabilità per il contenuto di queste diapositive (ivi incluse, ma non limitatamente, la correttezza, completezza, applicabilità, aggiornamento dell'informazione).

- In ogni caso non può essere dichiarata conformità all'informazione contenuta in queste diapositive.

- In ogni caso questa nota di *copyright* non deve mai essere rimossa e deve essere riportata anche in utilizzi parziali.

AGENDA

I° modulo: Architetture dei *Data Center*

II° modulo: *Overlay Virtual Networks*

III° modulo: Ethernet VPN (EVPN)

IV° modulo: Integrazione EVPN/VXLAN

V° modulo: Aspetti architetturali

VI° modulo: DC per l'AI

Copyright Tiziano Tofoni

I° modulo

Architetture dei *Data Center*

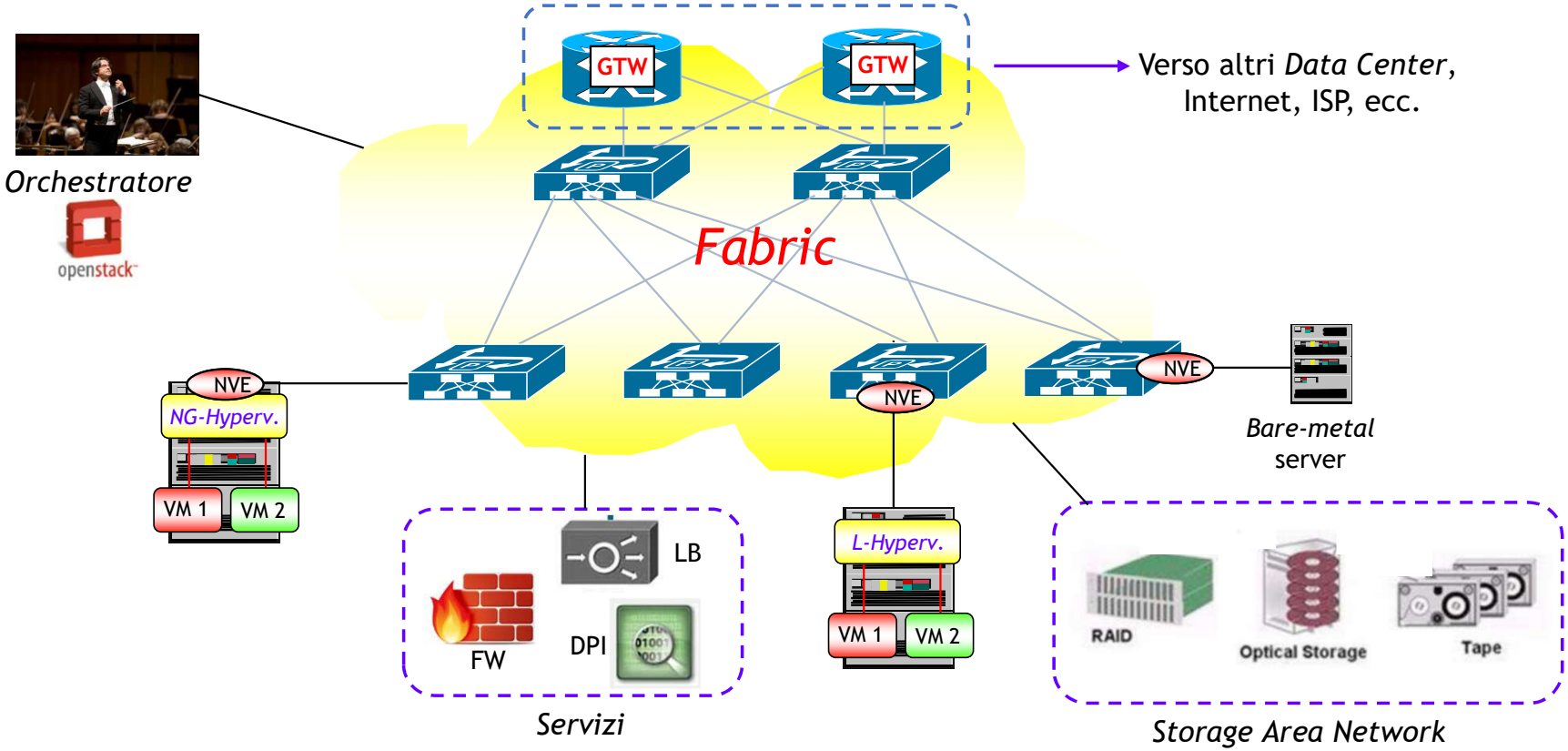
Copyright Tiziano Tofoni

I° modulo: Architetture dei *Data Center*

#1	Dai DC L2 ai DC L3
#2	*Clos IP Fabric*
#3	Aspetti di routing IP

Copyright Tiziano Tofoni

Architettura di un (moderno) *Data Center*



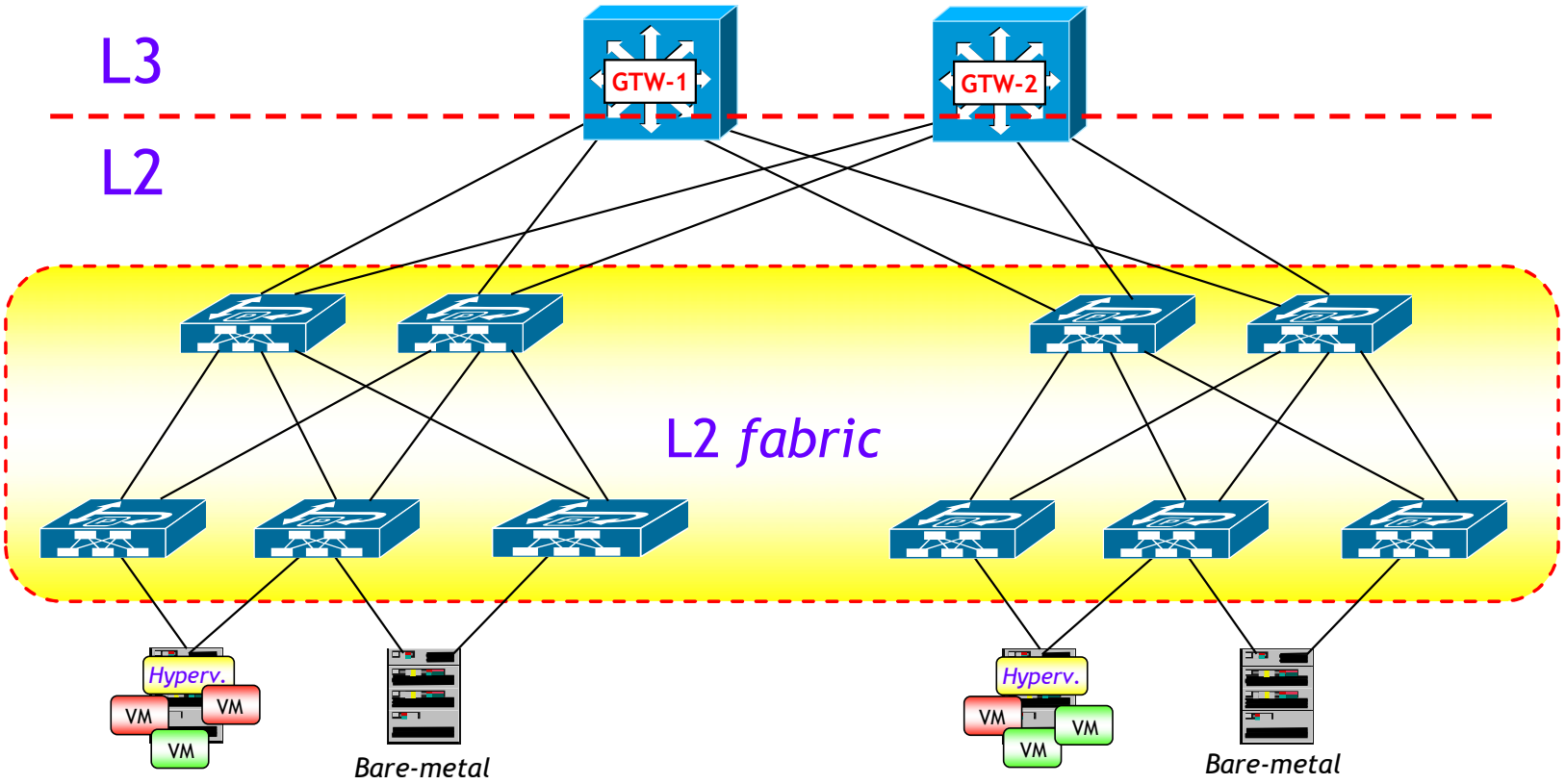
- *NG-Hyperv.* = *Next Generation hypervisor* (di nuova generazione)
- *L-Hyperv.* = *Hypervisor legacy* (di vecchia generazione)
- *NVE* = *Network Virtualization Endpoint*

Fabric: L2 o L3?

REISS ROMOLI

- Primo dilemma
  - *Fabric* L2 (= rete switched Ethernet)
  - *Fabric* L3 (= rete IP, IP+MPLS)
- Fabric L2
  - Switch di accesso e di aggregazione L2
  - Necessità di *Spanning Tree* per evitare *loop* e *broadcast storm*
- Fabric L3
  - Piattaforme L3 (router) sia in accesso che a livello di aggregazione
  - Connettività IP via protocolli di routing classici

L2 *fabric*



L2 switch

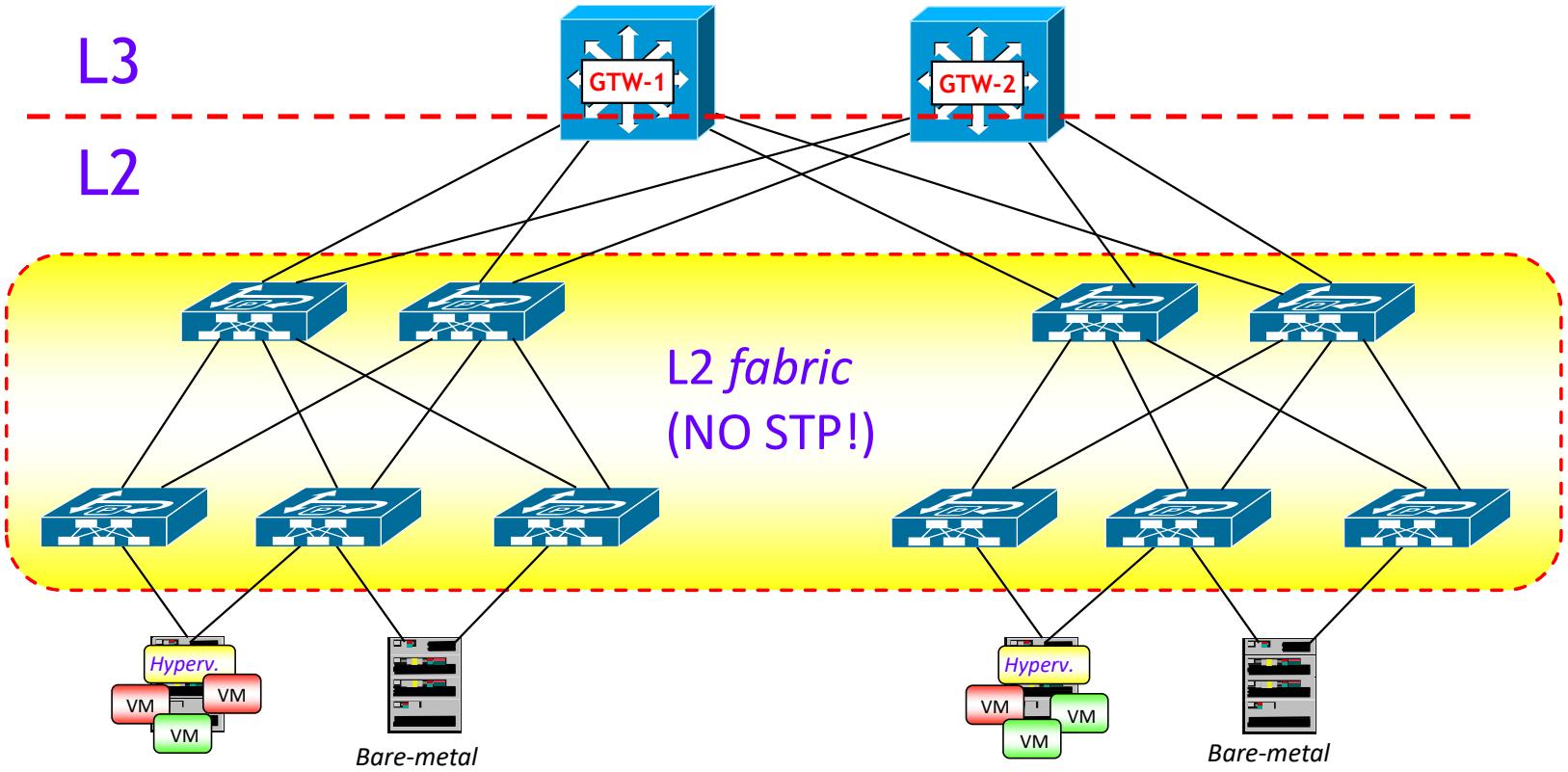
Perché L2 *fabric*?

- Consente un setup *plug-and-play*
- Configurazioni ridotte a minimo
- Facilita la mobilità delle *Virtual Machine*

Problemi delle L2 *fabric*

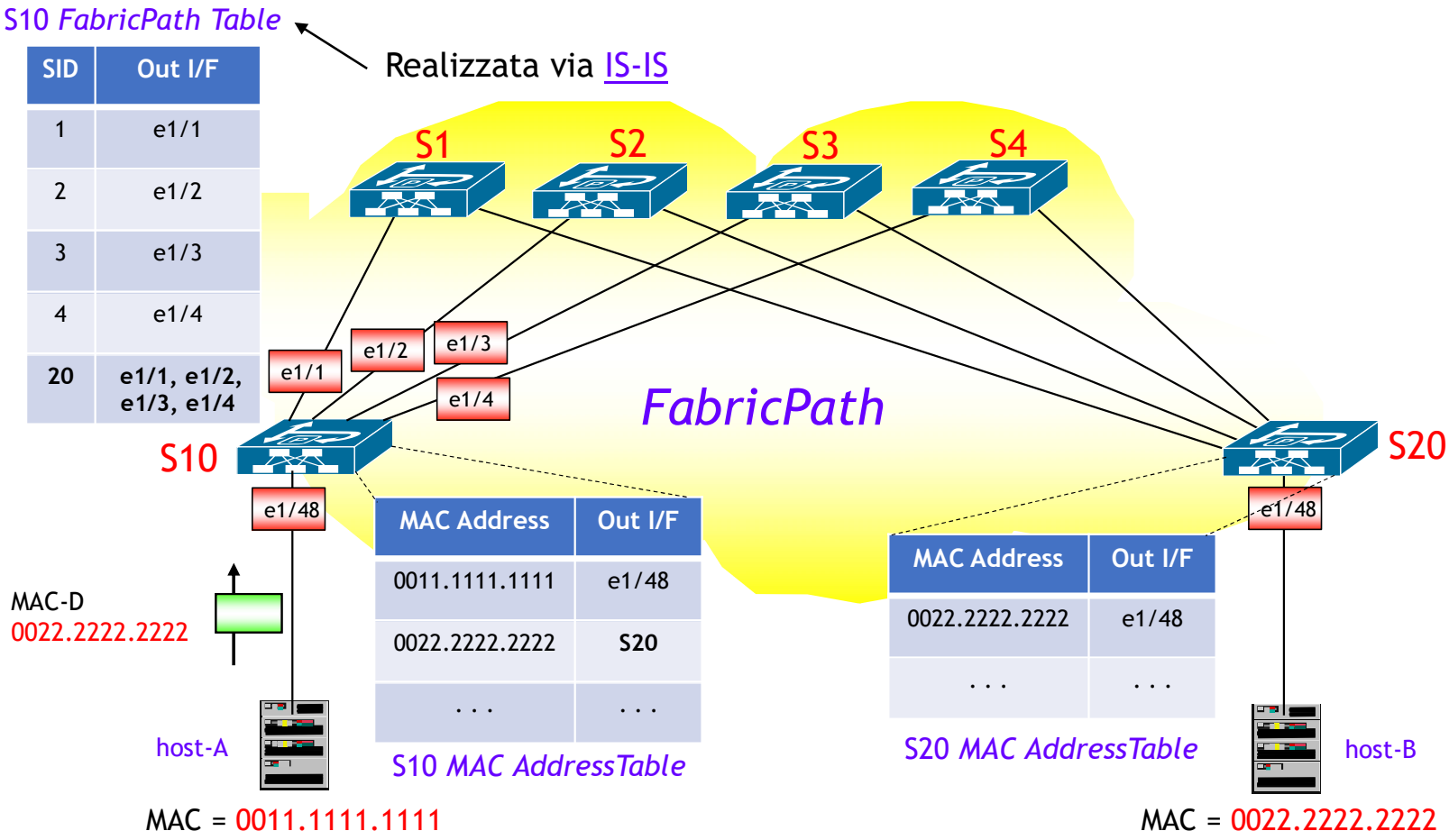
- Problemi locali hanno un impatto globale
  - Il meccanismo del *flooding* può causare *broadcast storm* che bloccano l'intera rete
- L'utilizzo del protocollo STP ha molti svantaggi
  - Riduzione consistente della *bisectional bandwidth*
    - STP è basato sul principio delle porte nello stato *blocking*
    - *vPC/MLAG* aiutano a mitigare l'effetto, ma non sono una soluzione!
  - Convergenza troppo lenta (anche nella sua versione *Rapid STP*)
  - Cammini L2 non ottimi a causa della struttura ad albero con *Root Bridge* fisso
  - Mancanza di funzionalità di *Load Balancing*
- Scalabilità della tabella CAM (=*MAC address table*)
  - Impossibilità di aggregare gli indirizzi MAC
- Solo 12 bit per la segmentazione
  - 12 bit = 4.096 VLAN, insufficienti per le moderne applicazioni *Cloud*

L2 *fabric* senza STP (L2MP, L2 MultiPath)



- Due nuovi protocolli
  - *TRILL* - *TR*ansparent *I*nterconnection of *L*ots of *L*inks (standard IETF)
  - *SPB* – *S*hortest *P*ath *B*ridging (standard IEEE802.1aq)

Esempio: Cisco *FabricPath* (1/2)



Esempio: Cisco *FabricPath* (2/2)

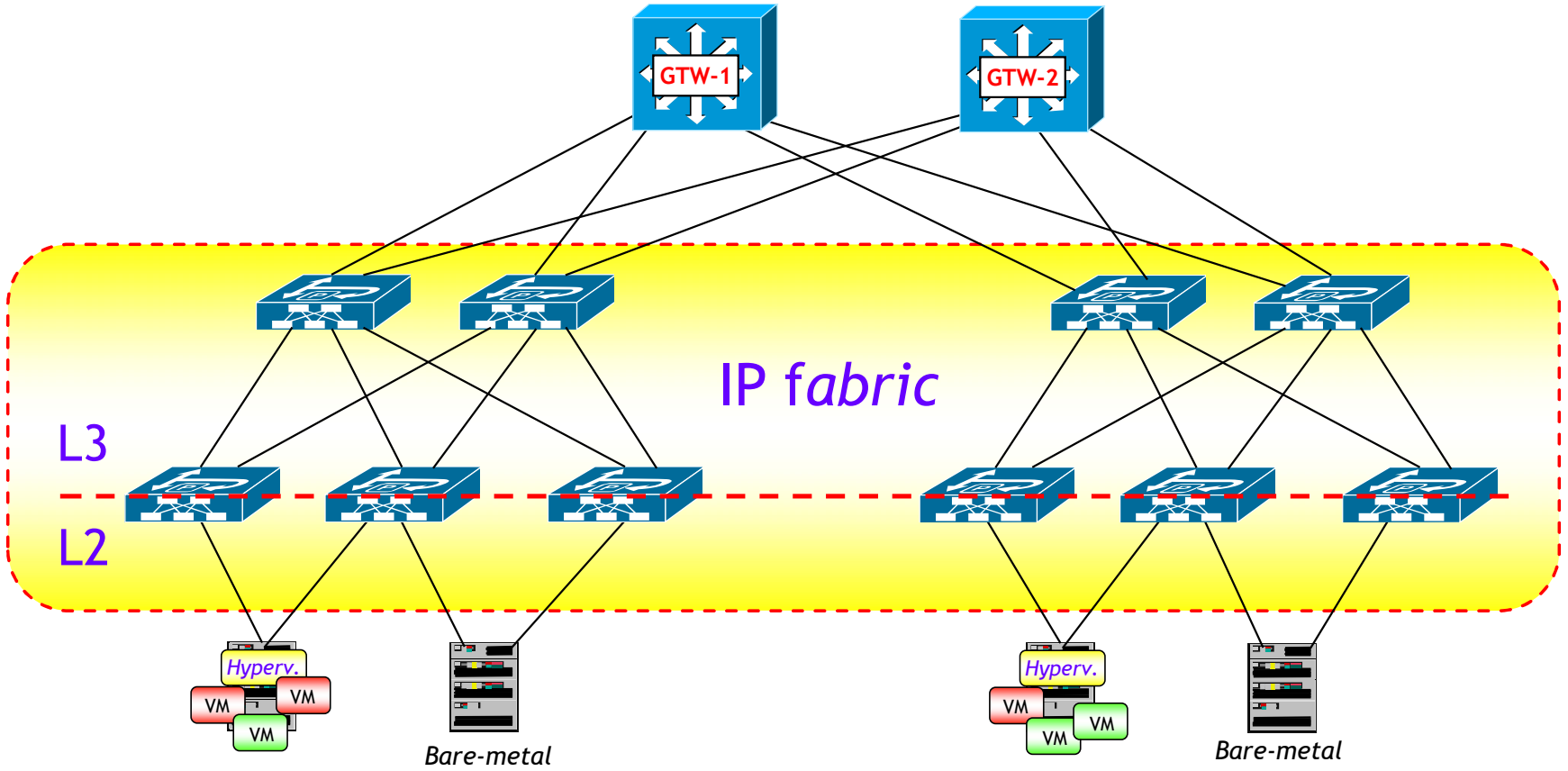
REISS ROMOLI

- Cisco *FabricPath* è supportato nelle piattaforme Nexus 7k/9k
- Vantaggi
  - Eliminazione completa del protocollo STP nella *Fabric*, e di tutte le sue problematiche
  - Possibilità di sfruttare i vantaggi dei protocolli di routing convenzionali (ECMP, (quasi) assenza di *loop*, convergenza veloce, ecc.)
  - Migliore gestione del traffico BUM attraverso i *multidestination tree*
  - Tabelle MAC più piccole grazie al meccanismo di *MAC learning* più efficiente (*conversational learning*)
  - Semplicità di configurazione

```
N7K(config)# interface ethernet 1/1
N7K(config-if)# switchport mode fabricpath
```

- Svantaggi
  - È un protocollo proprietario (derivato dal protocollo TRILL)
  - Alcuni problemi delle *Fabric* L2 rimangono (scalabilità delle tabelle CAM, VLAN ID troppo piccolo, *flooding* del traffico BUM)

L3 (IP) *fabric*


GTW-1
GTW-2
IP fabric
L3
L2
Hyperv.
VM
VM
VM
Bare-metal
Hyperv.
VM
VM
VM
Bare-metal

Perché L3 *fabric*?

REISS ROMOLI

- Elevata scalabilità
  - Possibilità di aggregazione dei prefissi IP
  - Un esempio su tutti: la rete Internet!
- Problemi locali NON hanno un impatto globale
- Vantaggi dei protocolli di routing IP
  - Ottimizzazione dei percorsi e della distribuzione del traffico
  - Completo sfruttamento della banda disponibile
  - Convergenza veloce
  - *Loop* assenti o mitigati
  - . . .

I° modulo: Architetture dei *Data Center*

#1	Dai DC L2 ai DC L3
#2	*Clos IP Fabric*
#3	Aspetti di routing IP

Copyright Tiziano Tofoni

La *fabric* – il cuore dei *Data Center*

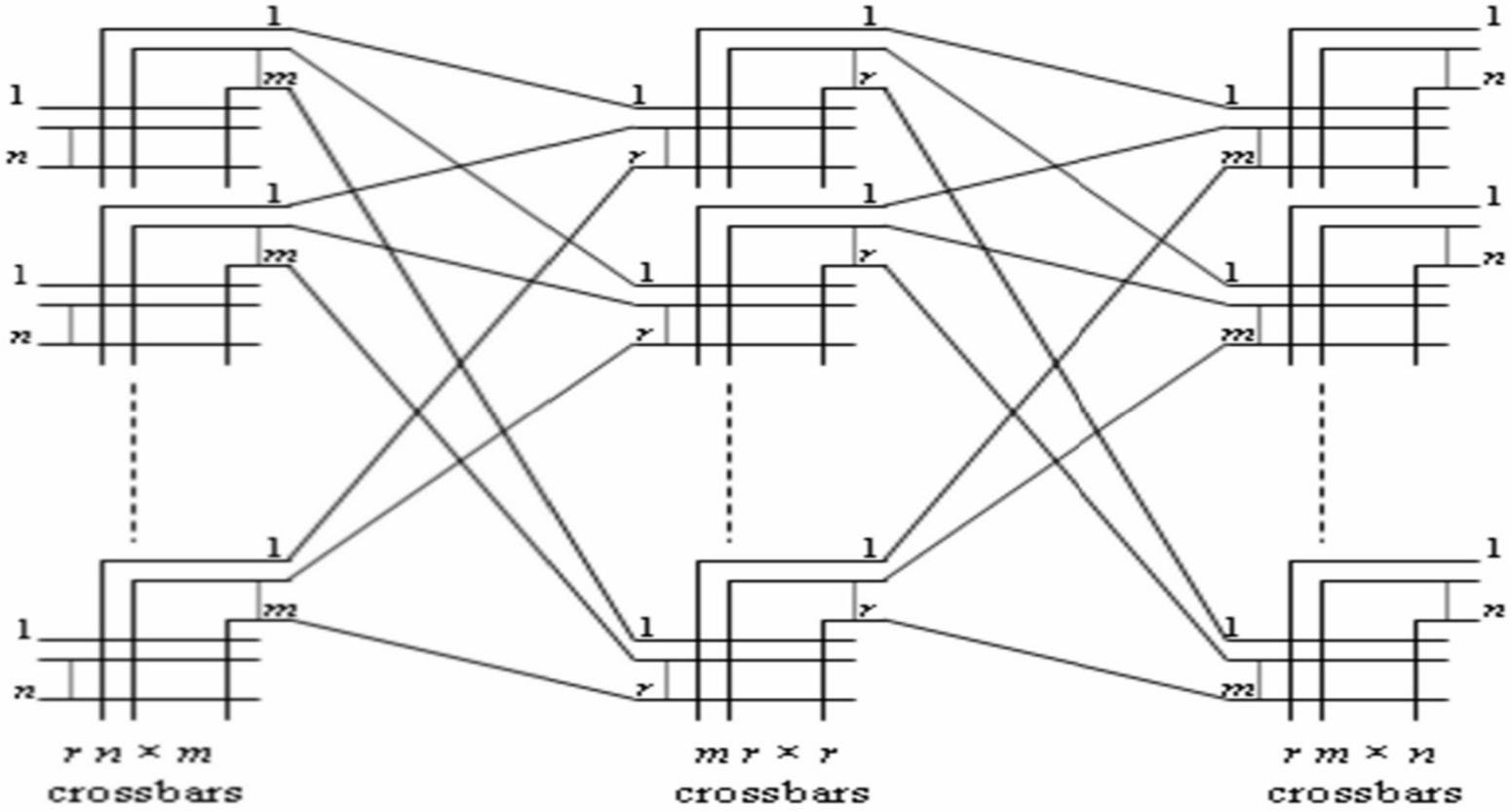
REISS ROMOLI

"The place to begin in the data center design is not with the physical cabling, or cooling, or power, but with the concept of a fabric…"

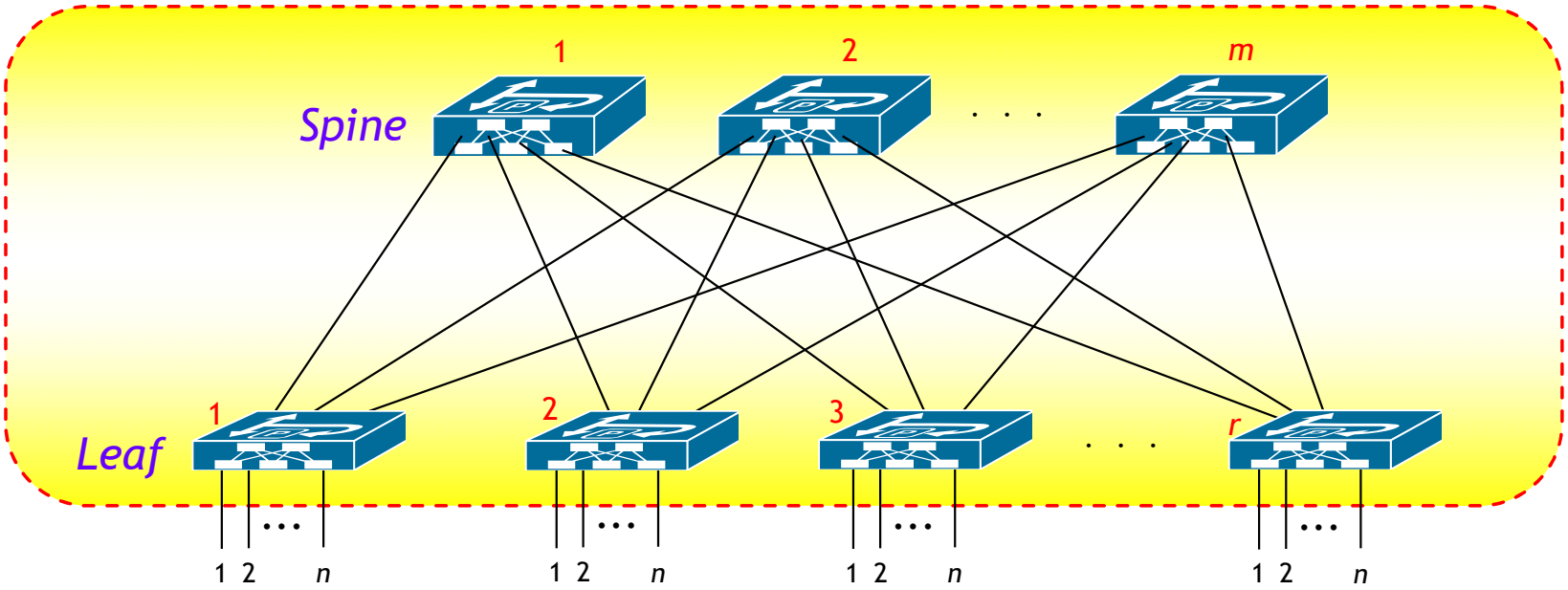
Fonte: Russ White, Denise Donohue - *The Art of Network Architecture (Business-Driven Design)*, Ed. Cisco Press, 2014

Prima un po’ di storia...

- Le matrici di commutazione multistadio (Charles Clos, marzo 1953)

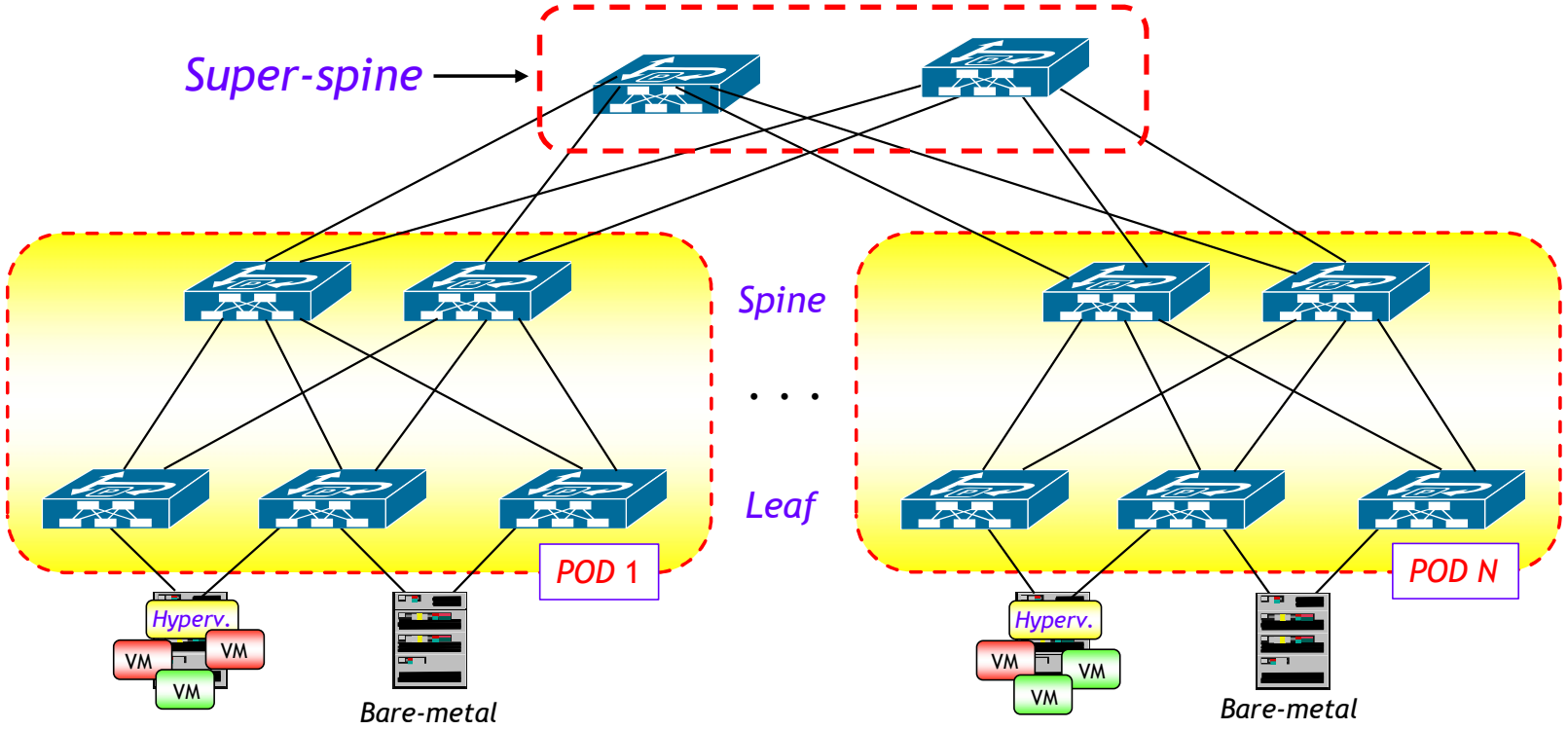


Clos IP fabric – 3 *stage*



- Anche nota come architettura *Leaf-and-Spine*
  - *Spine* – switch (router) di transito
  - *Leaf* – switch (router) di accesso

Clos IP fabric – 5 *stage*



- Ripartizione in POD (*Point Of Delivery*)
  - Ciascun POD è di tipo *Leaf-and-Spine*
  - Rete inter-POD costituita da *Super-spine*

Pro e contro

REISS ROMOLI

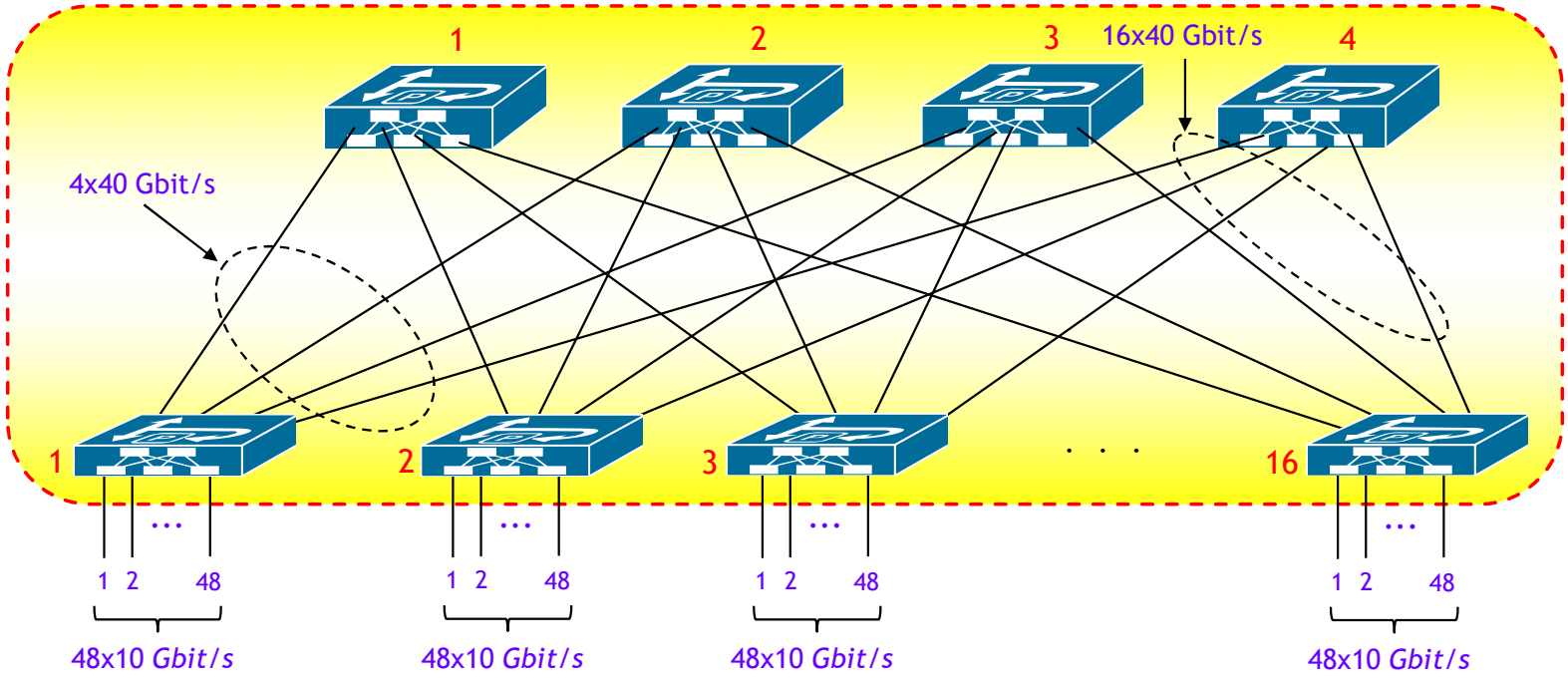
- Semplice e altamente scalabile
  - Proprietà di scalabilità «orizzontale» (*scale-out*)
- Consente una più agevole distribuzione del traffico tra i vari *link*
  - ECMP *inter-Leaf*
- Possibilità di utilizzo di switch in configurazione (fisica) fissa
  - Molto meno costosi di switch basati su *chassis* con *slot* modulari
- Sostanziale riduzione di CAPEX e OPEX
- Per contro
  - Limite di scalabilità dovuto agli switch in configurazione fissa
    - Soluzione: aumentare il numero degli stadi

Un po’ di conti ...

- Dati di partenza
  - Fattore di sovraccarico (*oversubscription ratio*): è il rapporto tra la banda totale lato accesso e la banda totale degli *uplink* di un *Leaf*
    - Valore tipico 3:1
    - Nei DC per l’AI 1:1 (*non-blocking*)
  - Numero totale di porte di accesso necessarie: consente di determinare il numero di *Leaf* necessari
  - Fattore di sovraccarico degli *Spine* 1:1 (*non-blocking*)
- Possibilità di utilizzo di *Super-spine*
  - Quando non è possibile soddisfare i requisiti di progetto con switch commerciali
  - Scenario tipico: *Data Center* di grandi dimensioni

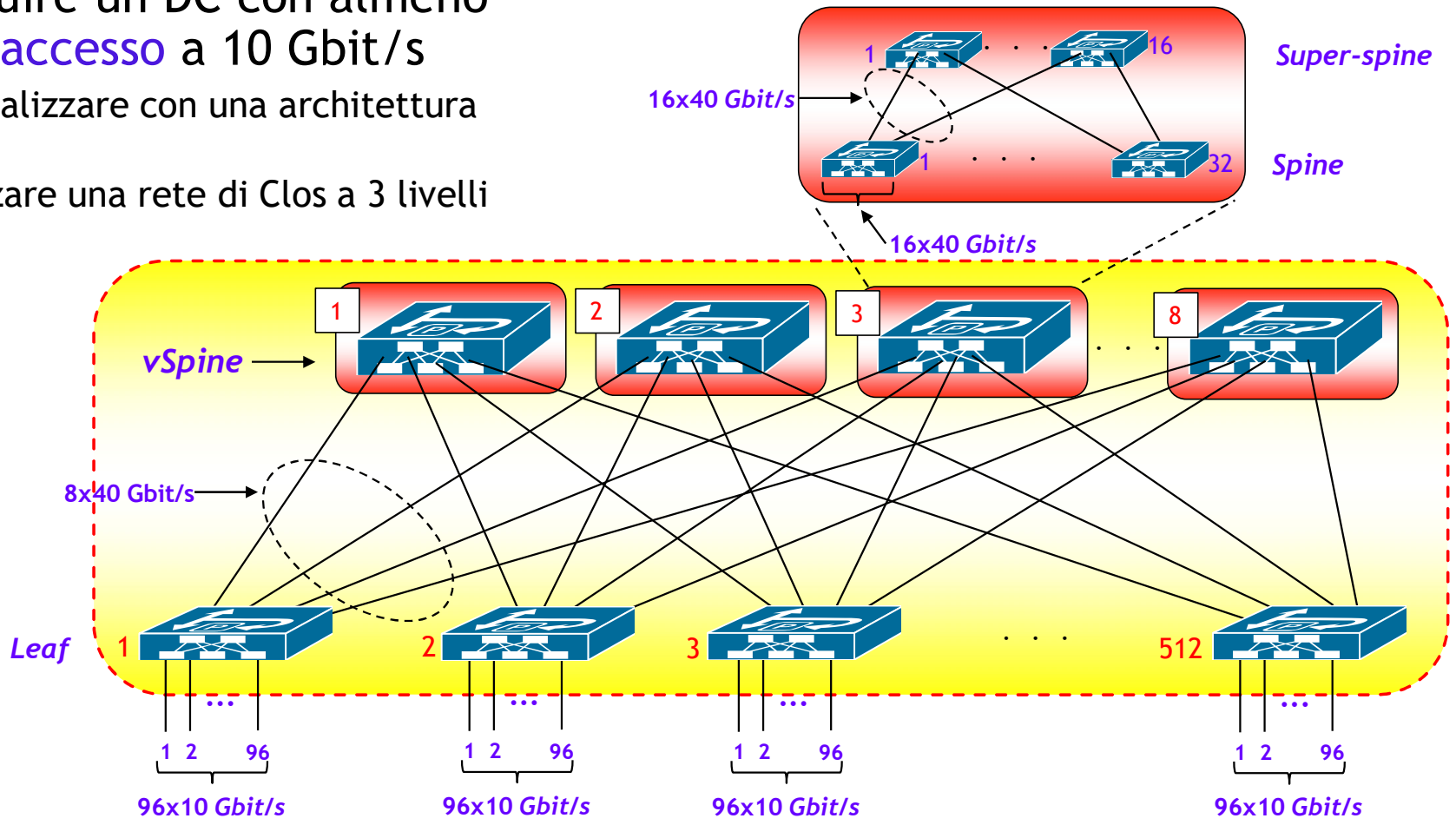
Esempio 1: *Data Center* di medie dimensioni

- Dati di partenza
  - Fattore di sovraccarico = **3:1**
  - Numero totale di porte di accesso necessarie = **700**
  - Disponibilità di switch *Leaf* con **48 porte di accesso a 10 *Gbit/s*** e **4 porte *uplink* a 40 *Gbit/s*** e switch *Spine* con **24 porte a 40 *Gbit/s***



Esempio 2: *Data Center* di grandi dimensioni

- Obiettivo: costruire un DC con almeno 49.000 porte di accesso a 10 Gbit/s
  - Impossibile da realizzare con una architettura a due livelli
  - Soluzione: utilizzare una rete di Clos a 3 livelli (5-*stage*)



vSpine = virtual Spine

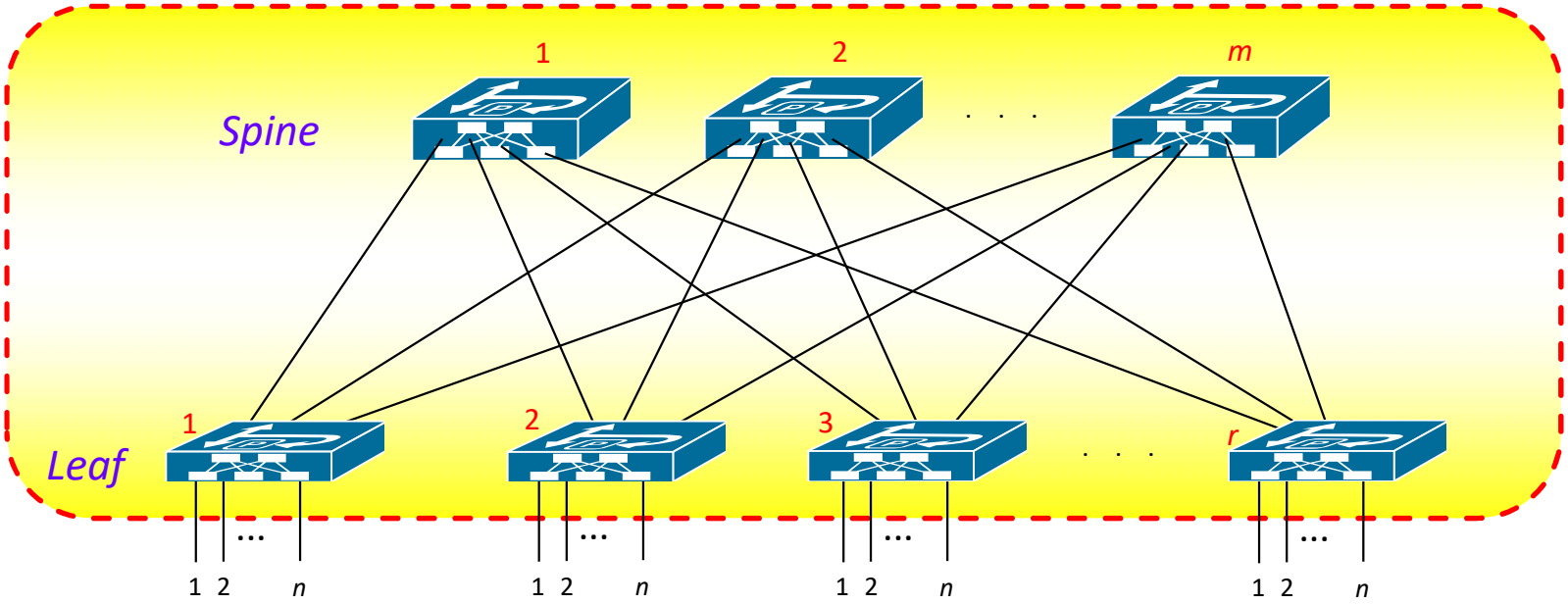
I° modulo: Architetture dei *Data Center*

#1	Dai DC L2 ai DC L3
#2	*Clos IP Fabric*
#3	Aspetti di routing IP

Copyright Tiziano Tofoni

Quale IGP nelle *Clos IP fabric*? (1/2)

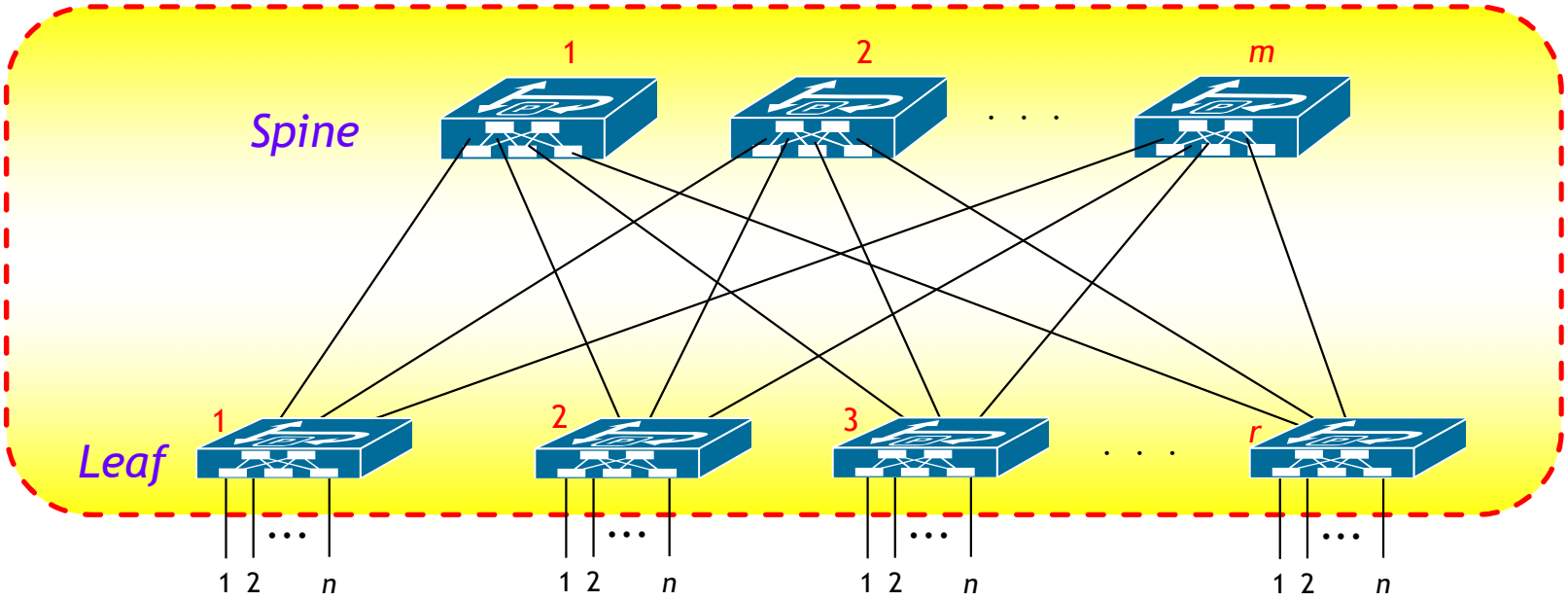
- *Data Center* di dimensioni medio-piccole (es. ≤100 switch): OSPF, IS-IS



Quale IGP nelle *Clos IP fabric*? (2/2)

REISS ROMOLI

- *Data Center* di grandi dimensioni: BGP
  - Semplice, scalabile, maturo, supporto di IPv4 e IPv6
  - (False) Obiezioni: complesso, difficile da capire, configurazione e *troubleshooting* complessi
  - Rif. RFC 7938 - *Use of BGP for Routing in Large-Scale Data Centers*, agosto 2016



Utilizzo del BGP come protocollo IGP: pro e contro

- Pro
  - Non ha bisogno di realizzare adiacenze di routing – le sessioni BGP si basano sul TCP
  - Minore traffico di segnalazione (periodo dei BGP *Keepalive* tipicamente molto maggiore del periodo dei messaggi *Hello*, nessun refresh dei messaggi *Update*)
  - BGP ha una complessità minore: le strutture dati interne e la macchina a stati sono più semplici rispetto alla maggior parte degli IGP *link-state* come OSPF
  - La diffusione delle informazioni (*flooding*) nel BGP ha un ambito di propagazione inferiore rispetto ai protocolli IGP *link-state*.
- Contro
  - Pianificazione dei numeri di AS
  - Può richiedere opportuni comandi per evitare il controllo dell'AS_PATH
  - Fenomeno del *Path Hunting* che potrebbe allungare i tempi di convergenza
  - Interazione con il piano di controllo, se basato su BGP (es. EVPN)

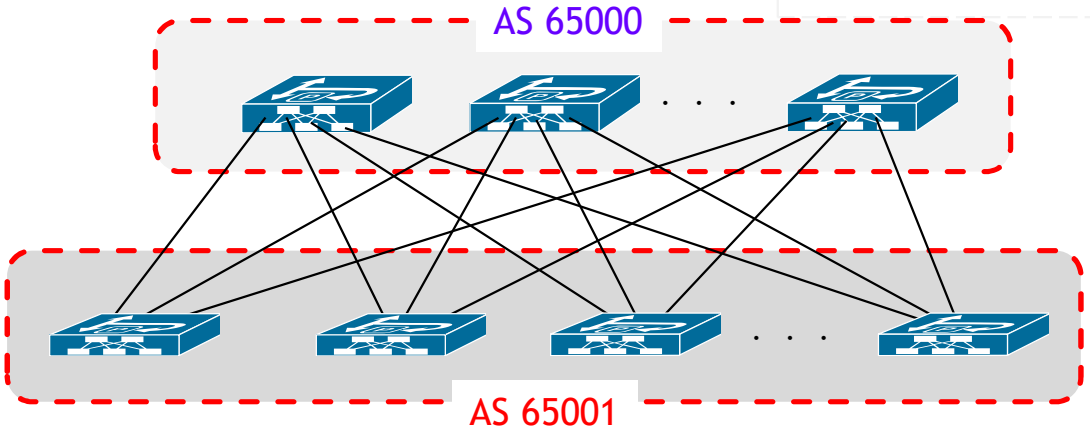
Sessioni eBGP o iBGP?

- L'utilizzo di sessioni iBGP, benché a prima vista più semplice, non è conveniente
  - Obbliga all'utilizzo di *Route Reflector*
  - Impone la presenza di un protocollo IGP
  - La presenza dei RR limita la possibilità di effettuare il *load balancing*

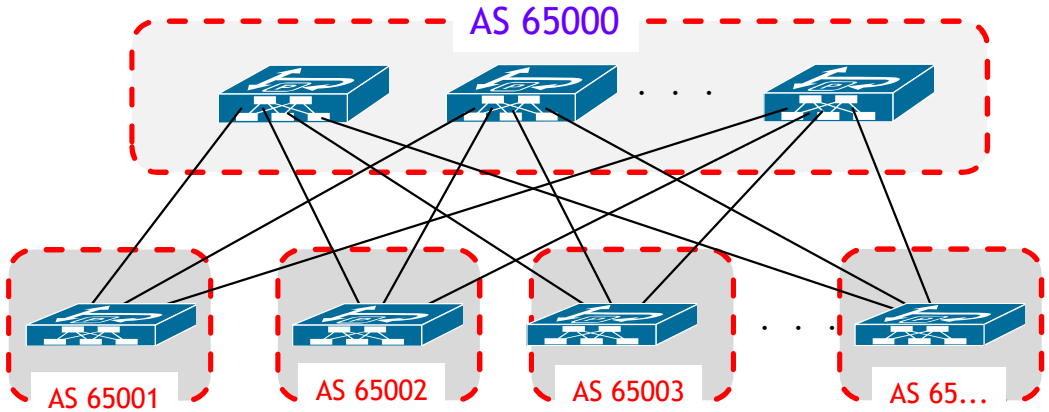
- CONCLUSIONE: è meglio utilizzare sessioni eBGP
  - Richiede una corretta pianificazione dei numeri di AS

Pianificazione dei numeri di AS

- *Dual AS*
  - Pro: semplice
  - Contro: necessario disabilitare il controllo dell'AS-Path, *Path Hunting*, soggetto a *Valley routing*

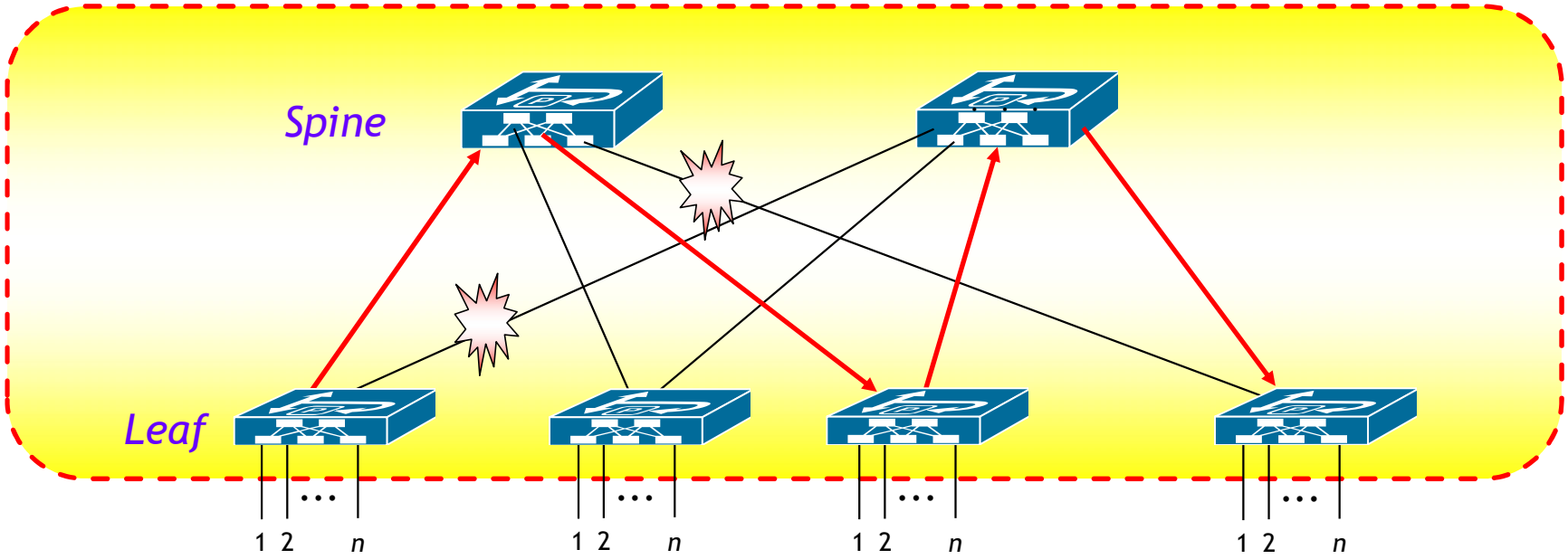


- *Multi AS*
  - Pro: non richiede di disabilitare il controllo dell'AS-Path, no *Path Hunting*, *Valley-free routing*
  - Contro: pianificazione più complessa



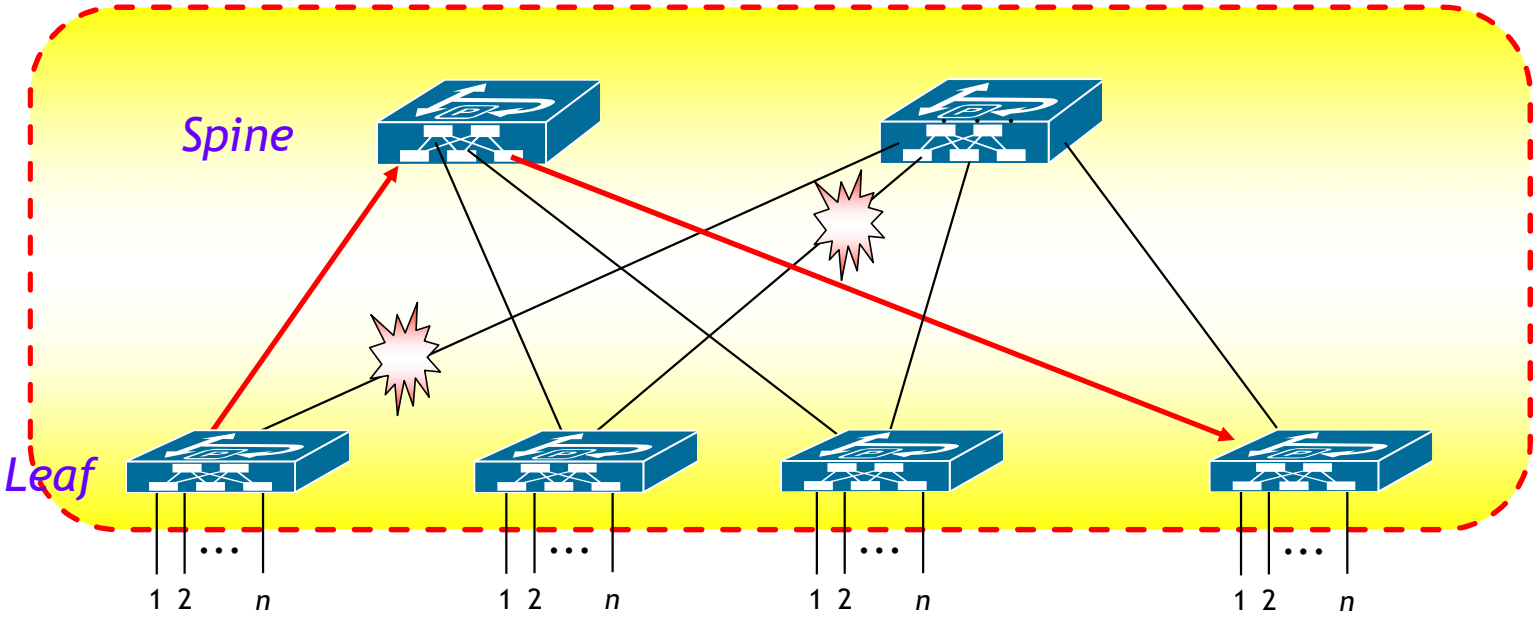
Valley routing

- Con *Valley routing* si indicano percorsi *Leaf-to-Leaf* che attraversano più *Spine*
  - Percorsi più lunghi
  - Latenza, percorsi e carico dei *link* non predicibili



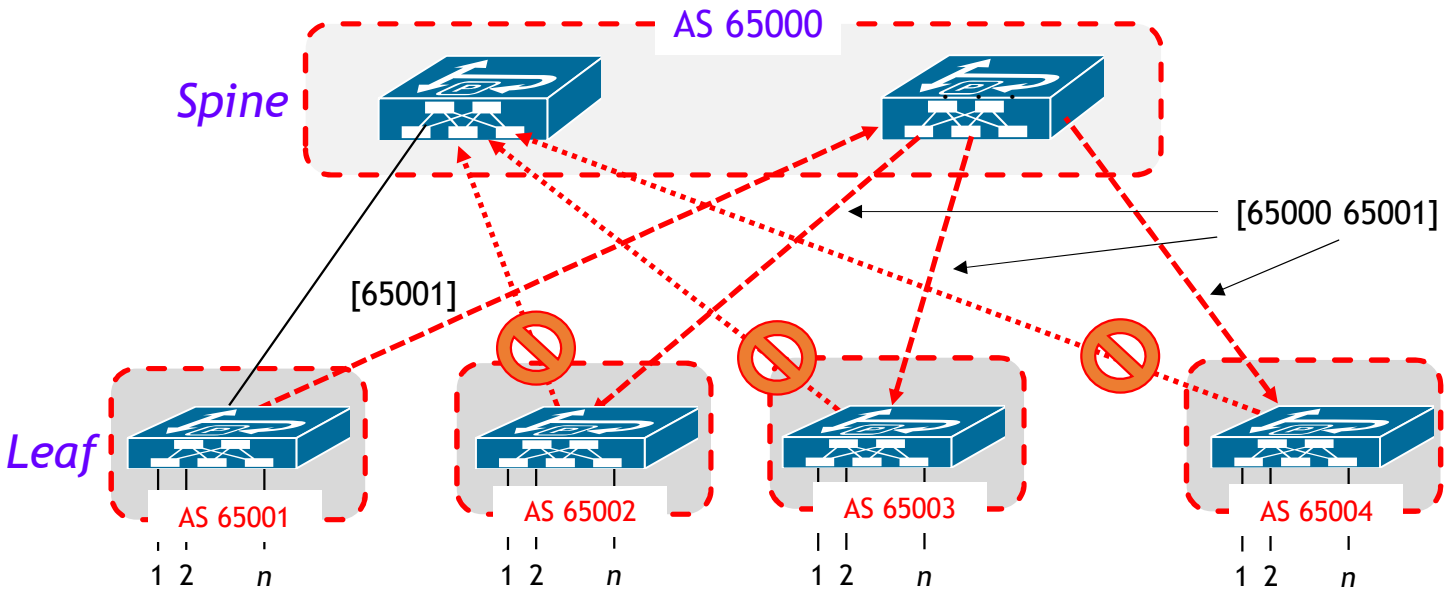
Valley-free routing (1/2)

- Con *Valley-free routing* si indicano percorsi *Leaf-to-Leaf* che attraversano un solo *Spine*
  - Non sempre possibili utilizzando i classici protocolli IGP (OSPF, IS-IS)



Valley-free routing (2/2)

- Si possono ottenere percorsi *Valley-free* utilizzando *BGP Multi AS*
  - Il controllo AS-Path del BGP previene il *Valley routing*: un annuncio viene rifiutato se l'AS_PATH contiene il proprio numero di AS
  - Alternativa (JunOS, IOS XR, NX-OS): un annuncio che contiene l'AS del *neighbor* non viene inviato



Annuncio BGP rete X

II° modulo

Overlay Virtual Networks

Copyright Tiziano Tofoni

II° modulo: *Overlay Virtual Networks*

#1	Dalle VLAN alle reti *Overlay*
#2	Protocolli di *tunneling*: VXLAN, NVGRE, Geneve, STT
#3	*Tunneling* VXLAN
#4	Determinazione delle mappe MAC-to-NVE

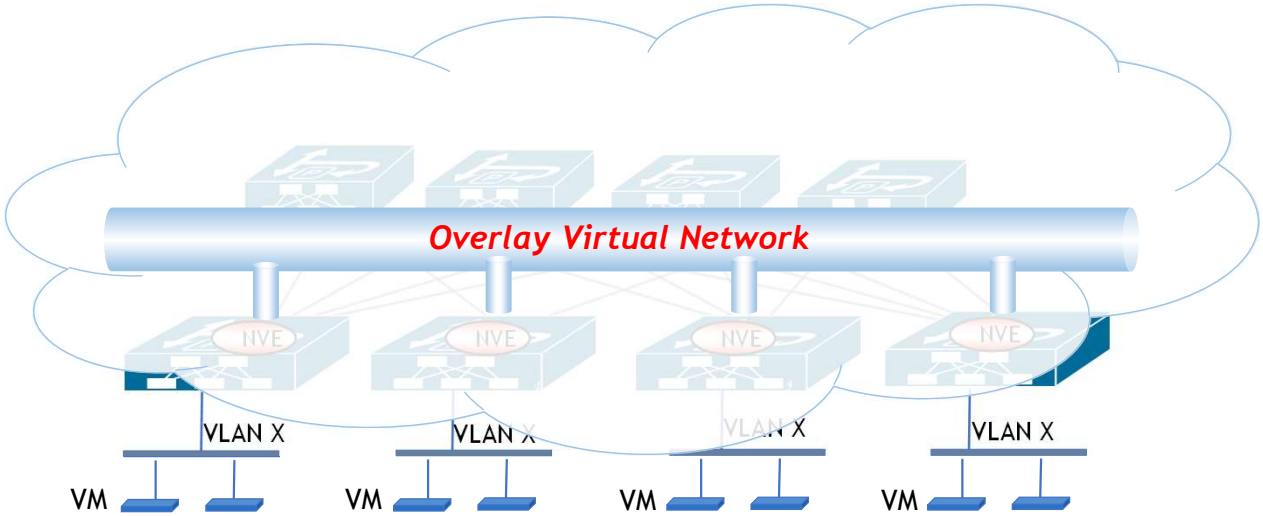
Copyright Tiziano Tofoni

L'idea base …

Disaccoppiare il servizio dal trasporto

- *Overlay Virtual Networks* = *Smart Edge + Dumb Core*
  - La rete *underlay* fornisce il solo semplice trasporto IP
  - Operazioni più complesse svolte dagli apparati di *edge* (che possono anche essere *virtual switch*): vedi RFC 3439
  - Il traffico della rete virtuale è trasportato su IP (MAC-in-IP), in pratica una nuova applicazione che si poggia su IP!

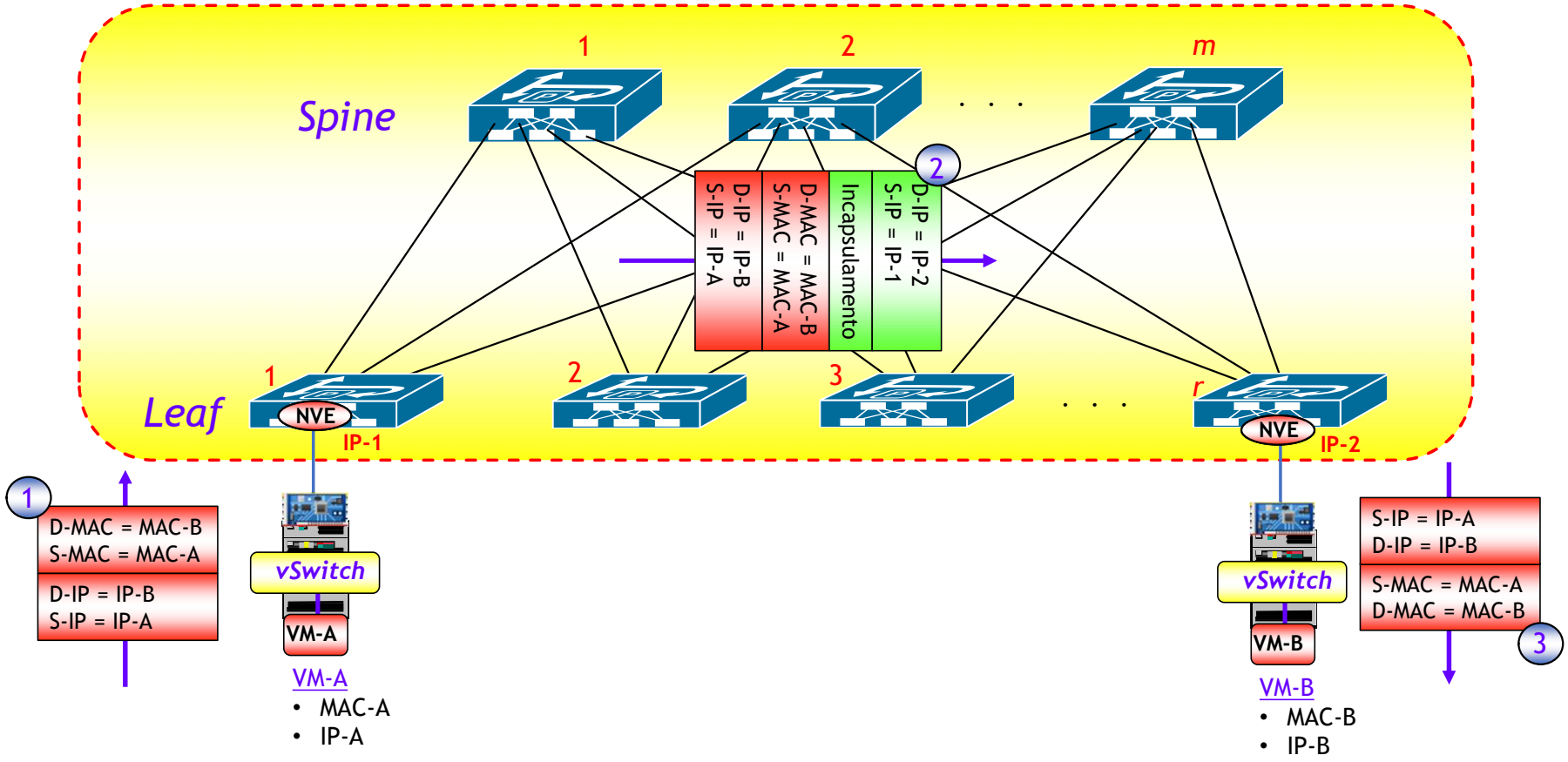
La visione logica …



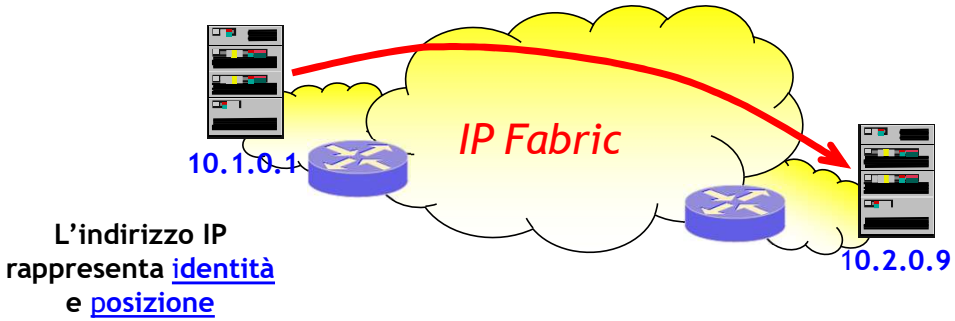
NVE = Network Virtualization Endpoint

- Rete *underlay* = *IP Fabric*
  - Tipicamente una rete di Clos
- Rete *overlay* = Segmento LAN esteso
  - Estendibilità e mobilità

MAC-in-IP...

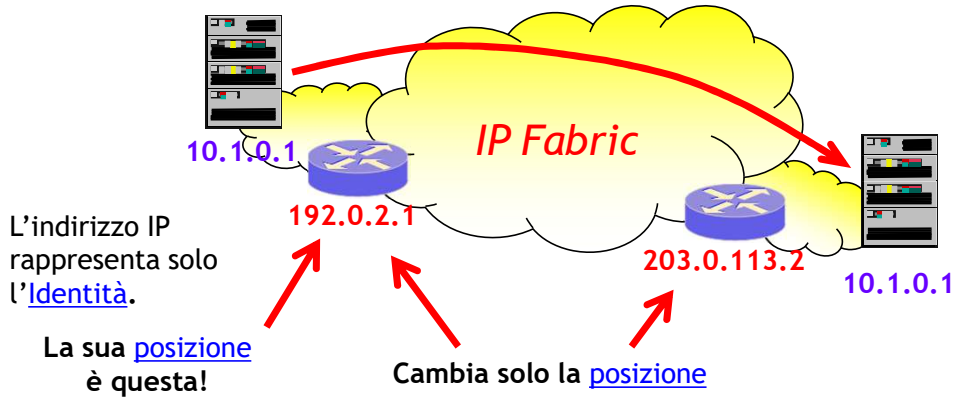

Spine
1
2
m
Leaf
1
2
3
r
NVE
IP-1
NVE
IP-2
2
D-IP = IP-B
S-IP = IP-A
D-MAC = MAC-B
S-MAC = MAC-A
Incapsulamento
D-IP = IP-2
S-IP = IP-1
1
D-MAC = MAC-B
S-MAC = MAC-A
D-IP = IP-B
S-IP = IP-A
vSwitch
VM-A
VM-A
• MAC-A
• IP-A
vSwitch
VM-B
VM-B
• MAC-B
• IP-B
S-IP = IP-A
D-IP = IP-B
S-MAC = MAC-A
D-MAC = MAC-B
3

Separazione di identità e posizione



Comportamento "classico"

Quando l'host si muove ottiene un nuovo indirizzo IP per la sua nuova identità e posizione



Comportamento "*overlay*"

Quando l'host si muove mantiene il suo indirizzo IP. L'identità non cambia!

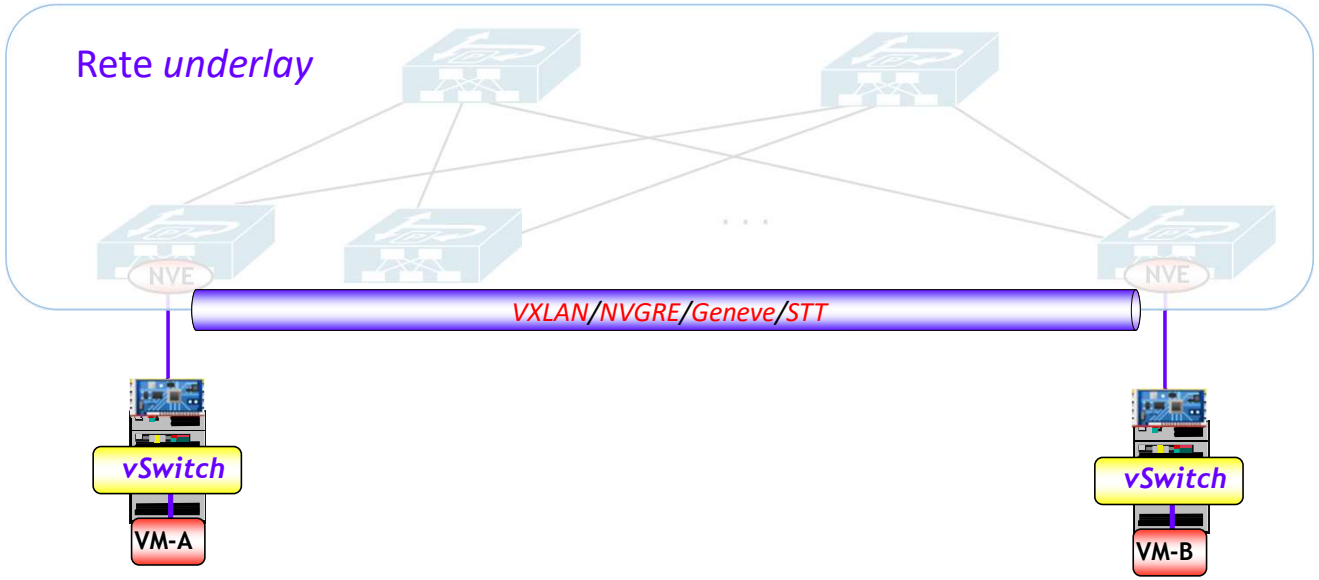
II° modulo: *Overlay Virtual Networks*

#1	Dalle VLAN alle reti *Overlay*
#2	Protocolli di *tunneling*: VXLAN, NVGRE, Geneve, STT
#3	*Tunneling* VXLAN
#4	Determinazione delle mappe MAC-to-NVE

Copyright Tiziano Tofoni

Tunneling (IP) delle trame Ethernet

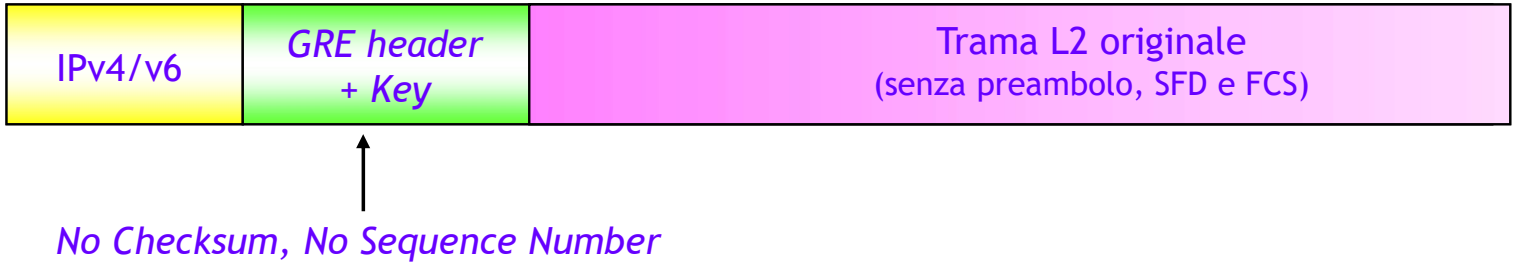
- Le trame Ethernet generate dagli *host* sono trasportate in *tunnel IP*
  - *End-point*: indirizzi IP degli switch *Leaf* dove sono attestati gli *host* sorgente e destinazione
  - Informazione minimale: un equivalente (possibilmente esteso) del *VLAN tag*



La (stupida) guerra degli incapsulamenti (1/2)

- *Virtual eXtensible LAN* (VXLAN)
  - Protocollo standard (RFC 7348)
  - *Supporter*: Cisco, Juniper (Contrail), WMware, Red Hat, Citrix, Arista, ecc.

- *Network Virtualization using GRE* (NVGRE)
  - Protocollo standard (RFC 7637)
  - *Supporter*: Microsoft, Intel, HP, Dell, Arista Networks



La (stupida) guerra degli incapsulamenti (2/2)

- *GE*neric *NE*twork *V*irtualization *E*ncapsulation (GENEVE)
  - Protocollo standard (*RFC* 8926)
  - *Supporter*: VMware, Microsoft, Red Hat, Intel

- *S*tateless *T*ransport *T*unneling (STT)
  - Protocollo ancora non standard (*draft-davie-stt*)
  - *Supporter*: Nicira (successivamente acquisita da VMware)

Il vero problema ...

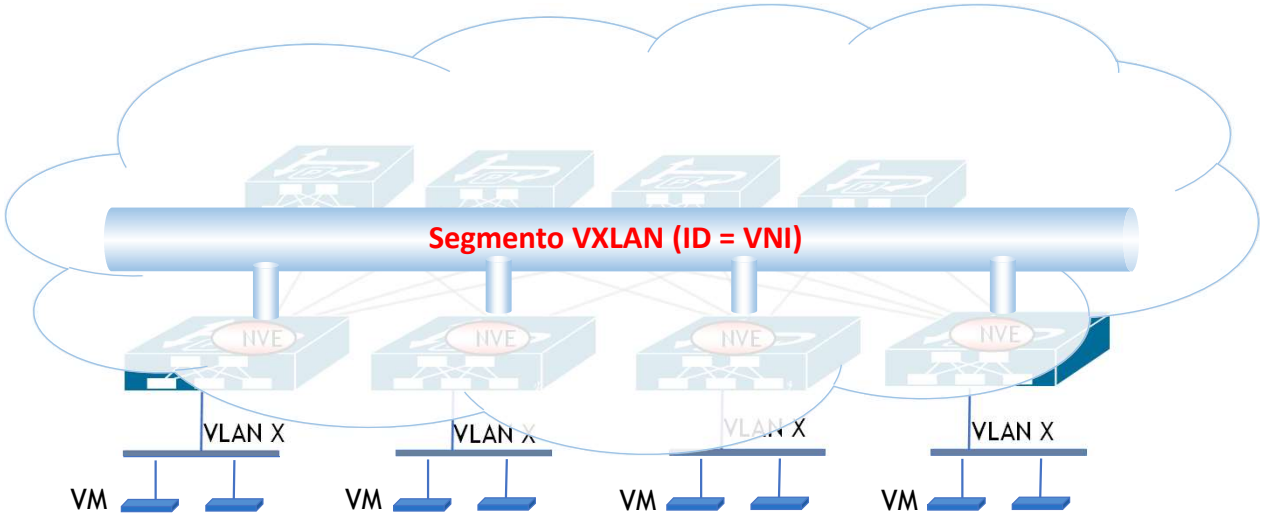
- Non è il tipo di incapsulamento ... ma la determinazione delle mappe MAC-to-NVE
  - In altre parole, come fa la NVE sorgente a determinare l'indirizzo IP dell'interfaccia NVE destinazione
- Come determinare la mappa MAC-to-NVE
  - *Routing multicast*
  - *Unicast-only*
  - Piano di controllo
- NOTA: in ogni caso, il *MAC learning* locale avviene in modo classico, ossia sul piano dati

II° modulo: *Overlay Virtual Networks*

#1	Dalle VLAN alle reti *Overlay*
#2	Protocolli di *tunneling*: VXLAN, NVGRE, Geneve, STT
#3	*Tunneling* VXLAN
#4	Determinazione delle mappe MAC-to-NVE

Copyright Tiziano Tofoni

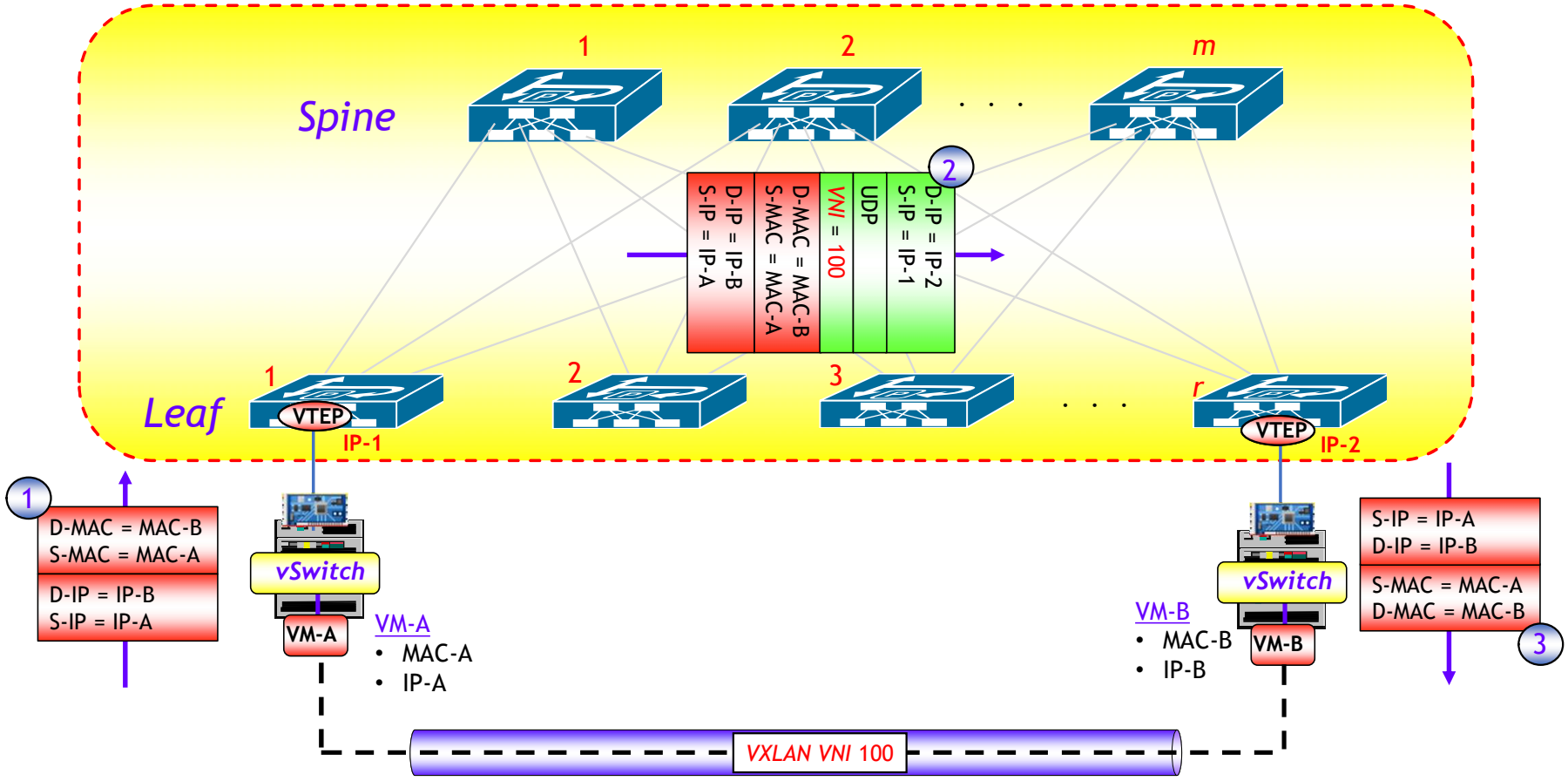
La visione logica …



NVE = *Network Virtualization Endpoint*

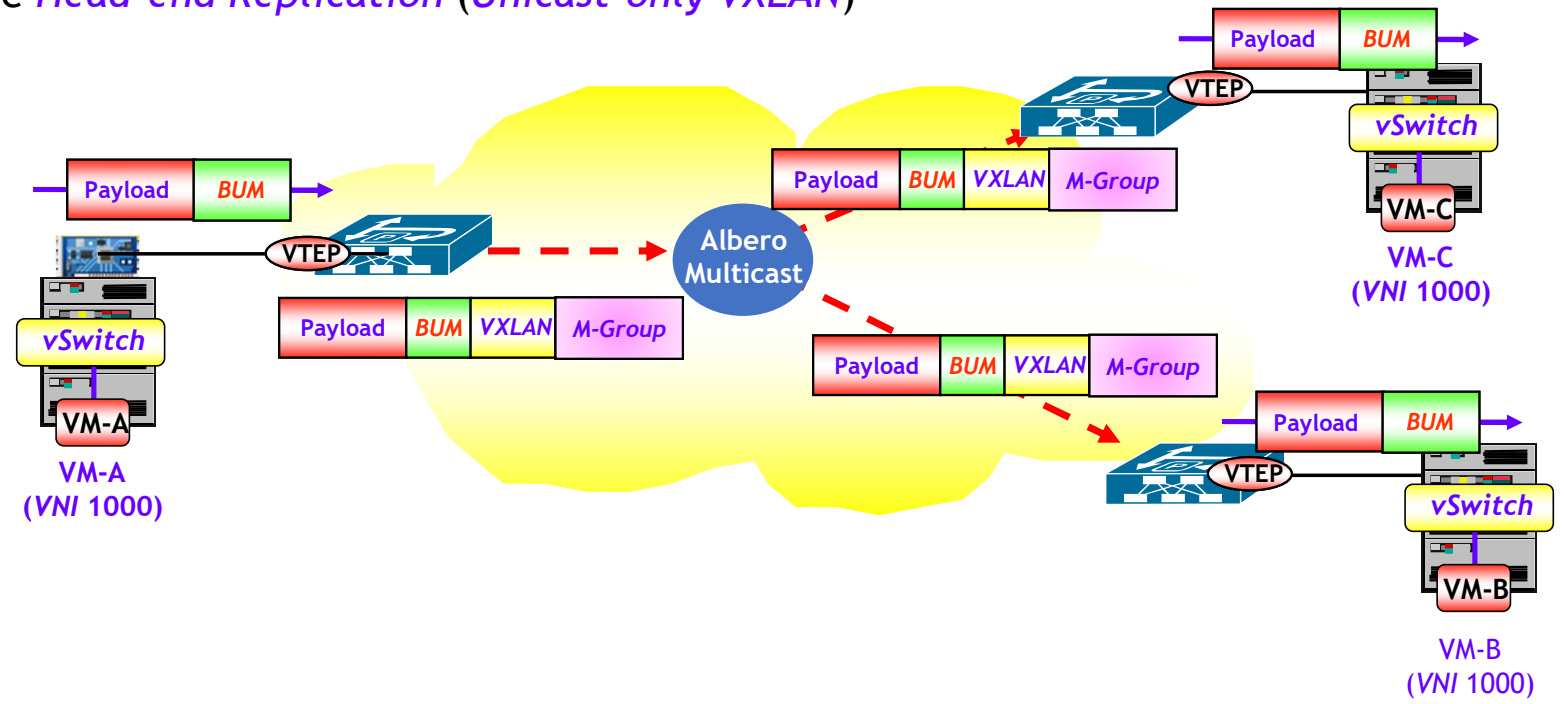
- Rete *underlay* = *IP Fabric*
  - Tipicamente una rete di Clos
- Rete *overlay* = Segmento LAN esteso
  - VNI = *VXLAN Network Identifier*
  - VTEP = *VXLAN Tunnel End Point* (equivalente di NVE)

Trasporto delle trame L2 *unicast*

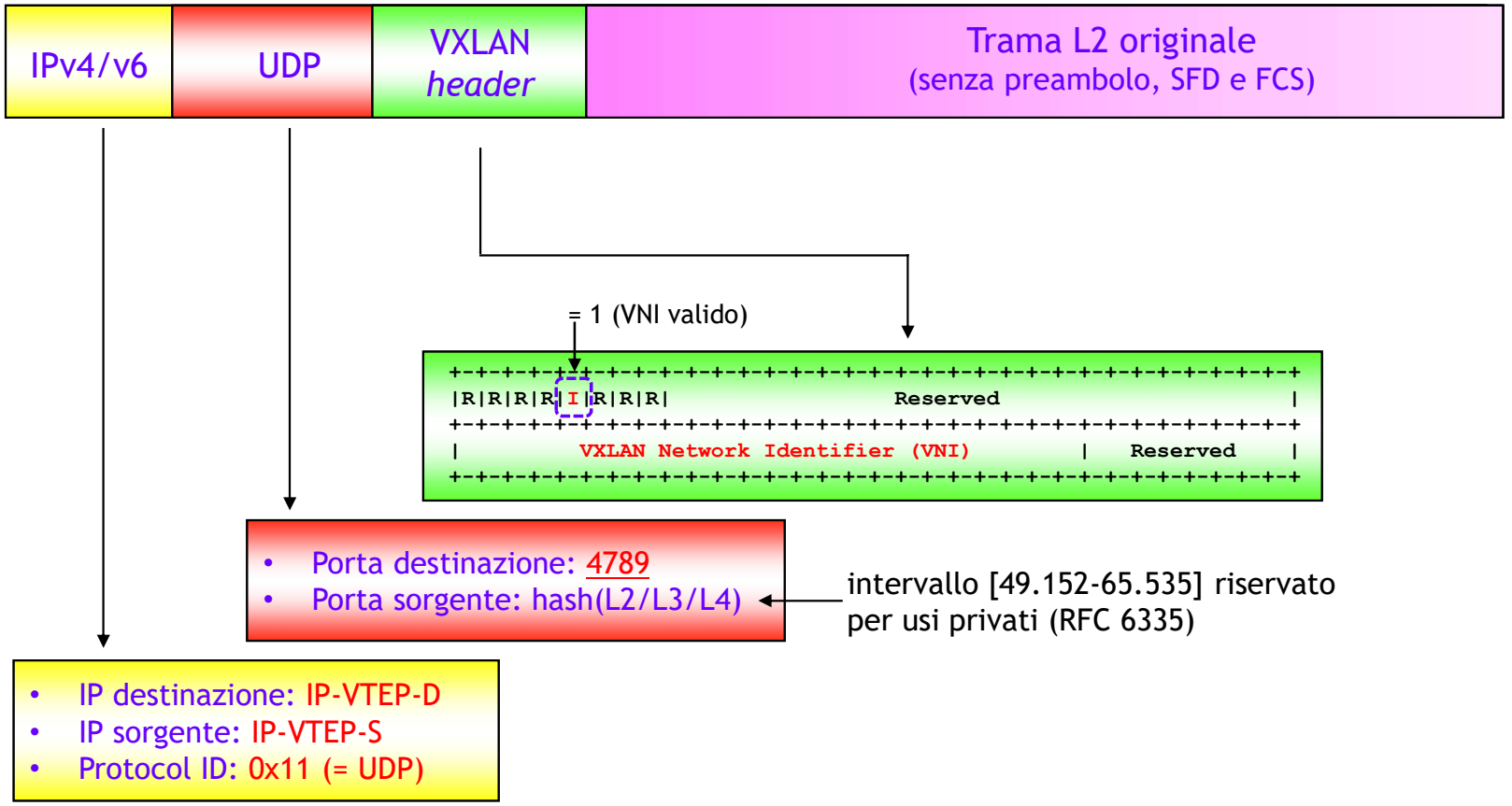


Trasporto delle trame L2 BUM

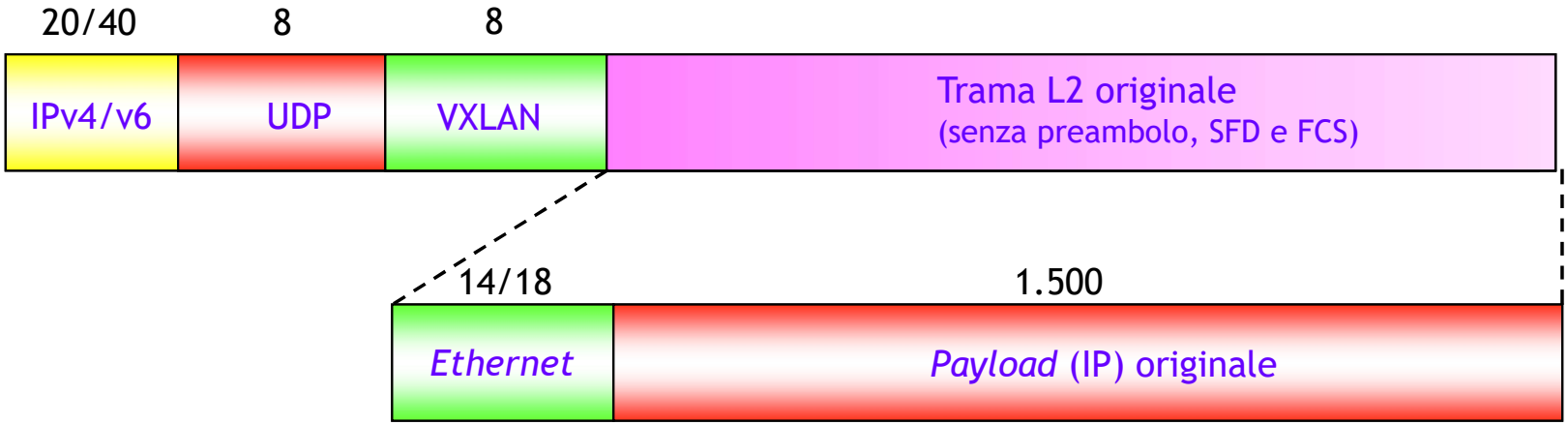
- Due alternative possibili
  - Utilizzare un albero *multicast* interno alla *IP Fabric*
    - Tipicamente realizzato con PIM BiDir
  - Utilizzare *Head-end Replication* (*Unicast-only VXLAN*)



Formato dei pacchetti VXLAN



Pacchetti VXLAN e MTU



- MTU IPv4 = 20 (IP) + 8 (UDP) + 8 (VXLAN) + Trama L2 originale
  - Se la trama L2 originale è Ethernet v2 → MTU = 1.550
  - Se la trama L2 originale è IEEE 802.1Q → MTU = 1.554
- MTU IPv6 = MTU IPv4 + 20 (= 1.570/1.574)

II° modulo: *Overlay Virtual Networks*

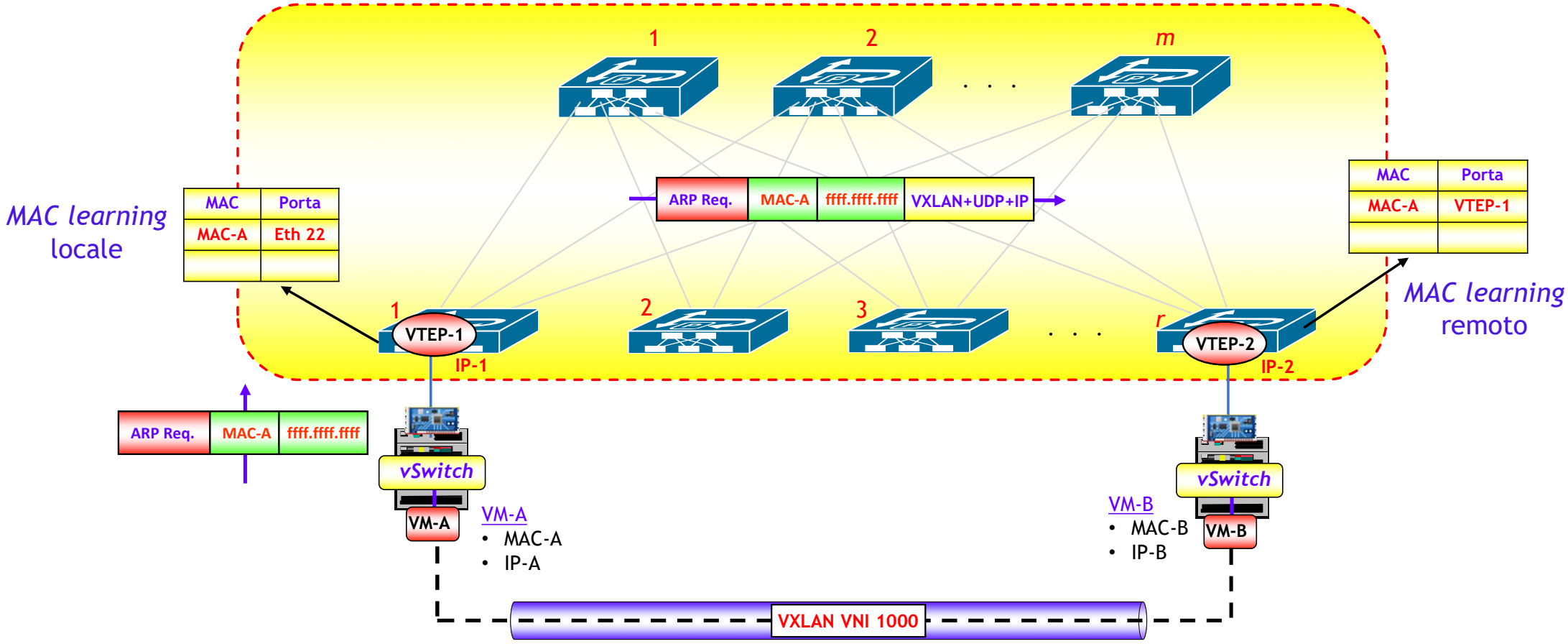
#1	Dalle VLAN alle reti *Overlay*
#2	Protocolli di *tunneling*: VXLAN, NVGRE, Geneve, STT
#3	*Tunneling* VXLAN
#4	Determinazione delle mappe MAC-to-NVE

Copyright Tiziano Tofoni

Le fasi di evoluzione

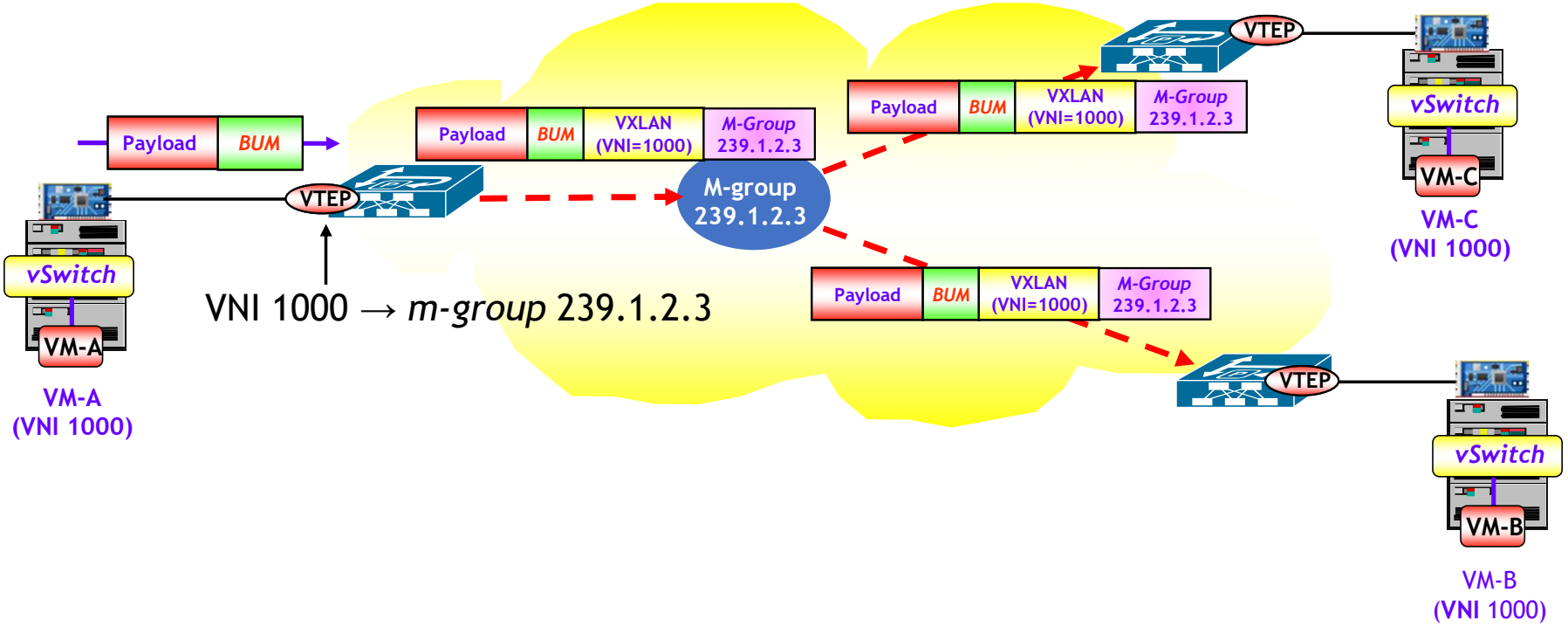
- Fase 1: nessun piano di controllo: *MAC learning* interamente sul piano dati con utilizzo di un protocollo di *routing multicast* sulla rete *underlay*
  - Problema principale: molte *IP fabric* non utilizzano *routing multicast*
- Fase 2: nessun piano di controllo: *MAC learning* interamente sul piano dati senza utilizzo di un protocollo di *routing multicast* sulla rete *underlay*
  - *Head-end Replication* del traffico BUM
  - Necessità di meccanismo per l'*auto-discovery* delle VTEP dove sono attestati *host* dello stesso segmento VXLAN
  - Problema principale: scarsa scalabilità
- Fase 3: utilizzo di piano di controllo per annunciare gli indirizzi MAC degli *host* (e se possibile anche gli indirizzi IP associati)
  - Piano di controllo utilizzato: EVPN (vedi prossimo modulo)
  - Problema principale: complessità di configurazione e di *troubleshooting*

MAC Learning sul piano dati



VXLAN fase 1: utilizzo del routing multicast

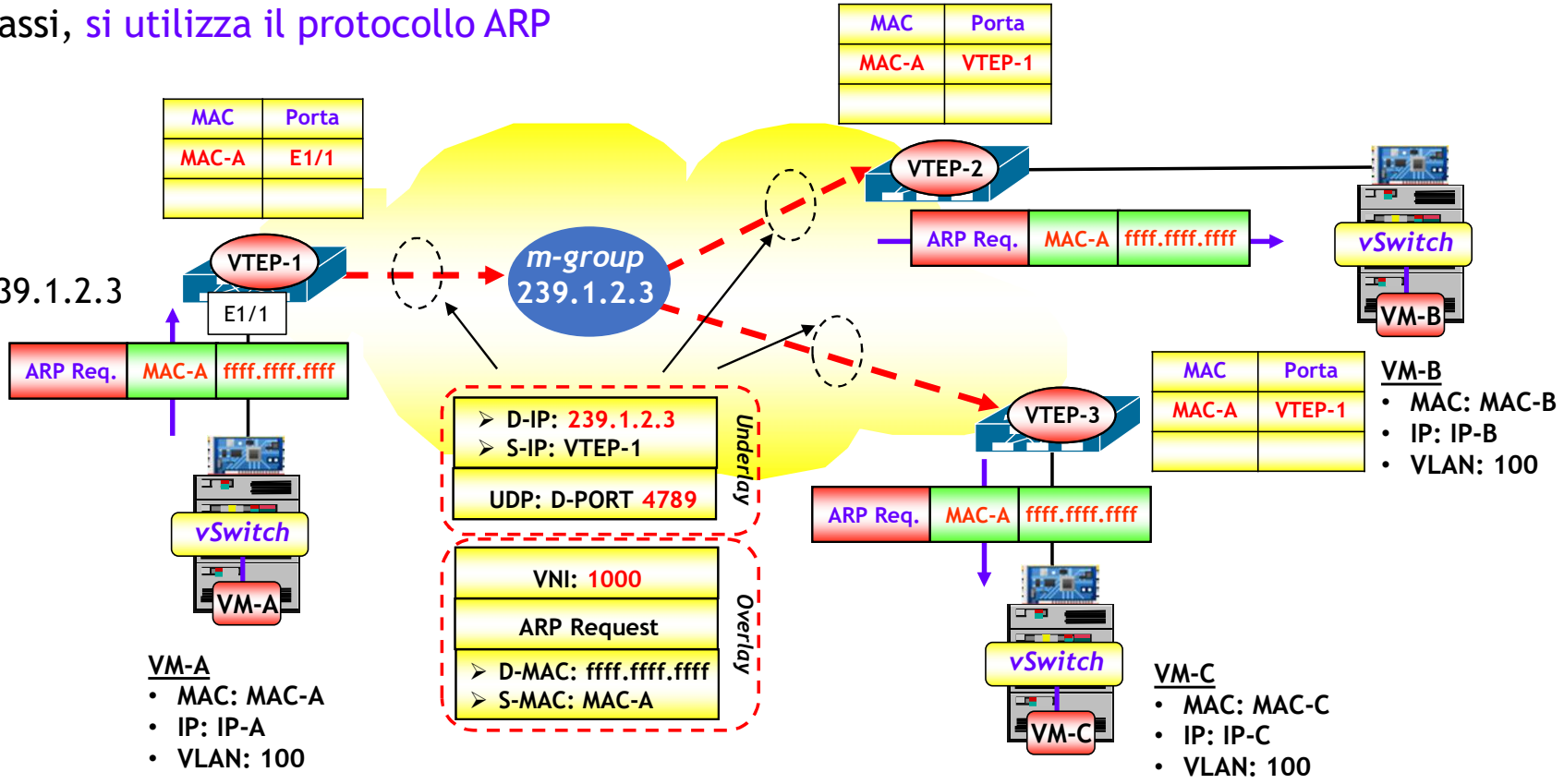
- Necessità di definire un gruppo multicast per uno o più segmenti VXLAN



VXLAN fase 1: esempio (1/2)

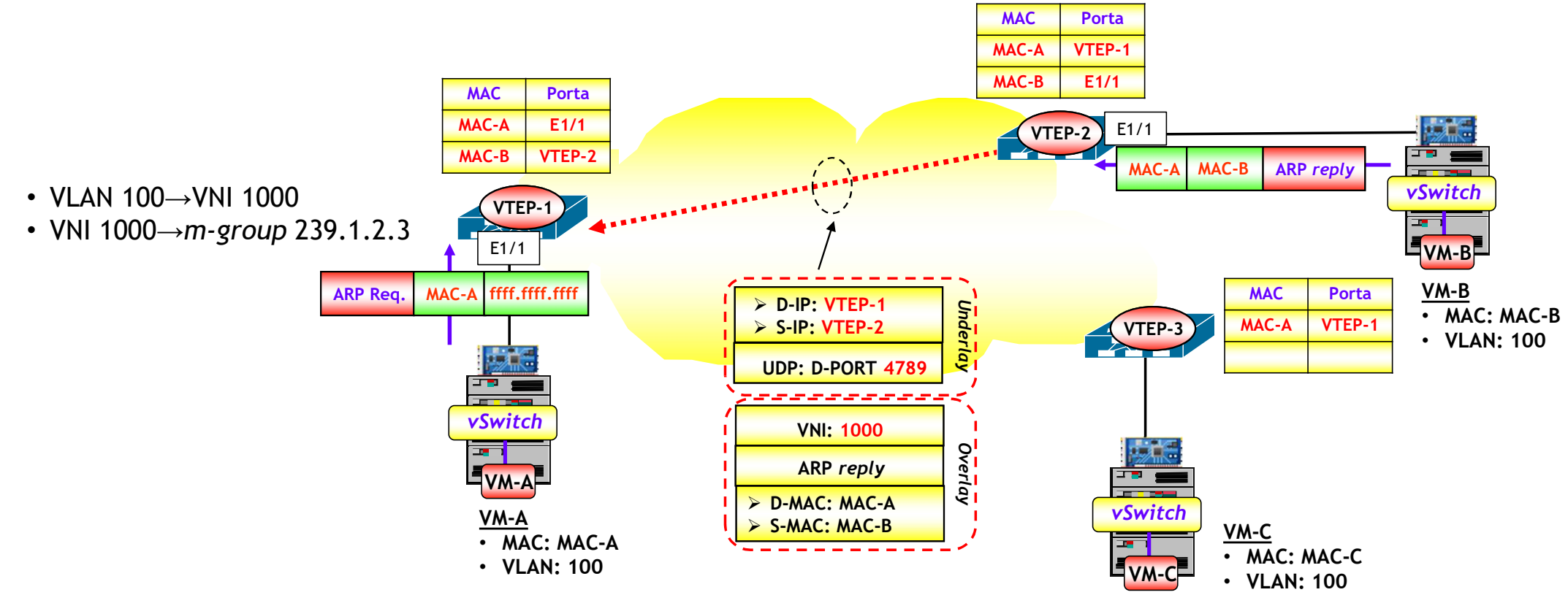
- L'host A deve inviare un pacchetto IP all'*host* B e quindi ha bisogno dell'indirizzo MAC dell'*host* B
  - Come da prassi, si utilizza il protocollo ARP

- VLAN 100→VNI 1000
- VNI 1000→*m-group* 239.1.2.3



VXLAN fase 1: esempio (2/2)

- L'host B risponde con un messaggio *ARP reply*
  - Il messaggio *ARP reply* viene inviato direttamente all'indirizzo MAC unicast dell'host A



- VLAN 100→VNI 1000
- VNI 1000→*m-group* 239.1.2.3

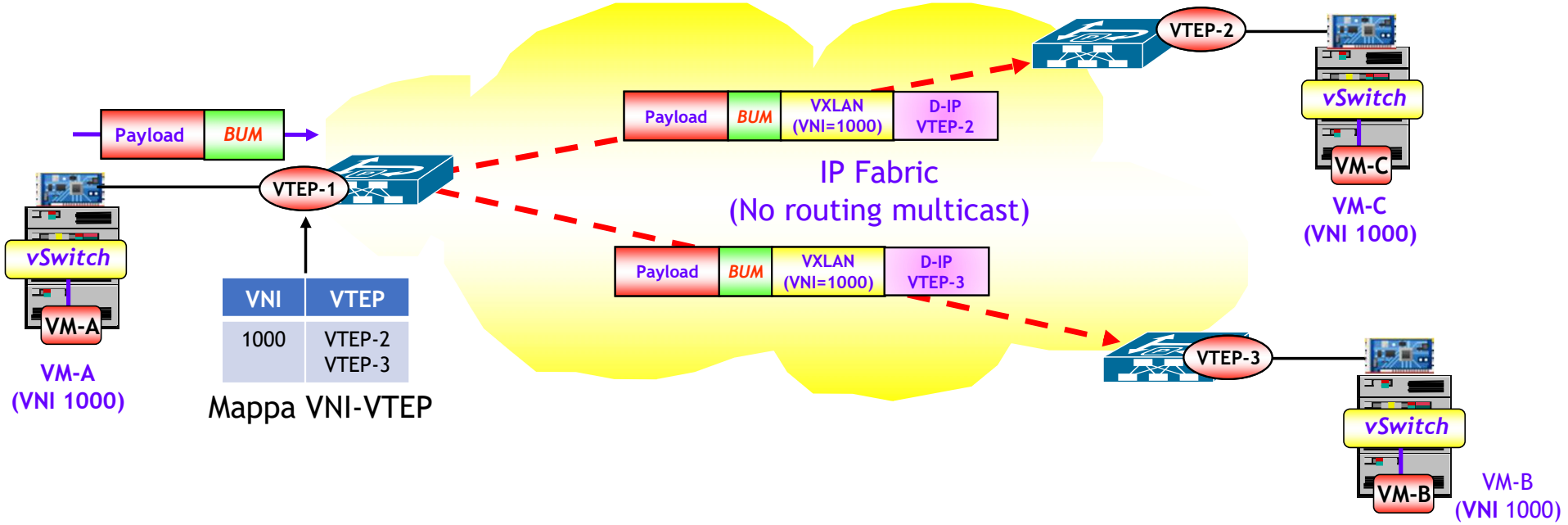
VM-A
- MAC: MAC-A
- VLAN: 100

VM-B
- MAC: MAC-B
- VLAN: 100

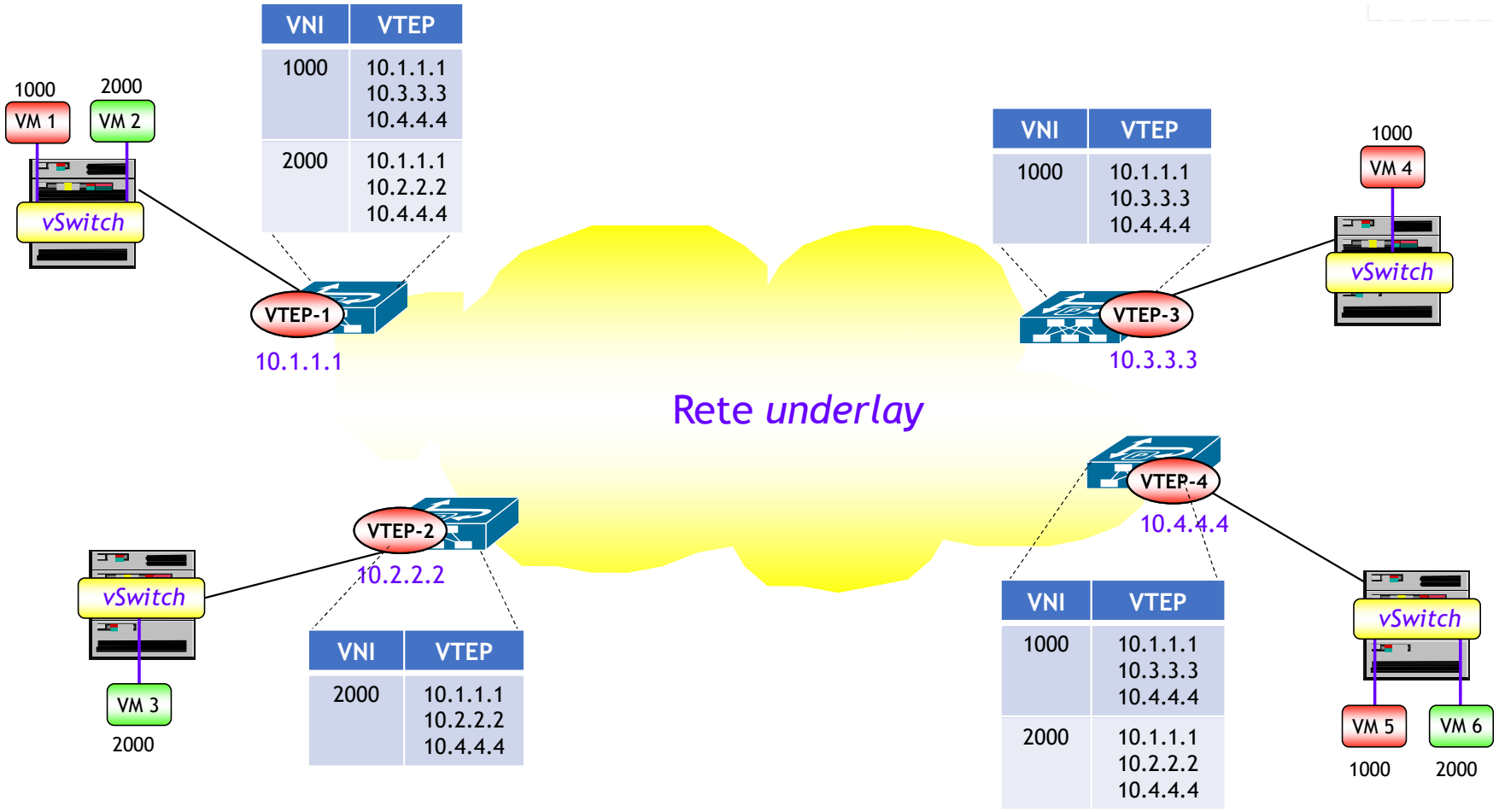
VM-C
- MAC: MAC-C
- VLAN: 100

VXLAN fase 2 (*unicast-only VXLAN*)

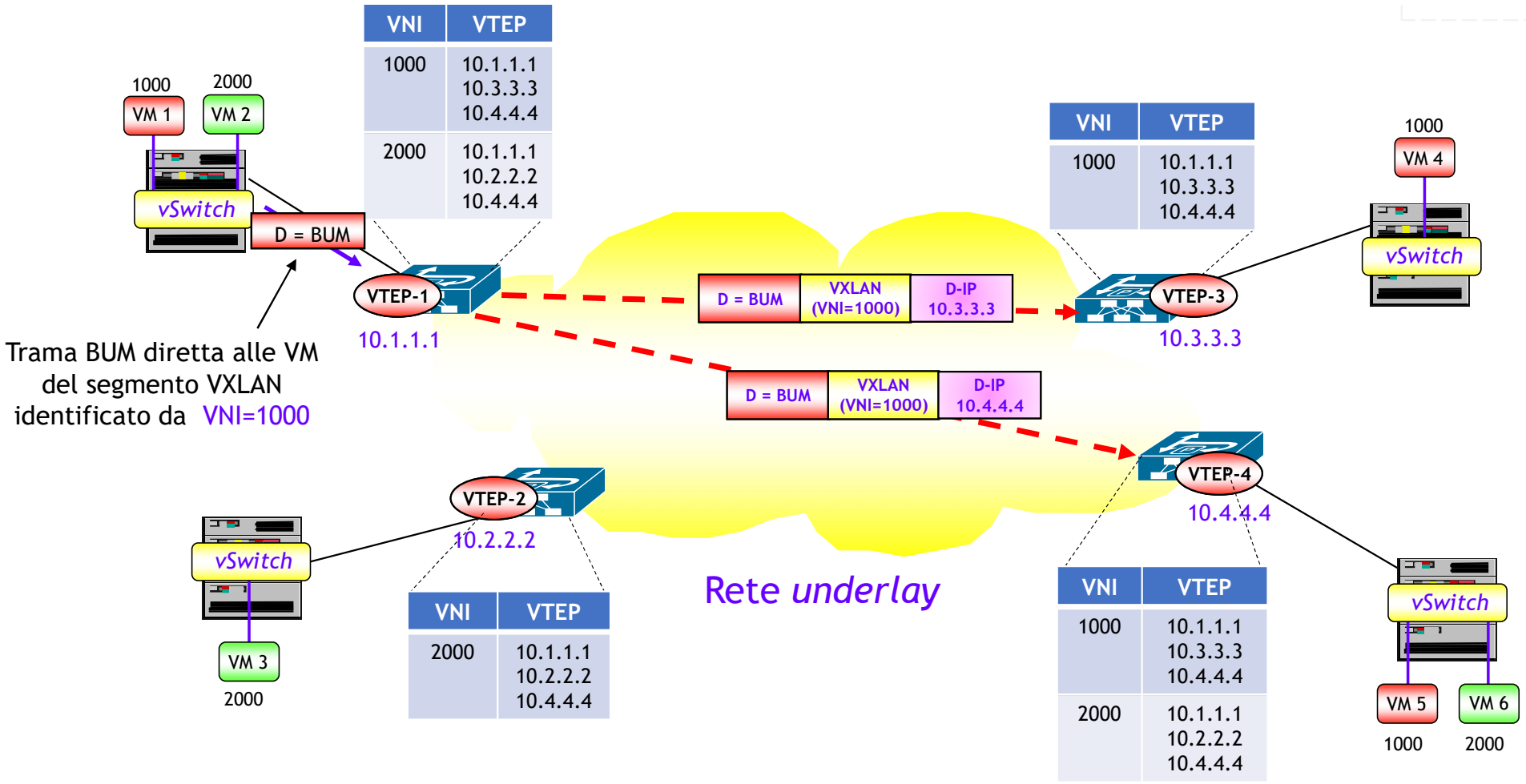
- I pacchetti VXLAN che trasportano traffico BUM vengono replicati in ingresso
  - Una copia per ciascuna VTEP destinazione
  - Problema: come fa lo switch *Leaf* di ingresso a conoscere a quali VTEP replicare il pacchetto VXLAN?
  - Soluzione: creare una mappa VNI-VTEP (staticamente o dinamicamente)



Unicast-only VXLAN: mappe VNI-VTEP (1/2)



Unicast-only VXLAN: mappe VNI-VTEP (2/2)



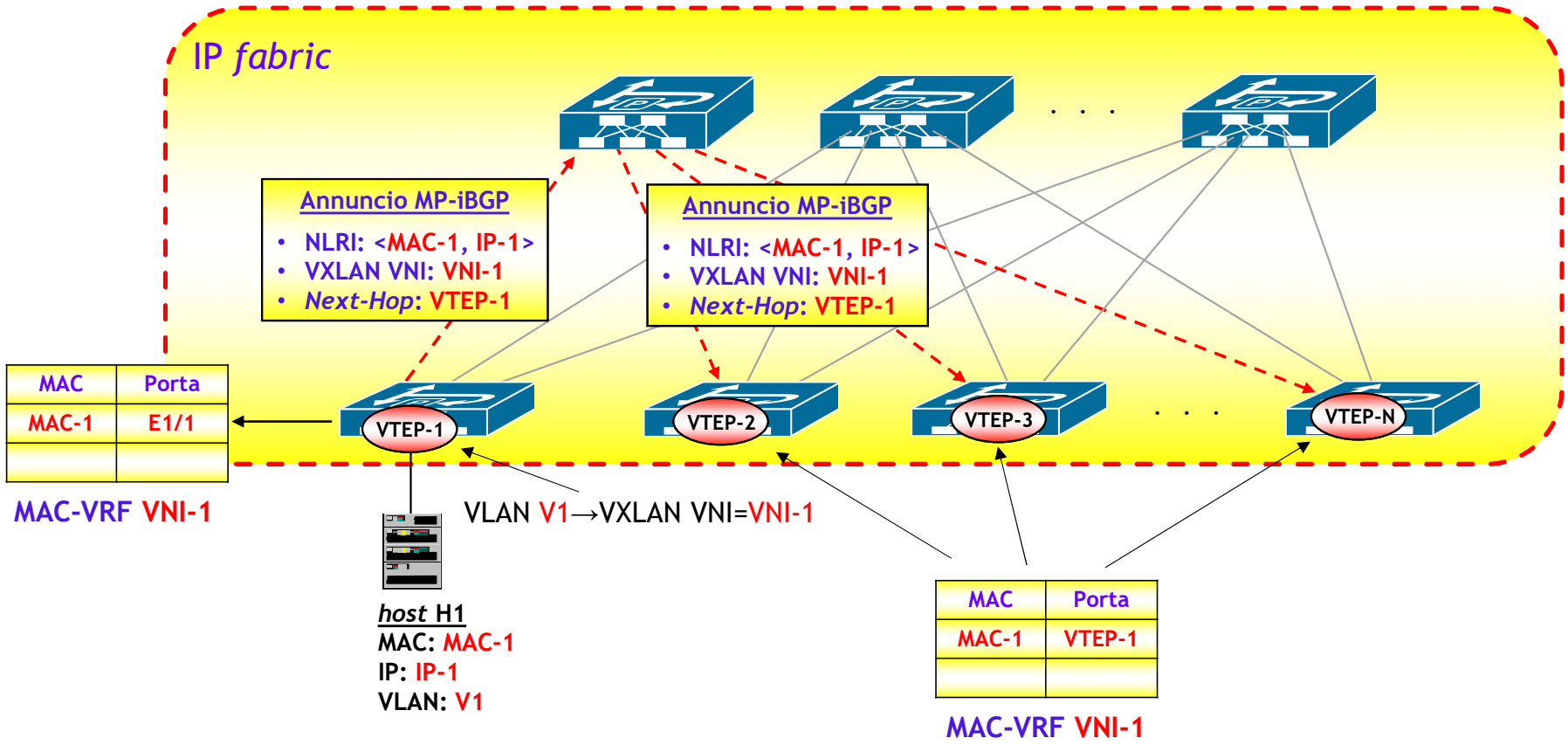
VXLAN fase 3: piano di controllo EVPN (1/2)

REISS ROMOLI

- Idea principale: netta separazione tra piano di controllo e piano dati
  - Piano di controllo: utilizza il BGP (multiprotocollo)
  - Piano dati: incapsulamento VXLAN
- Vantaggi
  - Forte riduzione del traffico BUM
  - Gestione efficiente della mobilità degli *host* (VM)
  - Meccanismo di *autodiscovery* basato sul BGP
  - Integrazione L2/L3 nativa
  - Possibilità di utilizzare la funzionalità *Distributed IP Anycast Gateway*
  - Possibilità di utilizzare tutte le funzionalità avanzate del modello EVPN
  - Possibilità di creare topologie logiche arbitrarie

Copyright Tiziano Tofoni

VXLAN fase 3: piano di controllo EVPN (2/2)



III° modulo

Ethernet VPN (EVPN)

Copyright Tiziano Tofoni

III° modulo: Ethernet VPN (EVPN)

#1	EVPN come evoluzione dei servizi L2VPN
#2	Piano di controllo
#3	Funzionalità *multi-homing*
#4	*ARP/ND suppression*
#5	*MAC mobility*

Copyright Tiziano Tofoni

Caratteristiche principali (1/2)

- I servizio L2VPN basati sull'architettura BGP/MPLS hanno vari difetti che ne impediscono l'utilizzo in scenari avanzati (es. *DCI*, *Data Center Interconnection*)
  - Vedi prossima pagina
- Modello EVPN
  - Obiettivo: rendere i servizi L2VPN indipendenti dal piano dati e più adatti a rispondere alle esigenze del DCI
  - Modello standard: RFC 7432 "*BGP MPLS-Based Ethernet VPN*", Febbraio 2015
  - Stesso modello di funzionamento del servizio L3VPN
    - Utilizzo di MP-BGP sul piano di controllo
    - Supporto di vari piani dati (MPLS, MPLSoGRE, VXLAN, NVGRE, MAC-in-MAC)
  - Nuove funzionalità per la gestione ottimale dei CE *multi-homed*

Limitazioni dei servizi L2VPN «classici»

- Servizi L2VPN punto-punto
  - Scarsa scalabilità (basati su sessioni LDP *multi-hop*, utilizzo del BGP non standard)
- Servizi L2VPN punto-punto: tre diverse opzioni di implementazione, ciascuna con i suoi pro e contro
  - Segnalazione LDP e *discovery* manuale (RFC 4762).
  - Segnalazione LDP e *auto-discovery* via BGP (citata come possibile soluzione nella RFC 4762).
  - Segnalazione e *auto-discovery* via BGP (RFC 4761)
- Altre limitazioni
  - Funzionalità limitate per la gestione dei CE *multi-homed*
  - Scarsa scalabilità
  - Trasporto del traffico *BUM* non ottimale
  - Nessuna funzionalità per la gestione della mobilità

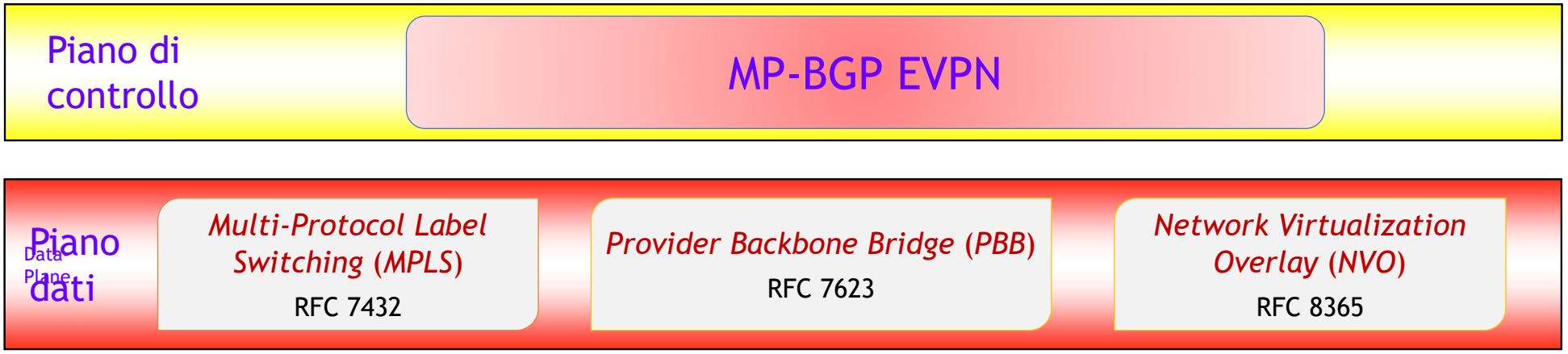
Copyright Tiziano Tofoni

I have a dream ...

- *Multi-homing* con modalità di *forwarding* *active-active* e *load balancing* da/verso i CE
- Trasporto ottimo del traffico BUM
- Scalabilità mediante l'utilizzo della funzionalità di *Route Reflection*
- Convergenza veloce
- Riduzione, del meccanismo del *flooding*
- Gestione della *MAC mobility*

Un piano di controllo per tutte le stagioni...

- Il piano di controllo EVPN può essere utilizzato con diversi piani dati
  - RFC 7432: piano dati MPLS
  - RFC 7623: piano dati *MAC-in-MAC*, anche noto come PBB (*Provider Backbone Bridge*), definito dallo standard IEEE 802.1ah.
  - RFC 8365: piano dati generico utilizzato nelle *Overlay Virtual Network* (es. VXLAN, NVGRE, ...)



III° modulo: Ethernet VPN (EVPN)

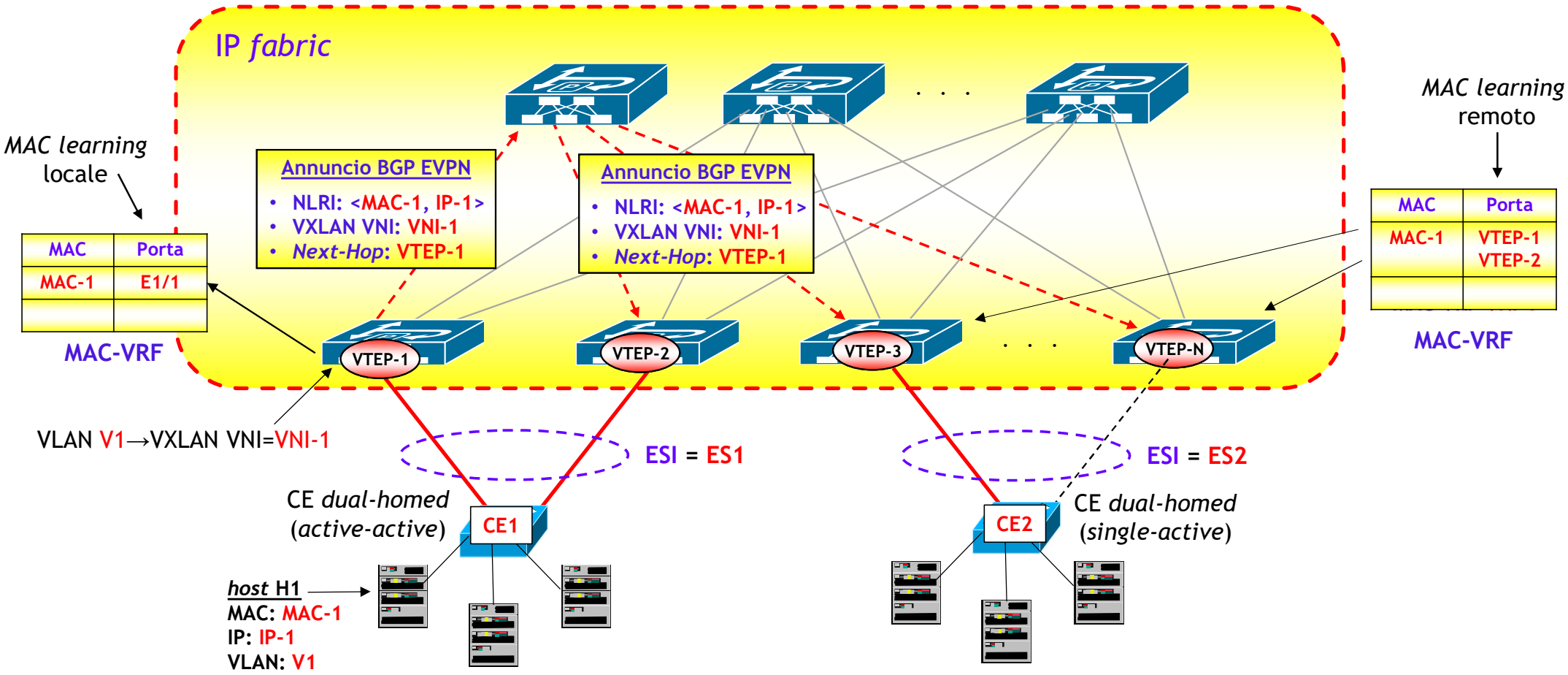
#1	EVPN come evoluzione dei servizi L2VPN
#2	Piano di controllo
#3	Funzionalità *multi-homing*
#4	*ARP/ND suppression*
#5	*MAC mobility*

Copyright Tiziano Tofoni

Un po' di terminologia...

- EVI (*EVPN Instance*): istanza EVPN definita su più PE
- MAC-VRF: tabella MAC su un PE per una determinata EVI
- ES (*Ethernet Segment*): insieme dei collegamenti presenti tra CE e PE
- ESI (*Ethernet Segment Identifier*): identificativo non nullo di lunghezza 10 *byte*, per l'insieme dei collegamenti tra un CE *multi-homed* a due o più PE
  - Nel caso di collegamento PE-CE *single-homed*, ESI=0
- *Ethernet tag*: è un identificativo di un dominio *broadcast* (es. una VLAN, un segmento VXLAN, ecc.)
- Modalità di ridondanza
  - *Active-Active*: i collegamenti sono gestiti in *Load Balancing*
  - *Single-Active*: i collegamenti sono gestiti in modalità *primario/backup*

Logica di funzionamento (con piano dati VXLAN)

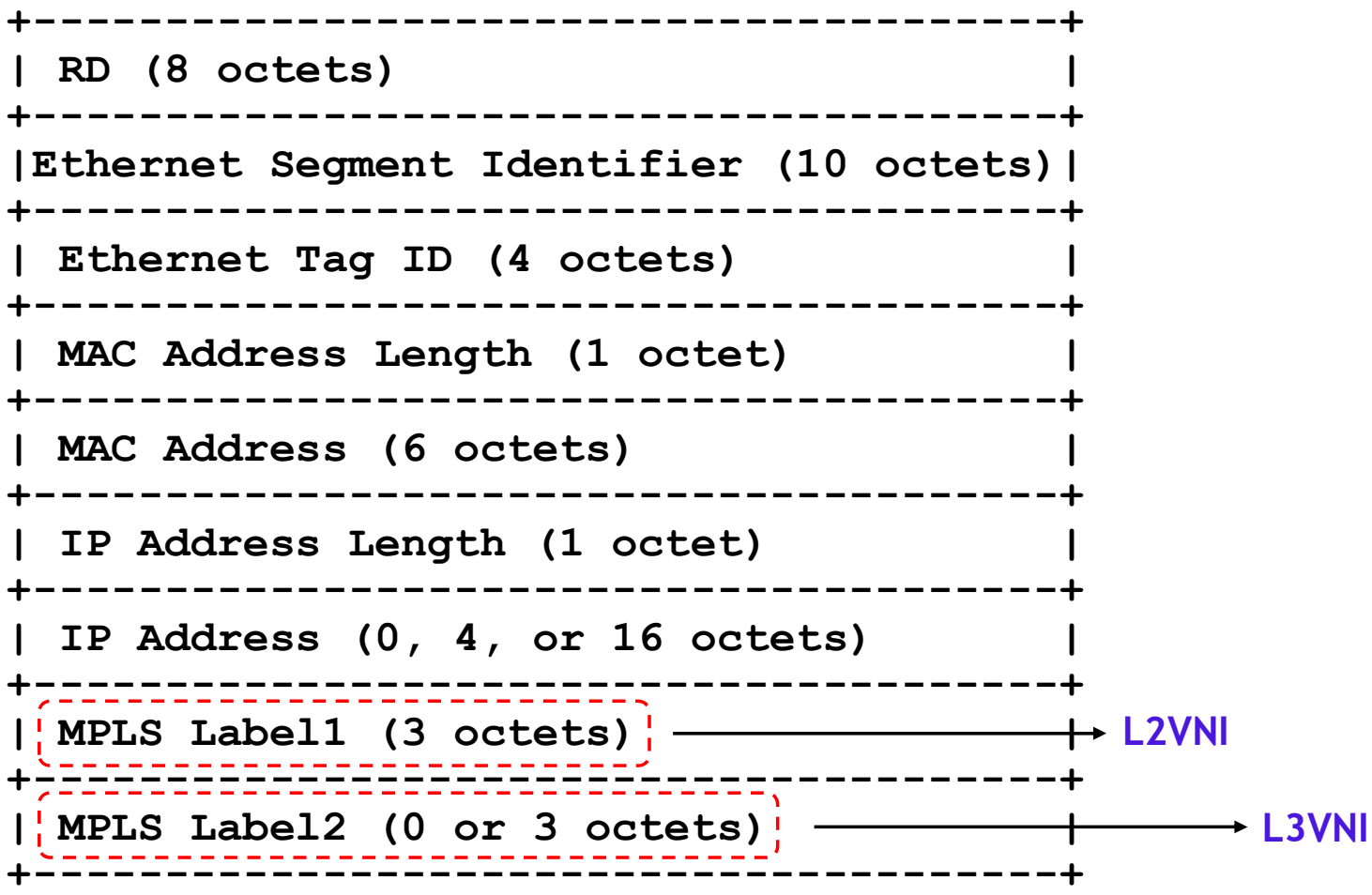


La nuova *address-family* BGP EVPN (AFI/SAFI=25/70)

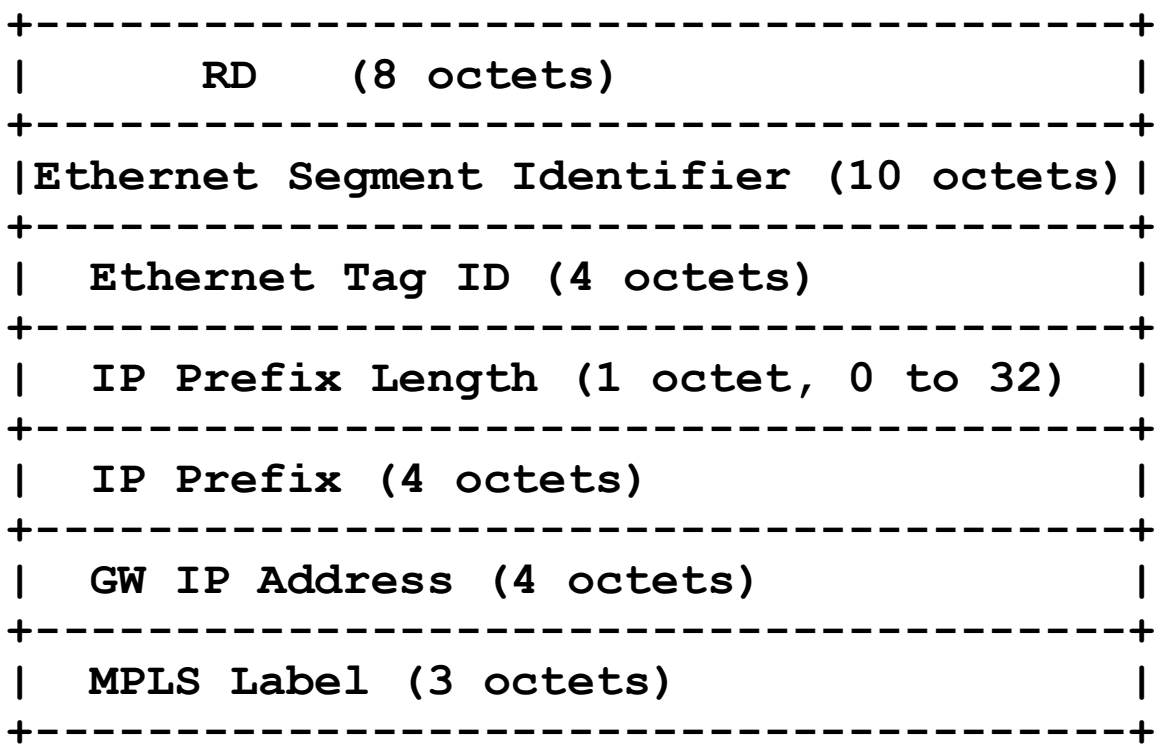
Type	Nome	Campi principali	Utilizzo
1	***Ethernet Auto-Discovery (A-D) route***	• *ESI* • *Ethernet tag* • *MPLS Label*	• *Fast convergence* • *Aliasing* • Annuncio dell'etichetta per lo *Split-Horizon*
2	***MAC/IP Advertisement route***	Vedi prossima pagina	• Annuncio dei MAC • Annuncio delle associazioni MAC/IP
3	***Inclusive Multicast Ethernet Tag route***	• *Ethernet tag* • Indirizzo IP del router che genera l'annuncio	Scoperta dei PE dove sono attestati CE della stessa istanza EVI
4	***Ethernet Segment route***	• *ESI* • Indirizzo IP del router che genera l'annuncio	• Scoperta dei PE *multi-homed* allo stesso segmento ES • Elezione del DF
5	***IP Prefix route***	• *ESI* • *Ethernet tag* • *Prefisso IP* • *GW IP address* • *MPLS Label*	Annuncio di prefissi IP

Copyright Tiziano Tofoni

EVPN NLRI di tipo 2 (*MAC/IP Advertisement route*)



EVPN NLRI di tipo 5 (*IP Prefix Route*)

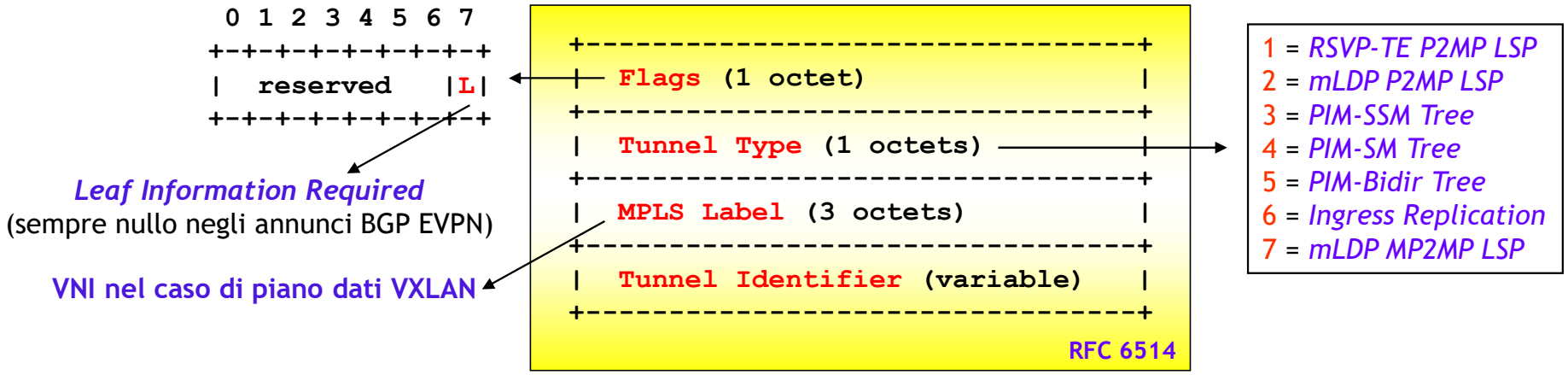


Le nuove *Extended Community* BGP

Extended Community	Utilizzo	Applicabilità NLRI
ESI MPLS Label	• Contiene l'etichetta per lo *Split-Horizon* • Indica la modalità di ridondanza (*All-Active/Single-Active*)	*Ethernet Auto-Discovery (A-D) route*
ES-Import	Limita l'ambito di propagazione delle NLRI di tipo *Ethernet Segment route*	*Ethernet Segment route*
MAC Mobility	Indica che un indirizzo MAC è passato da un segmento Ethernet a un altro	*MAC/IP Advertisement route*
Default Gateway	Indica l'associazione MAC/IP di un *Default Gateway*	*MAC/IP Advertisement route*

Segnalazione della modalità di trasporto del traffico BUM

- Il modello *EVPN* prevede l'utilizzo di varie modalità di trasporto del traffico BUM
  - LSP MPLS P2MP o MP2MP
  - *Ingress Replication*
  - PIM SM/SSM/BiDir
- La modalità utilizzata viene segnalata sul piano di controllo attraverso il nuovo attributo BGP «*PMSI Tunnel*»



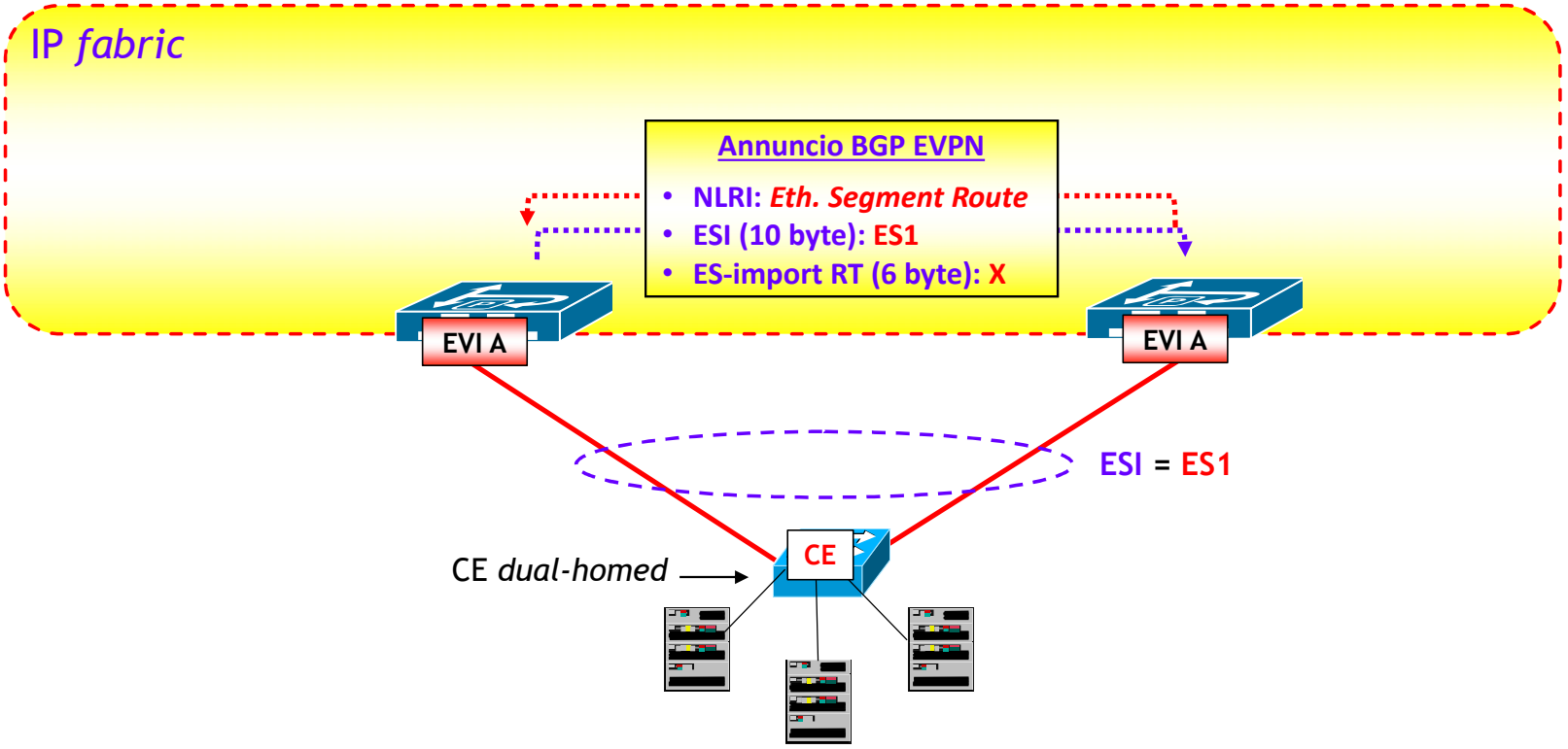
III° modulo: Ethernet VPN (EVPN)

#1	EVPN come evoluzione dei servizi L2VPN
#2	Piano di controllo
#3	Funzionalità *multi-homing*
#4	*ARP/ND suppression*
#5	*MAC mobility*

Copyright Tiziano Tofoni

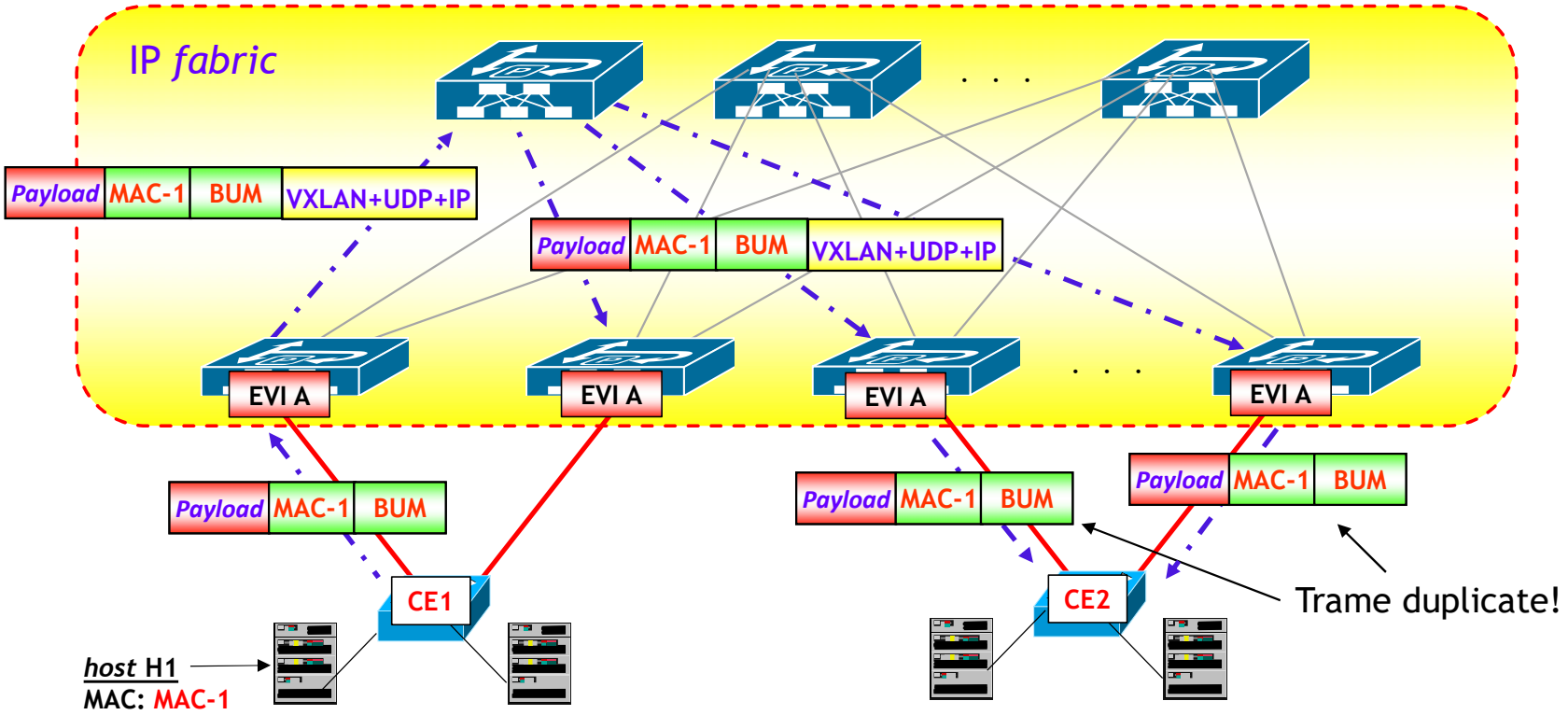
Collegamenti CE-PE *multi-homed*

- Passo preliminare: come fanno i PE (switch *Leaf*) a scoprire di essere sullo stesso ES?
  - Soluzione: tramite lo scambio di un annuncio BGP EVPN di tipo 4 (*Ethernet Segment Route*)

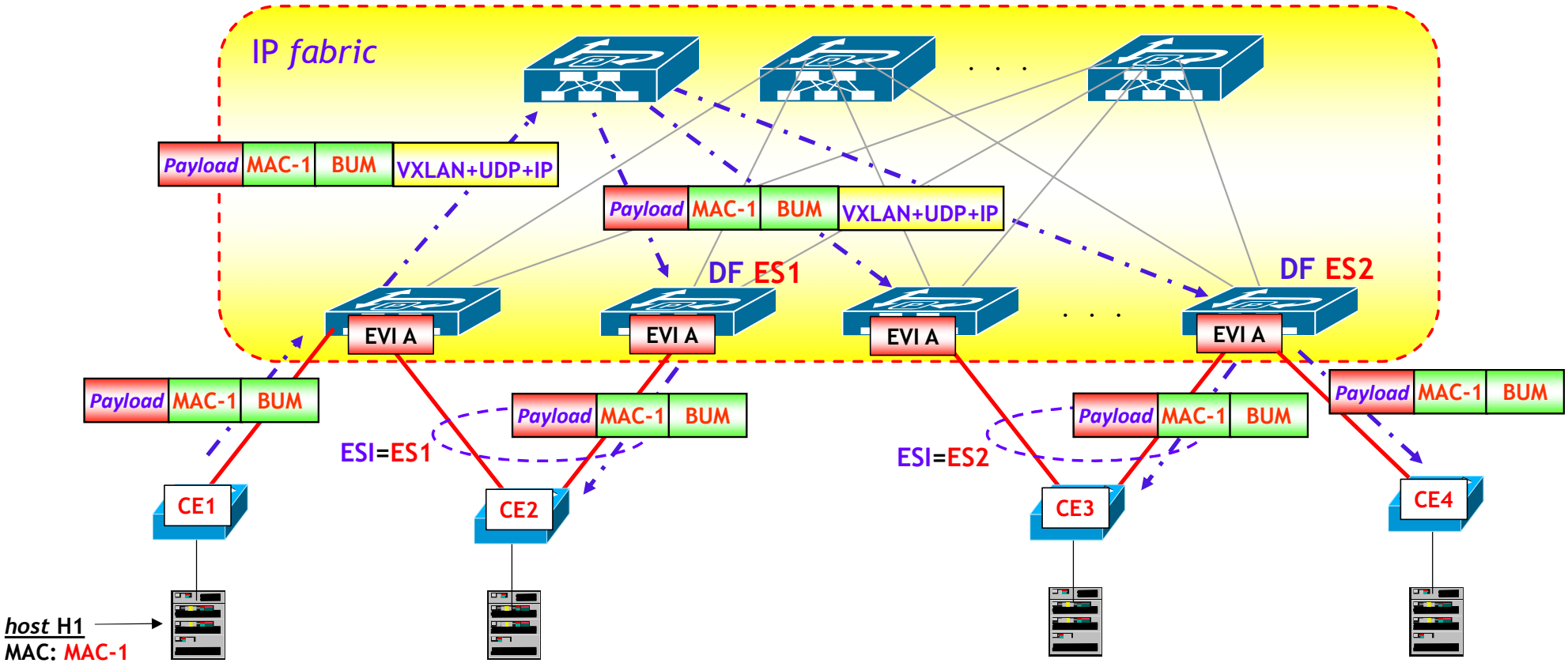


Duplicazione delle trame BUM

- **Problema**: come evitare la duplicazione delle trame BUM verso il CE destinazione, quando questo ha un collegamento *multi-homed active-active*?
- **Soluzione**: eleggere un *Designated Forwarder*

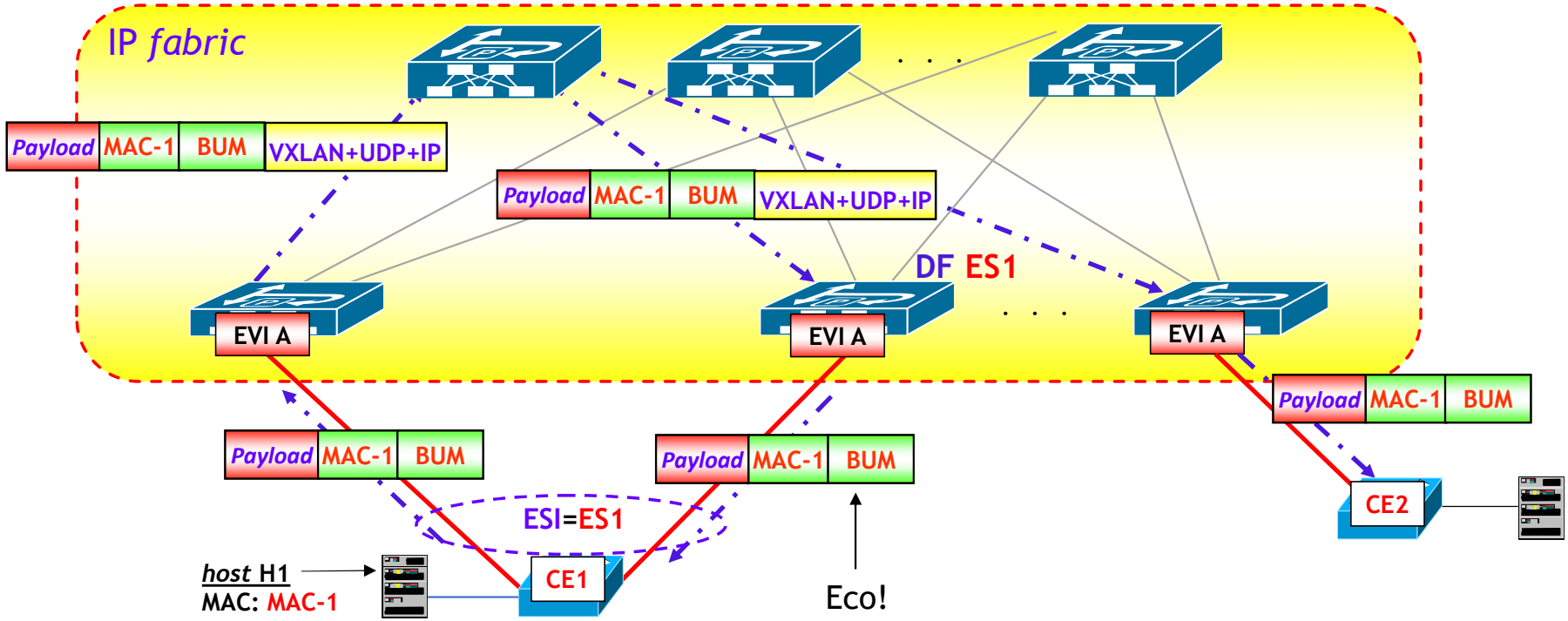


Esempio


IP fabric
Payload MAC-1 BUM VXLAN+UDP+IP
Payload MAC-1 BUM VXLAN+UDP+IP
DF ES1
DF ES2
EVI A
EVI A
EVI A
EVI A
Payload MAC-1 BUM
Payload MAC-1 BUM
Payload MAC-1 BUM
Payload MAC-1 BUM
ESI=ES1
ESI=ES2
CE1
CE2
CE3
CE4
host H1
MAC: MAC-1

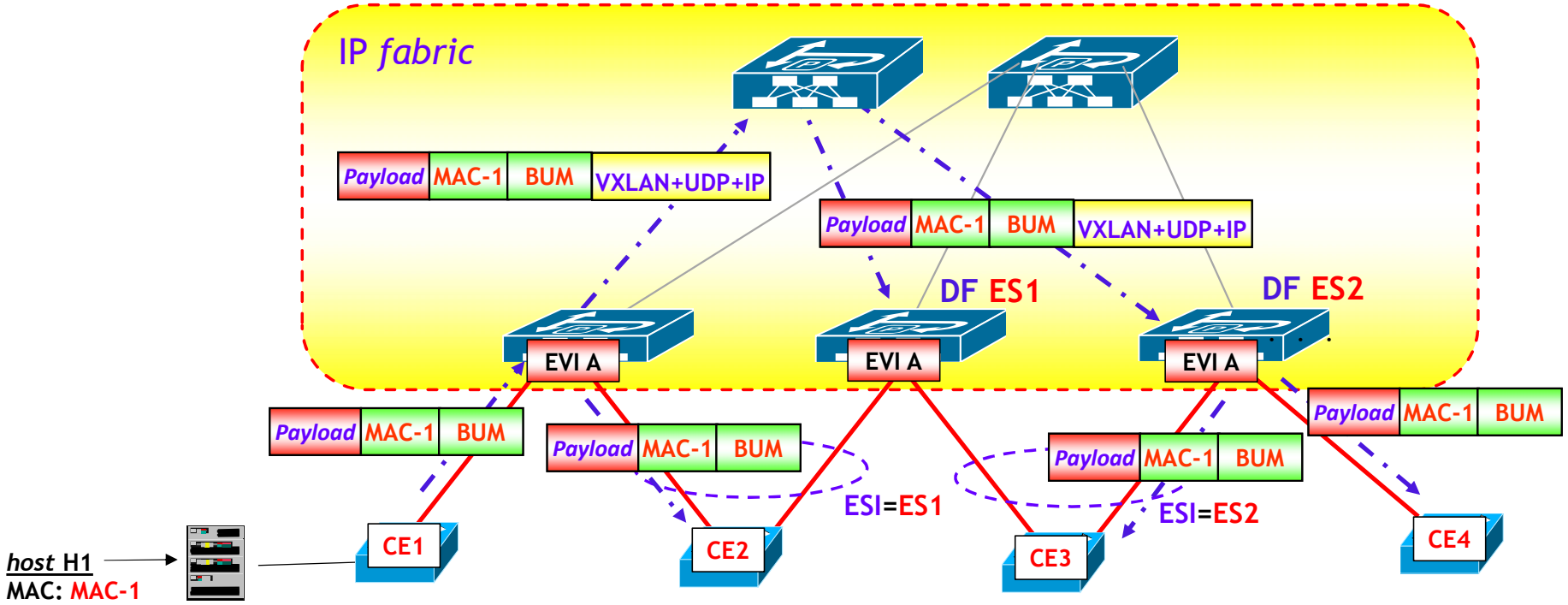
Split Horizon

- **Problema**: come fare in modo che trame BUM generate da un CE su un collegamento con un PE non-DF, non ritornino indietro al CE che le ha generate, quando questo ha un collegamento *multi-homed active-active*?
- **Soluzione**: impedire al PE non-DF di inoltrare le trame BUM verso il CE (*Split Horizon*)



Regole dello *Split Horizon*

- Regole dello *Split Horizon*
  - Ciascun PE esegue il *flooding* locale, a meno che l'indirizzo IP sorgente del tunnel VXLAN sia di una VTEP che condivide con il PE lo stesso ESI, nel qual caso la trama BUM non viene inoltrata al CE
    - NOTA: questo vale anche se il PE ha il ruolo di DF per il segmento ESI
  - Solo il DF inoltra le trame BUM verso i CE

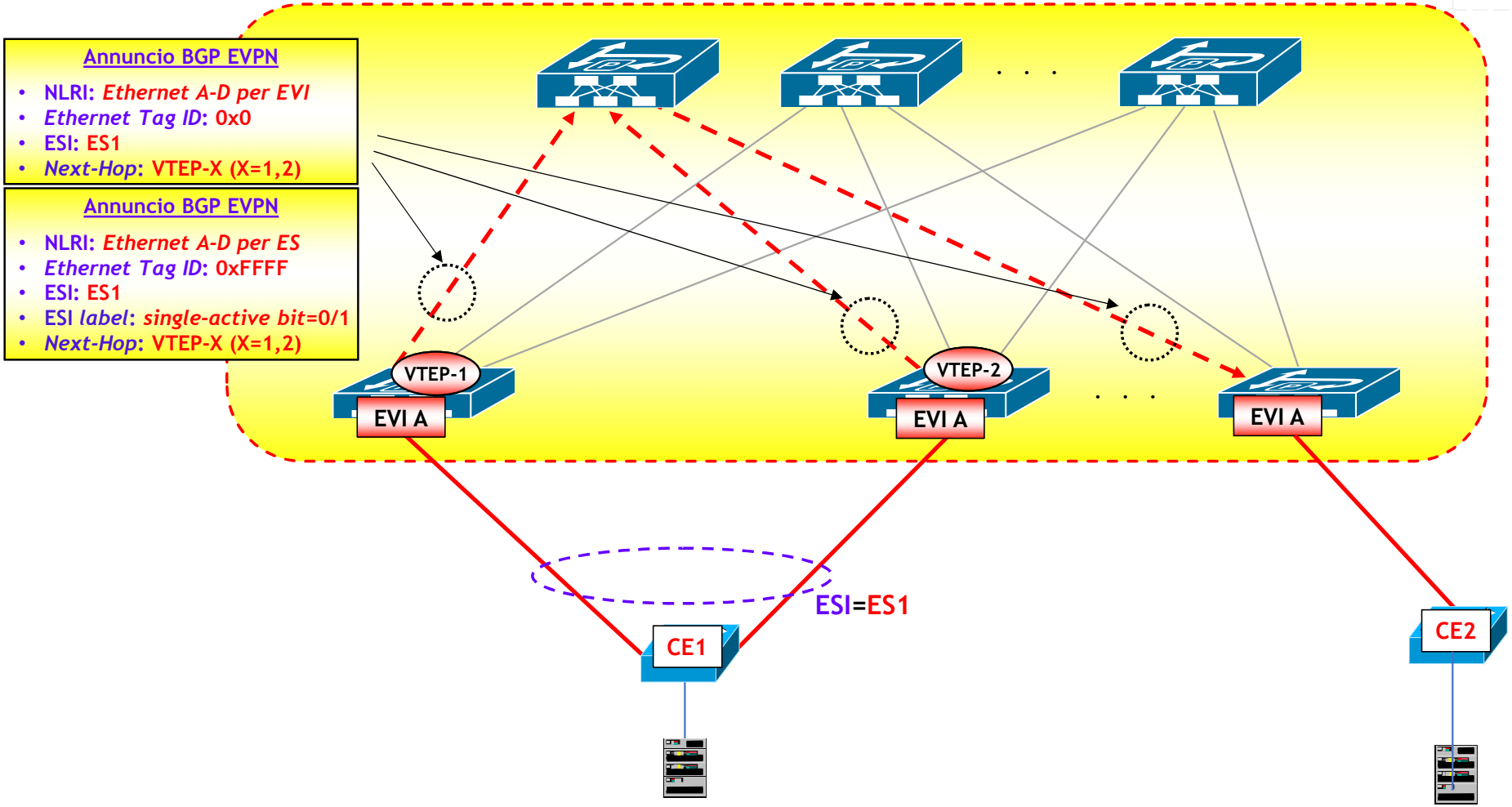


Aliasing e *Backup path* (1/3)

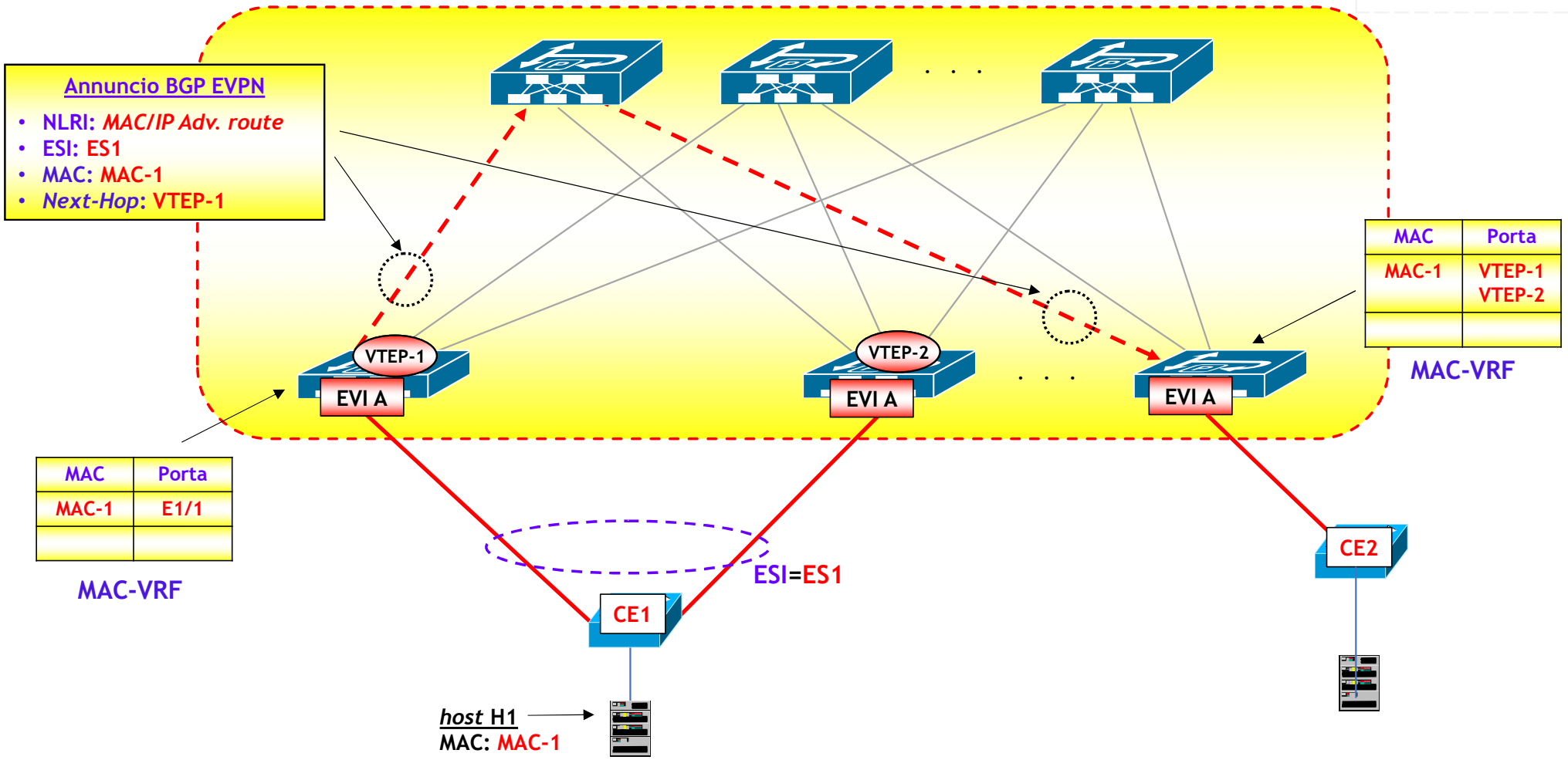
- **Problema**: come possono i PE remoti effettuare un *load balancing* del traffico destinato all'indirizzo MAC annunciato da un solo PE di un ES?
- **Soluzione**: utilizzare annunci BGP EVPN di tipo 1

- Due tipi di annunci BGP EVPN di tipo 1
  - Ethernet A-D per ES
    - *Ethernt Tag ID* = **0xFFFF**
    - Contiene il valore di ESI e la *Extended Community ESI label* (vedi note)
    - Scopo: serve ai PE che ricevono l'annuncio per capire se l'ES annunciato lavora in modalità *all-active* oppure *single-active*
  - Ethernet A-D per EVI
    - *Ethernt Tag ID* = **0x0**
    - Contiene il valore di ESI
    - Scopo: serve ai PE che ricevono l'annuncio per capire quali sono i PE terminazione dell'ES

Aliasing e Backup path (2/3)

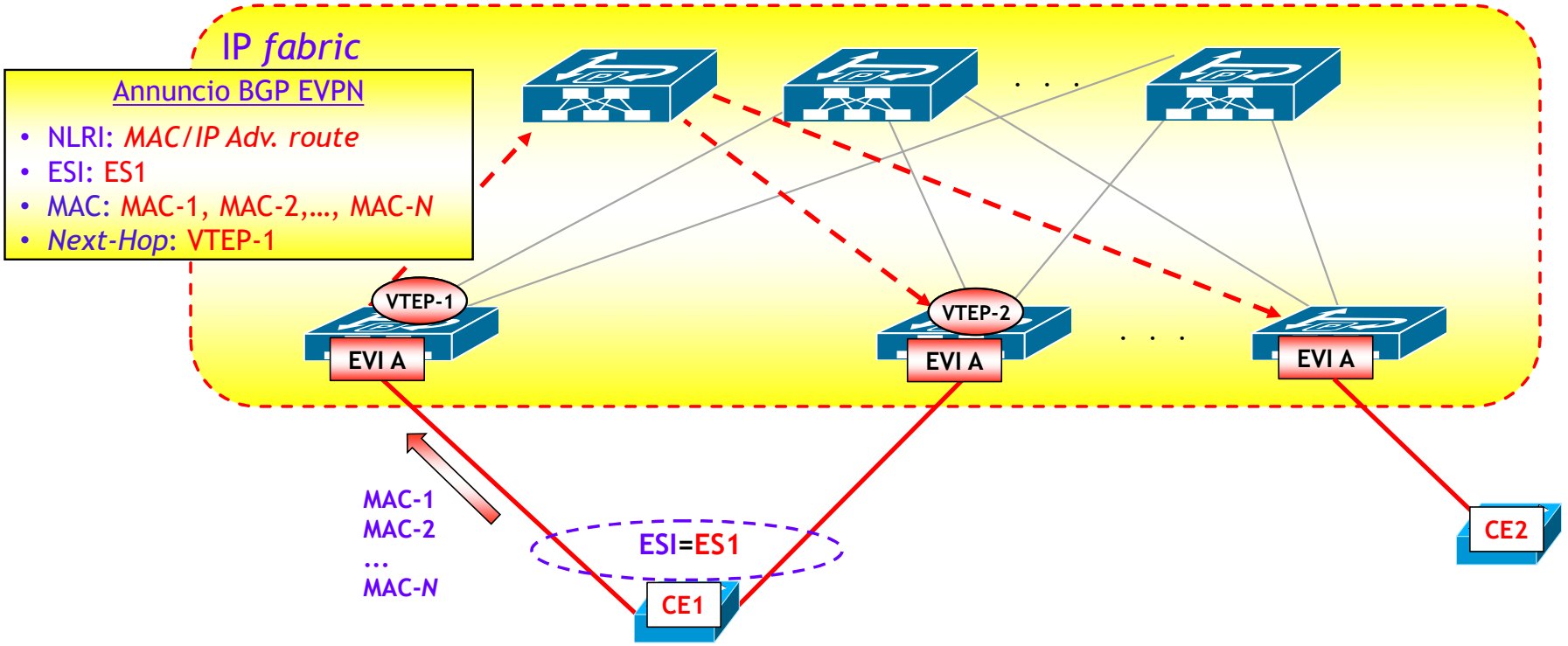


Aliasing e Backup path (3/3)



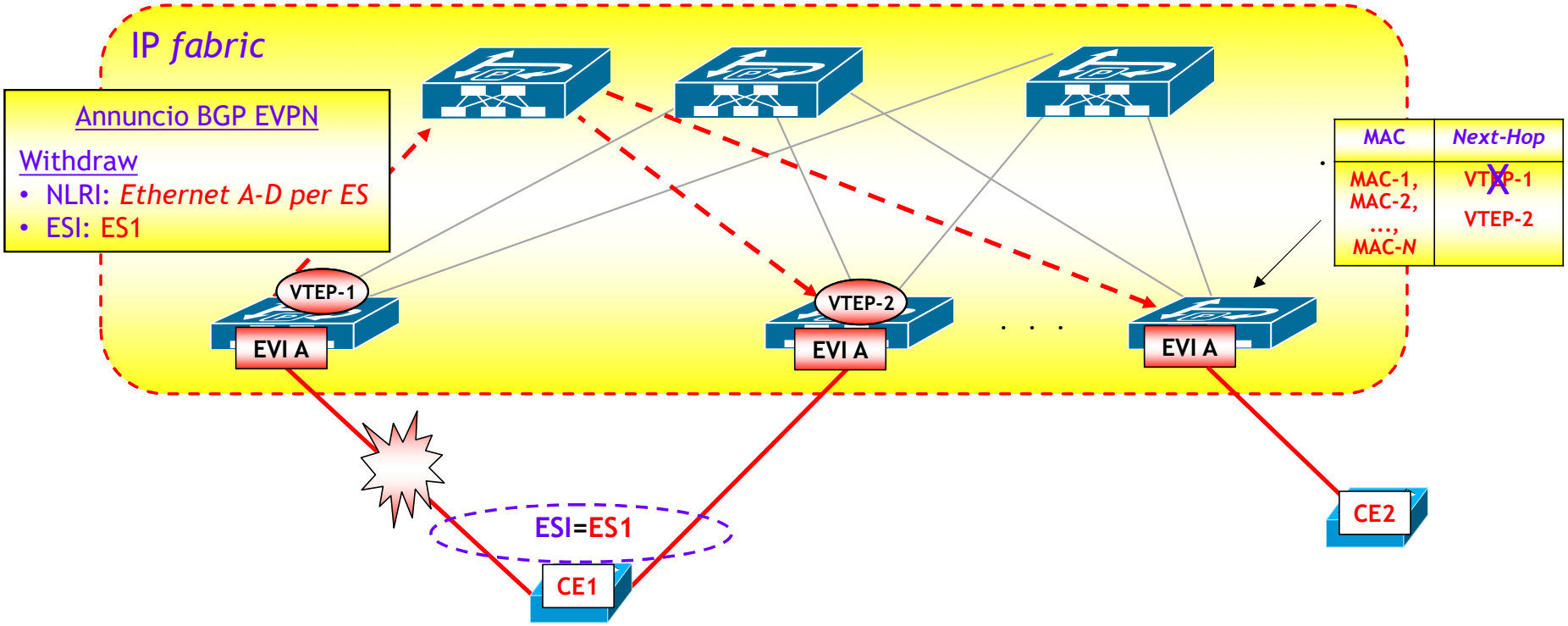
Fast Convergence (1/2)

- **Problema:** in un ambiente dove gli indirizzi MAC sono molti, i tempi di convergenza potrebbero diventare molto alti



Fast Convergence (2/2)

- **Soluzione:** ritirare prima l'annuncio BGP EVPN del tipo *Ethernet A-D per ES* e quindi i singoli MAC e MAC/IP
  - VANTAGGIO: la velocità di convergenza non dipende dal numero di MAC appresi dal PE

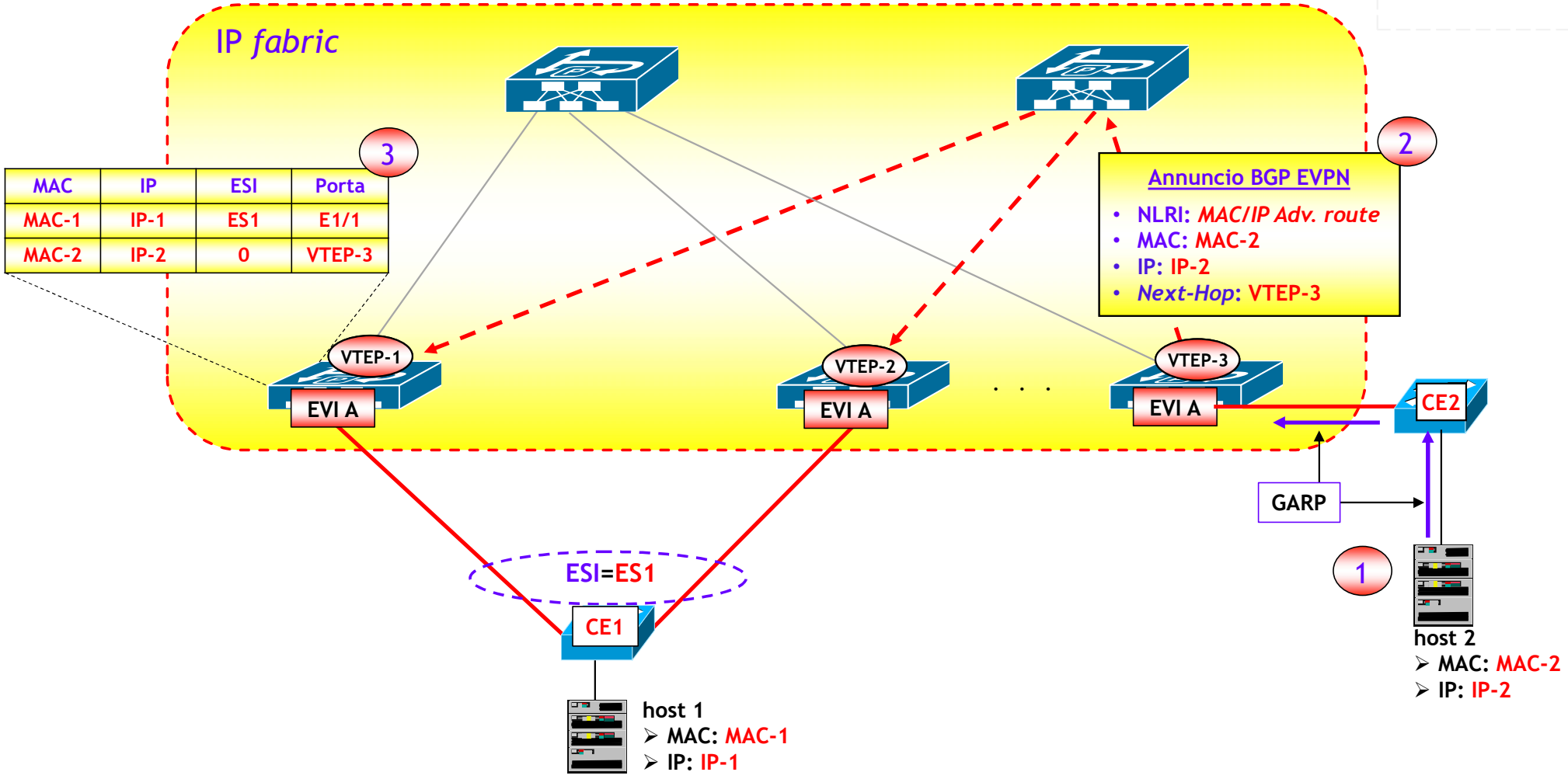


III° modulo: Ethernet VPN (EVPN)

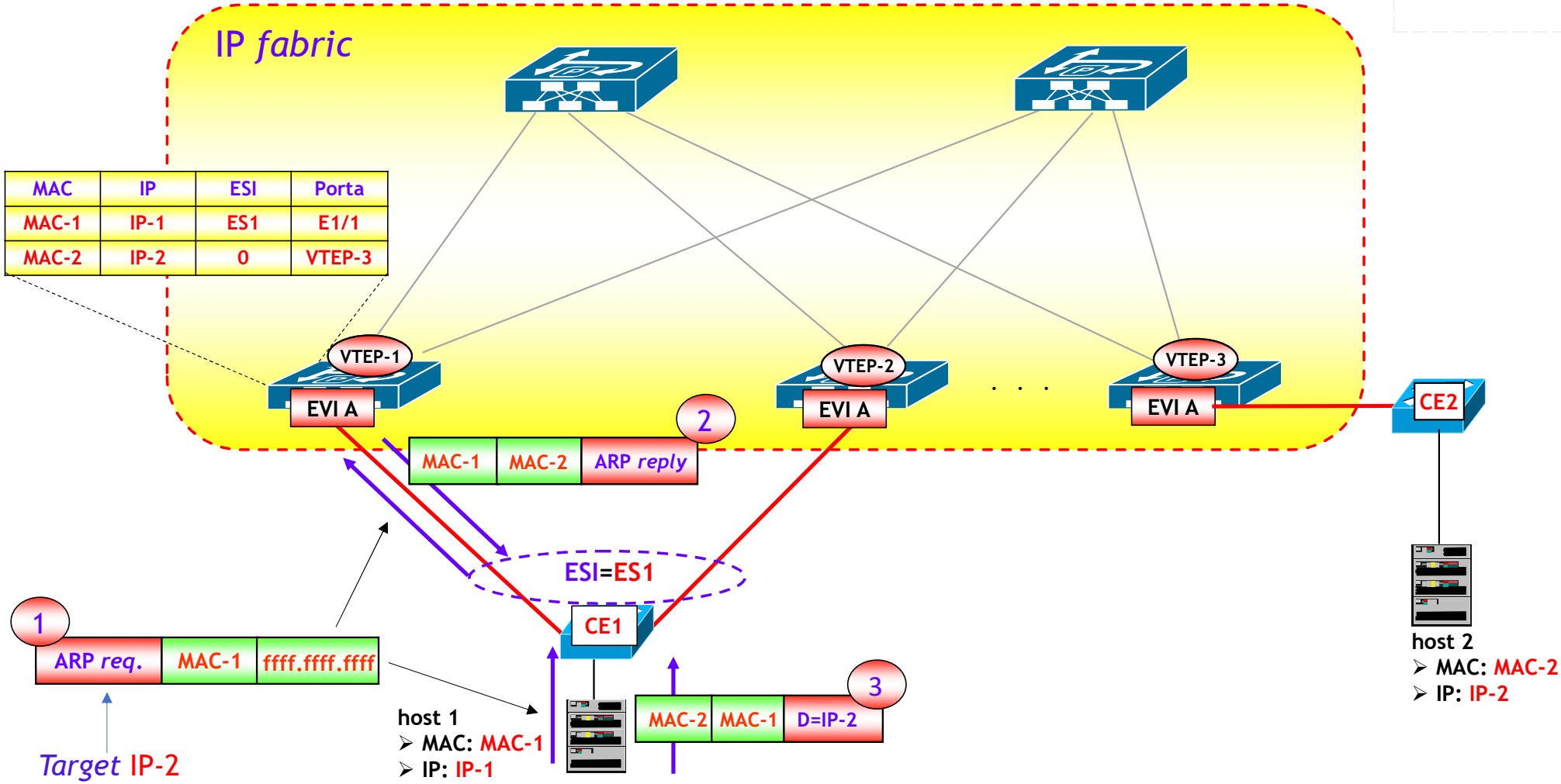
#1	EVPN come evoluzione dei servizi L2VPN
#2	Piano di controllo
#3	Funzionalità *multi-homing*
#4	*ARP/ND suppression*
#5	*MAC mobility*

Copyright Tiziano Tofoni

Funzionamento (1/2)



Funzionamento (2/2)



III° modulo: Ethernet VPN (EVPN)

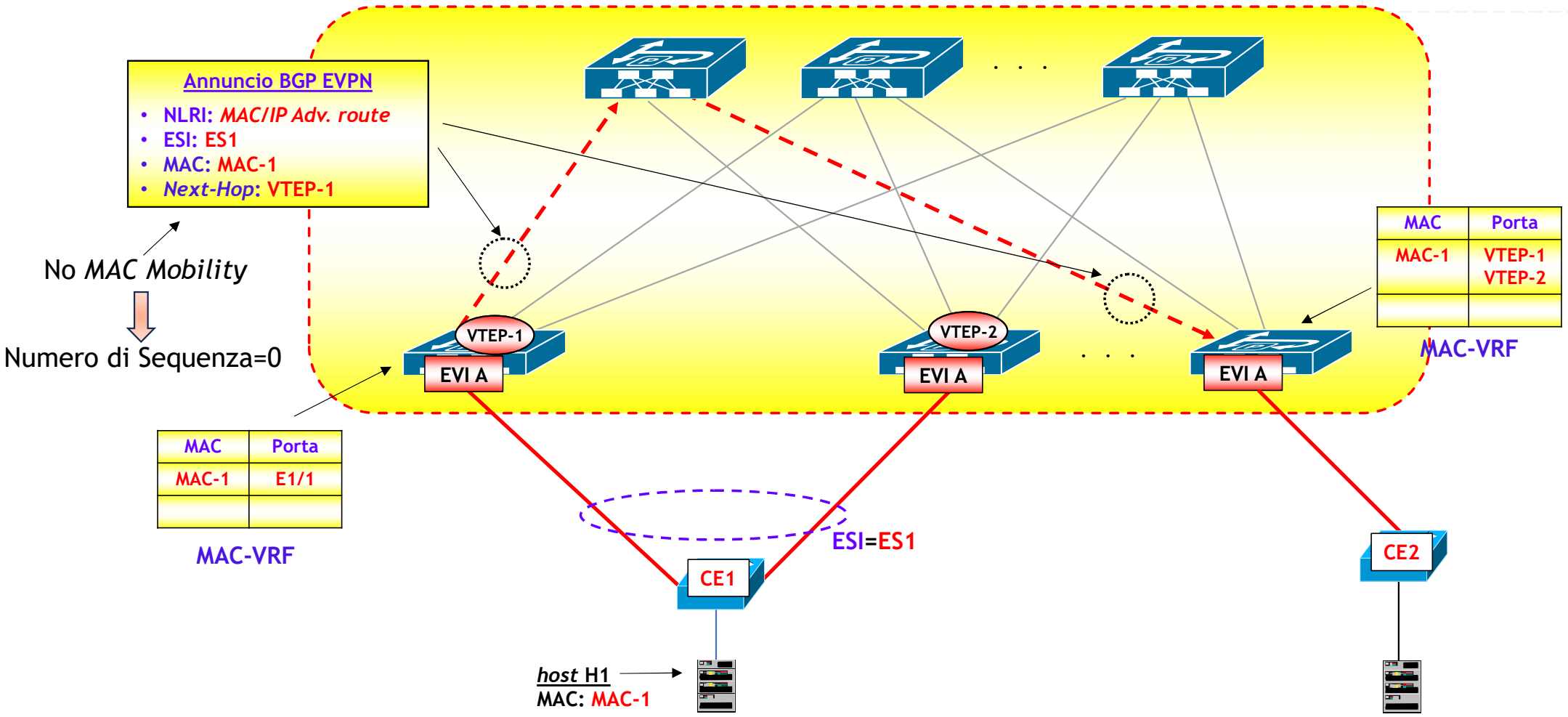
#1	EVPN come evoluzione dei servizi L2VPN
#2	Piano di controllo
#3	Funzionalità *multi-homing*
#4	*ARP/ND suppression*
#5	*MAC mobility*

Copyright Tiziano Tofoni

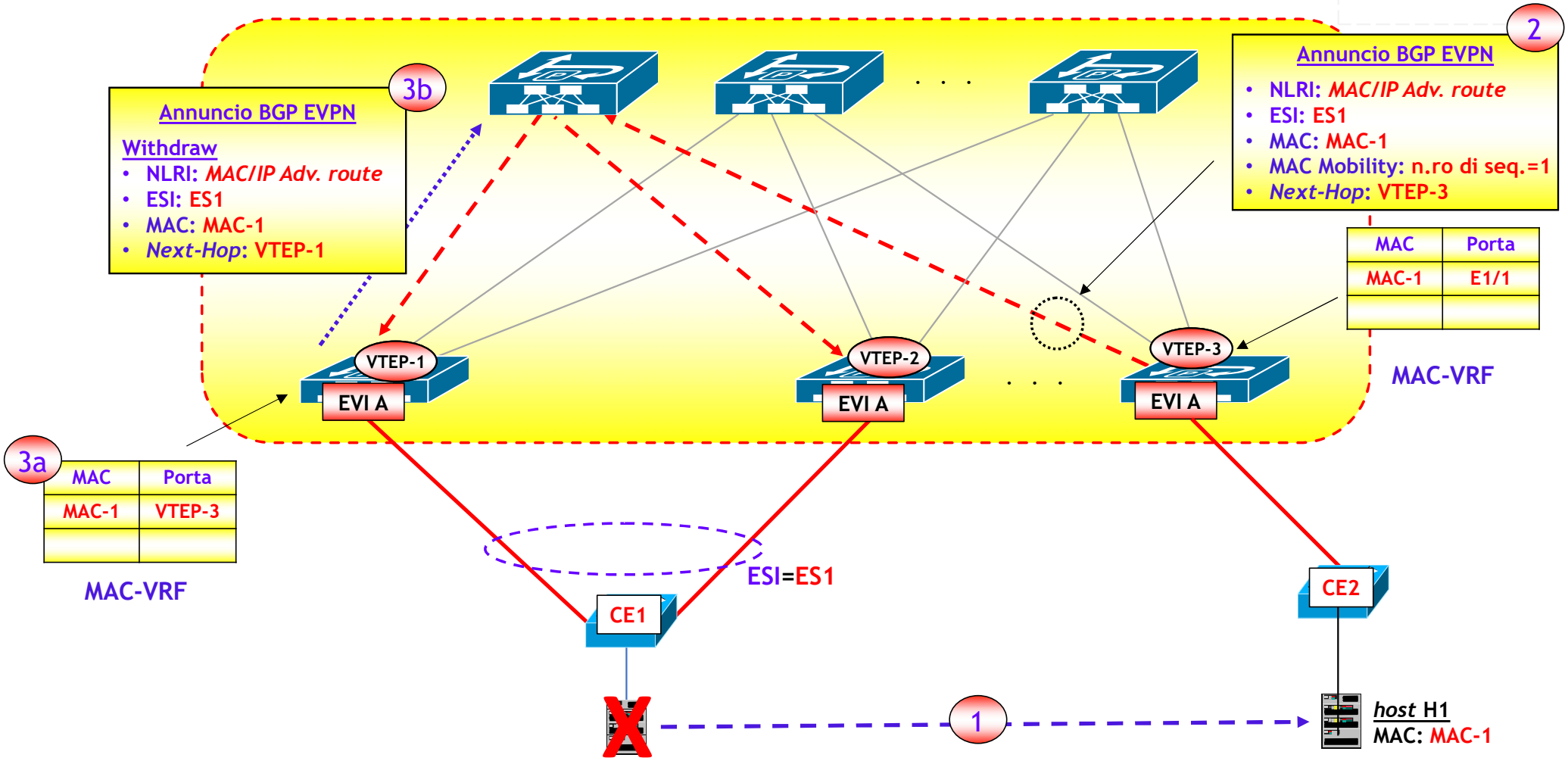
Funzionamento

- **Problema**: Lo spostamento di un host da un PE a un altro PE comporta la presenza dello stesso indirizzo MAC su due MAC-VRF diverse, quella di partenza e quella di arrivo
- **Soluzione**: utilizzare un «numero di sequenza» degli spostamenti che consenta ai PE di valutare se un *host* è stato spostato o meno
  - Il numero di sequenza è contenuto nella BGP *Extended Community MAC Mobility*
- Procedimento:
  1. I PE dell'ES dove è "approdato" il MAC, ne annunciano la presenza tramite annunci BGP EVPN di tipo *MAC/IP Advertisement*, aggiungendo all'annuncio la *Extended Community MAC Mobility* con il numero di sequenza aumentato di una unità (Nota: nel primo annuncio l'attributo *MAC Mobility* non viene inserito, e ciò corrisponde di default a un numero di sequenza nullo)
  2. Tutti gli annunci BGP EVPN di tipo *MAC/IP Advertisement* inviati dal o dai PE da dove era raggiungibile in precedenza l'indirizzo MAC, devono essere eliminati da tutti gli altri PE dove sono attestati CE della stessa EVI.

Esempio (1/2)


Annuncio BGP EVPN
• NLRI: MAC/IP Adv. route
• ESI: ES1
• MAC: MAC-1
• Next-Hop: VTEP-1
No MAC Mobility
Numero di Sequenza=0
VTEP-1
EVI A
VTEP-2
EVI A
EVI A
MAC Porta
MAC-1 VTEP-1 VTEP-2
MAC-VRF
MAC Porta
MAC-1 E1/1
MAC-VRF
ESI=ES1
CE1
CE2
host H1
MAC: MAC-1

Esempio (2/2)


2
Annuncio BGP EVPN
• NLRI: MAC/IP Adv. route
• ESI: ES1
• MAC: MAC-1
• MAC Mobility: n.ro di seq.=1
• Next-Hop: VTEP-3
MAC | Porta
MAC-1 | E1/1
MAC-VRF
3b
Annuncio BGP EVPN
Withdraw
• NLRI: MAC/IP Adv. route
• ESI: ES1
• MAC: MAC-1
• Next-Hop: VTEP-1
VTEP-1
VTEP-2
VTEP-3
EVI A
EVI A
EVI A
3a
MAC | Porta
MAC-1 | VTEP-3
MAC-VRF
ESI=ES1
CE1
CE2
1
host H1
MAC: MAC-1

IV° modulo

Integrazione EVPN/VXLAN

Copyright Tiziano Tofoni

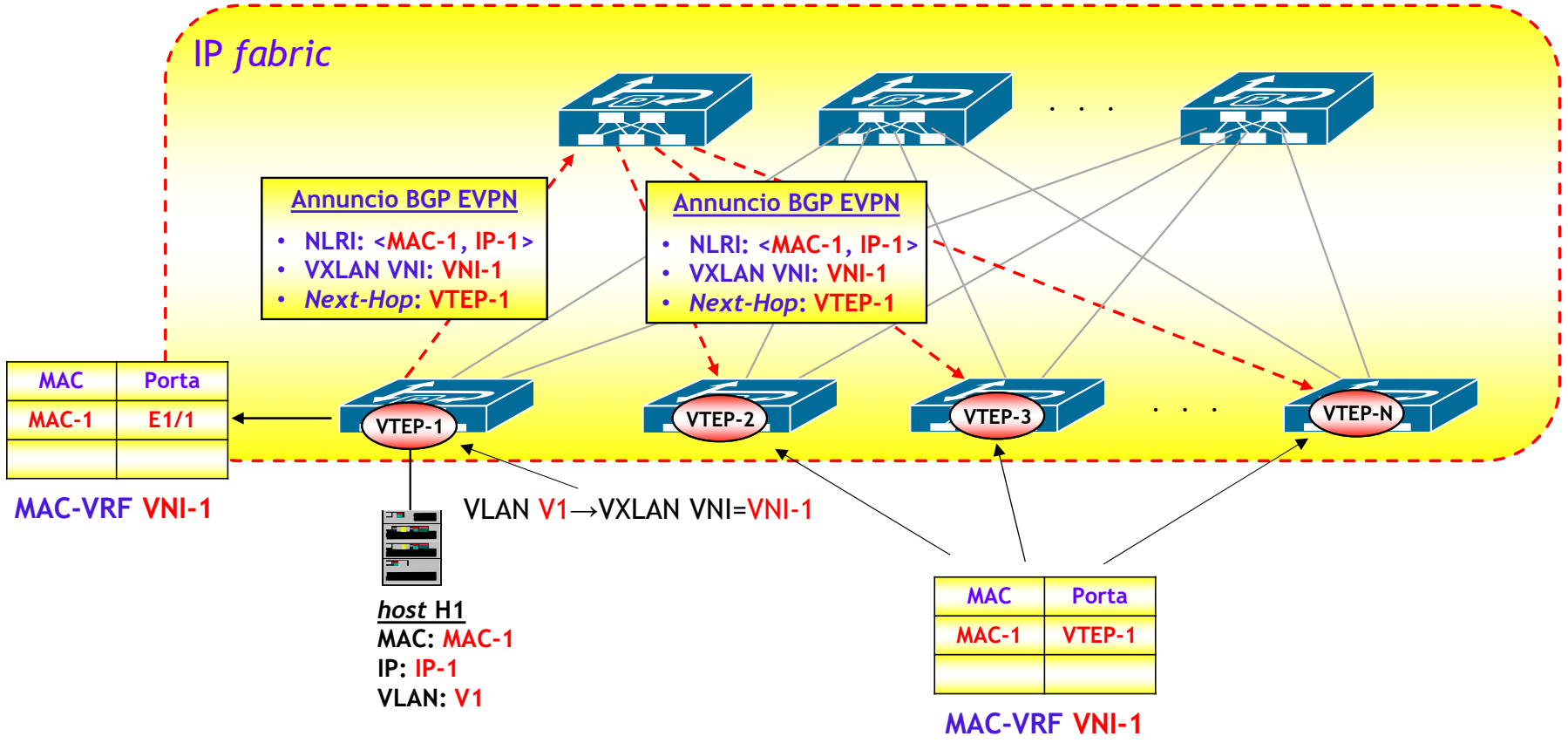
IV° modulo: Integrazione EVPN/VXLAN

#1 Piano dati VXLAN con piano di controllo EVPN

#2 *Inter-VXLAN routing*

Copyright Tiziano Tofoni

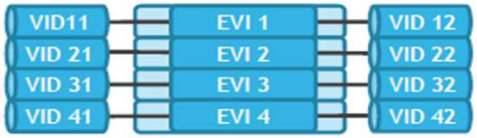
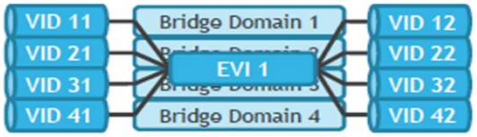
VXLAN con piano di controllo EVPN



Segmentazione dei *tenant*

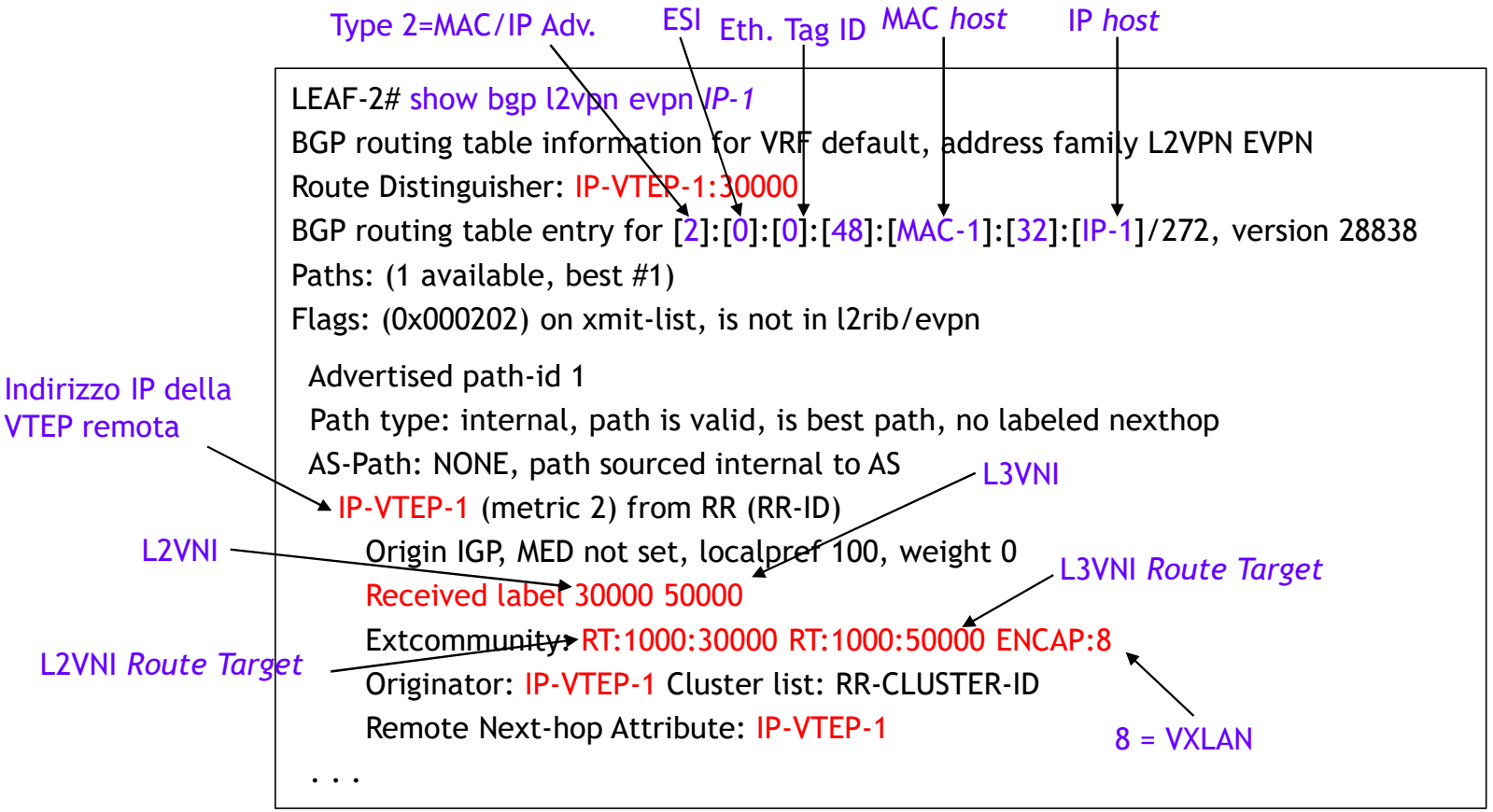
- Due opzioni
  - Servizio *VLAN Based*→singolo segmento VXLAN per EVI
  - Servizio *VLAN Aware Bundle*→ Più segmenti VXLAN per EVI

Non utilizzato

	VLAN Based Service Interface	VLAN Bundle Service Interface	VLAN Aware Bundle Service Interface
Service Interface Diagram			
Mapping Between VLAN ID and EVI	1:1	N:1	N:1
Customer VID per EVI	Single	Multiple	Multiple
Bridge Domains per EVI	Single	Single	Multiple
Overlapping MACs Across VLANs	✓	✗	✓
VLAN Translation	✓	✗	✓

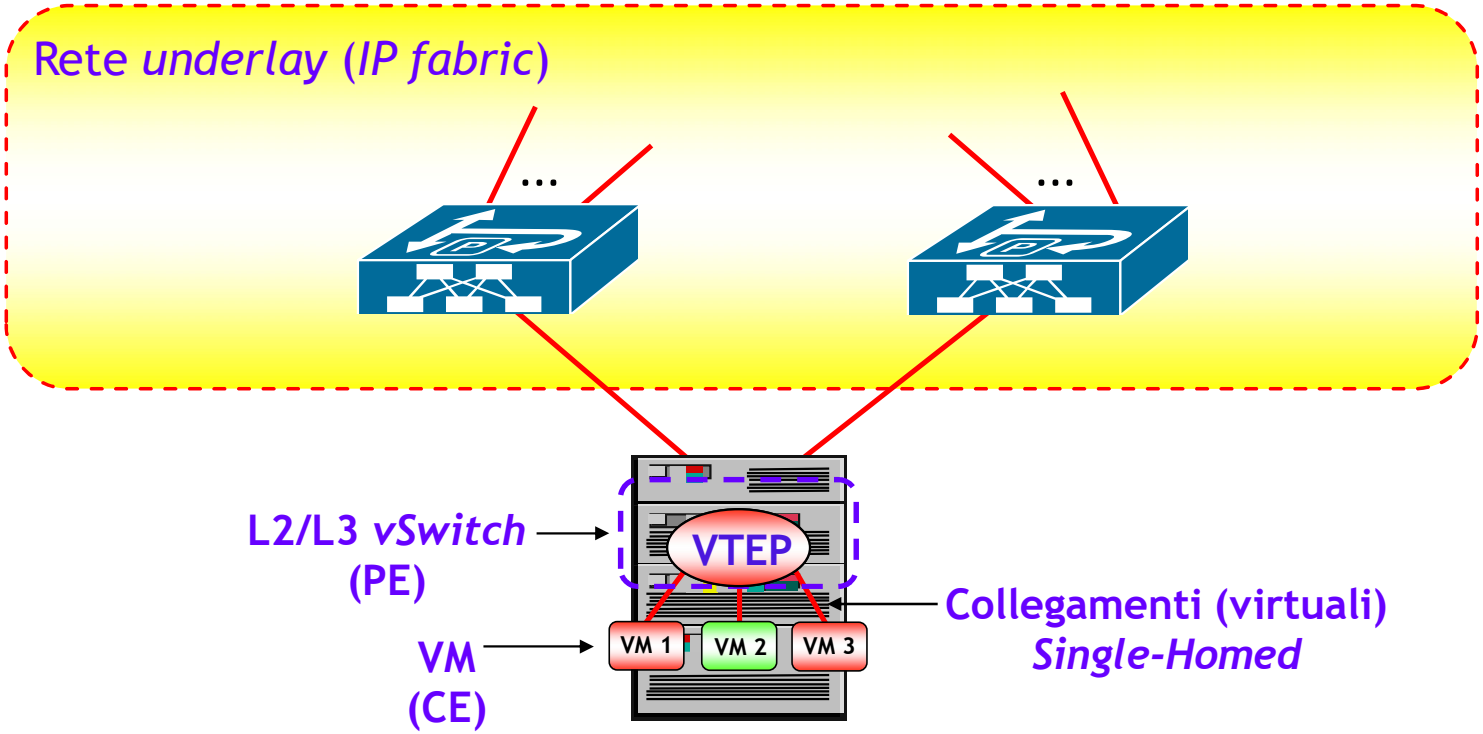
Trasporto dei VNI

- Il valore di VNI viene trasportato nei campi riservati alle etichette MPLS dei messaggi BGP EVPN con NLRI di tipo 2



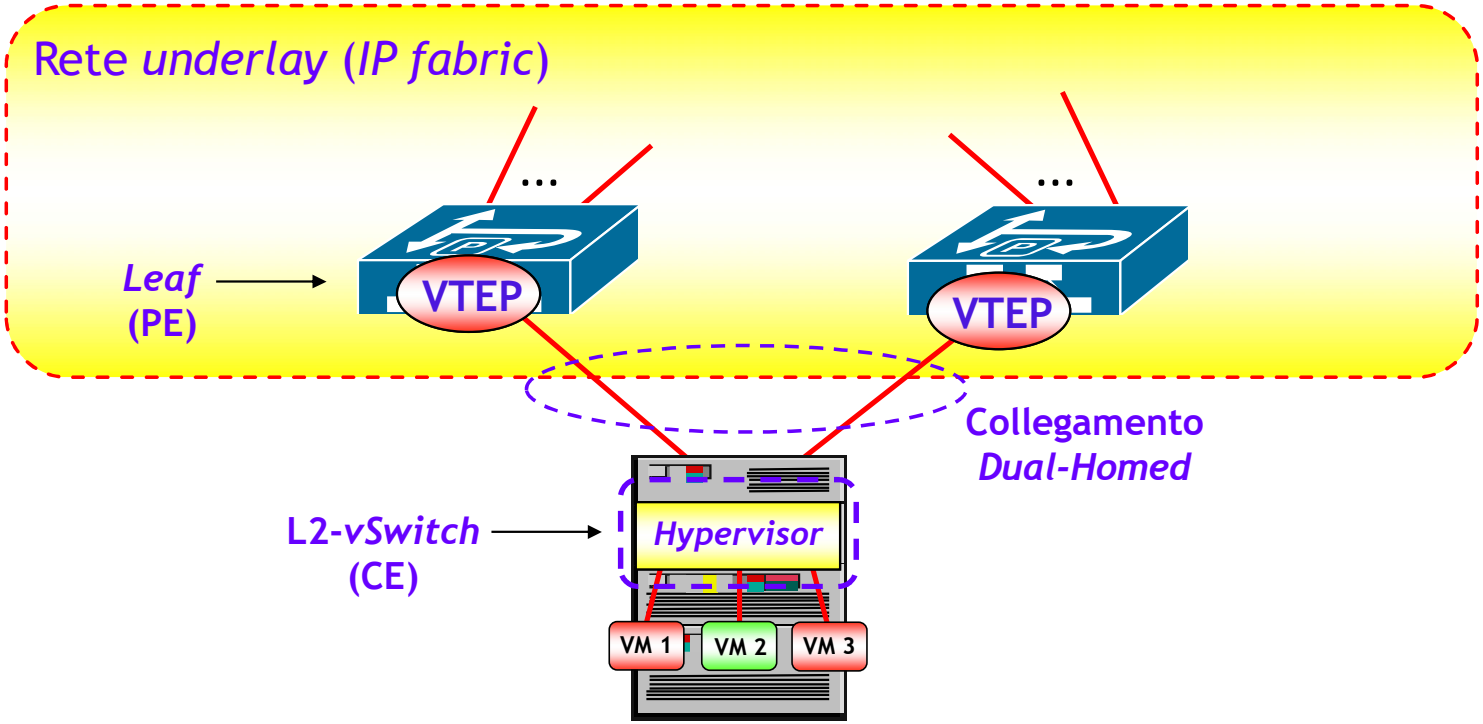
Aspetti operativi: VTEP in uno *vSwitch*

- La funzione VTEP è svolta via software (o hardware se supportato dalla NIC) e localizzata nello *vSwitch* di un *hypervisor*
- In questo caso le funzionalità *multi-homing* di EVPN non sono necessarie

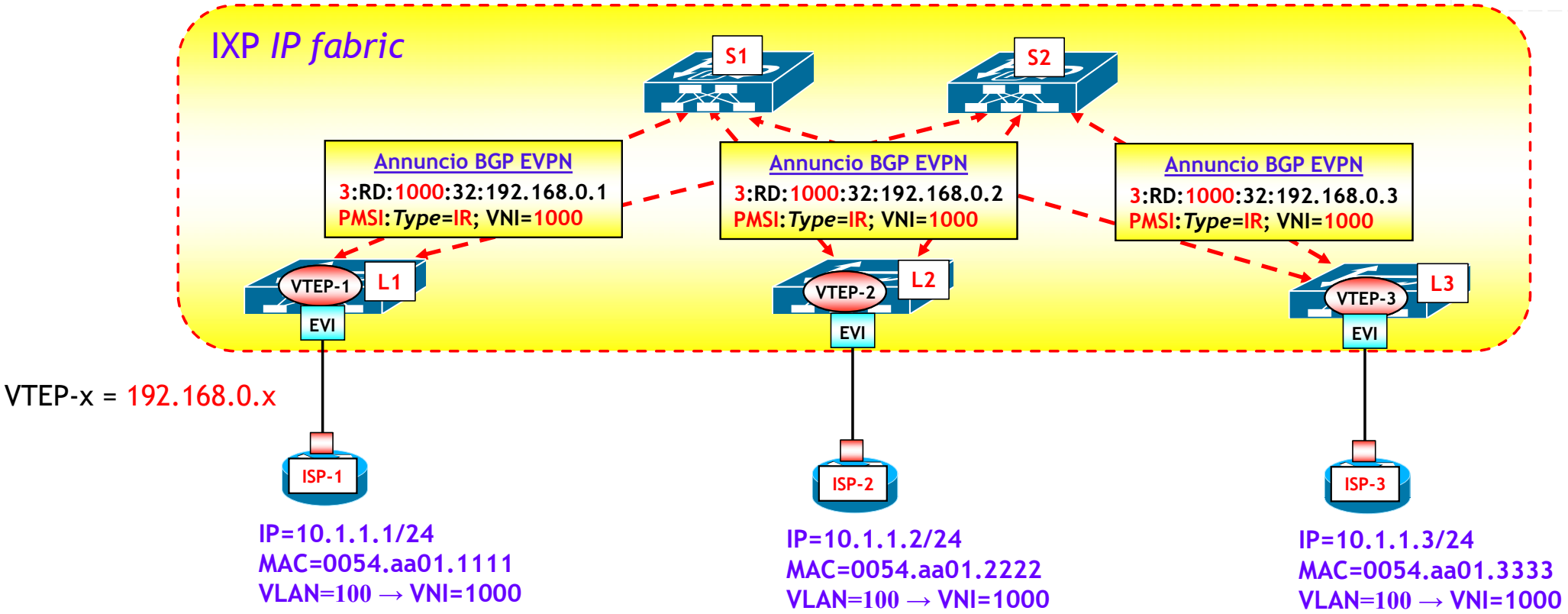


Aspetti operativi: VTEP in uno *switch Leaf*

- La funzione VTEP è realizzata in hardware in uno *switch Leaf*
- In questo caso possono essere utilizzate le funzionalità *multi-homing* di EVPN, o in alternativa MC-LAG

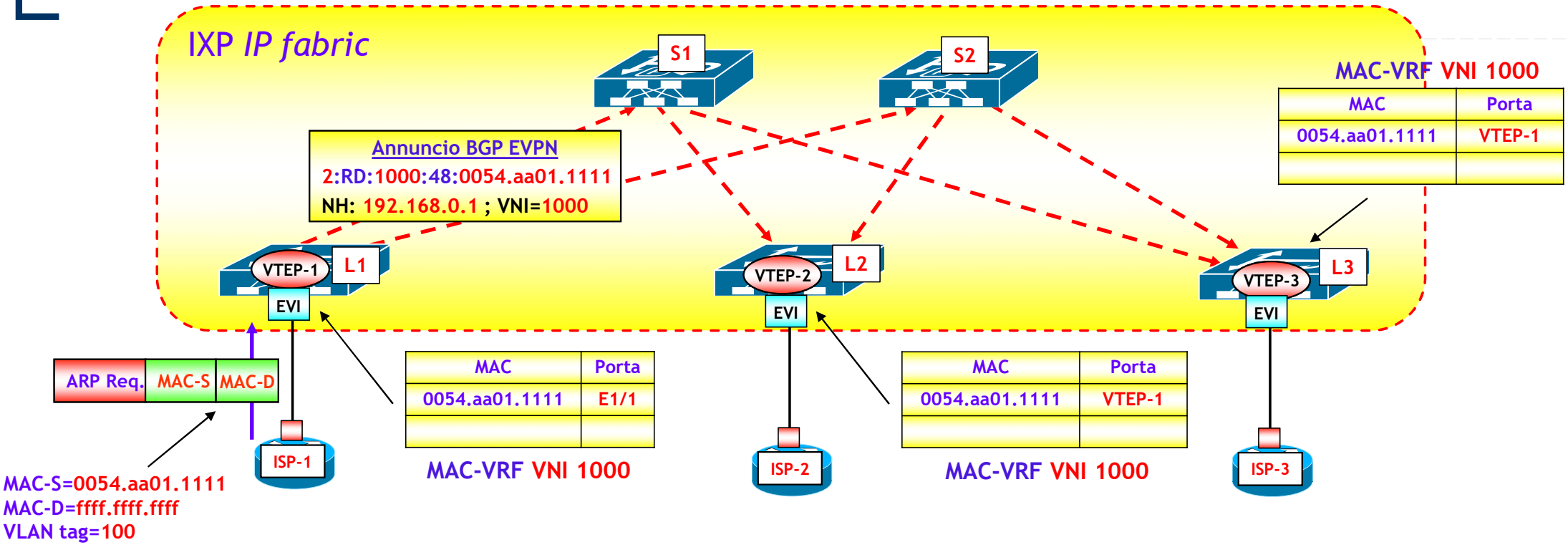


Esempio: *IP fabric* di un IXP (1/5)



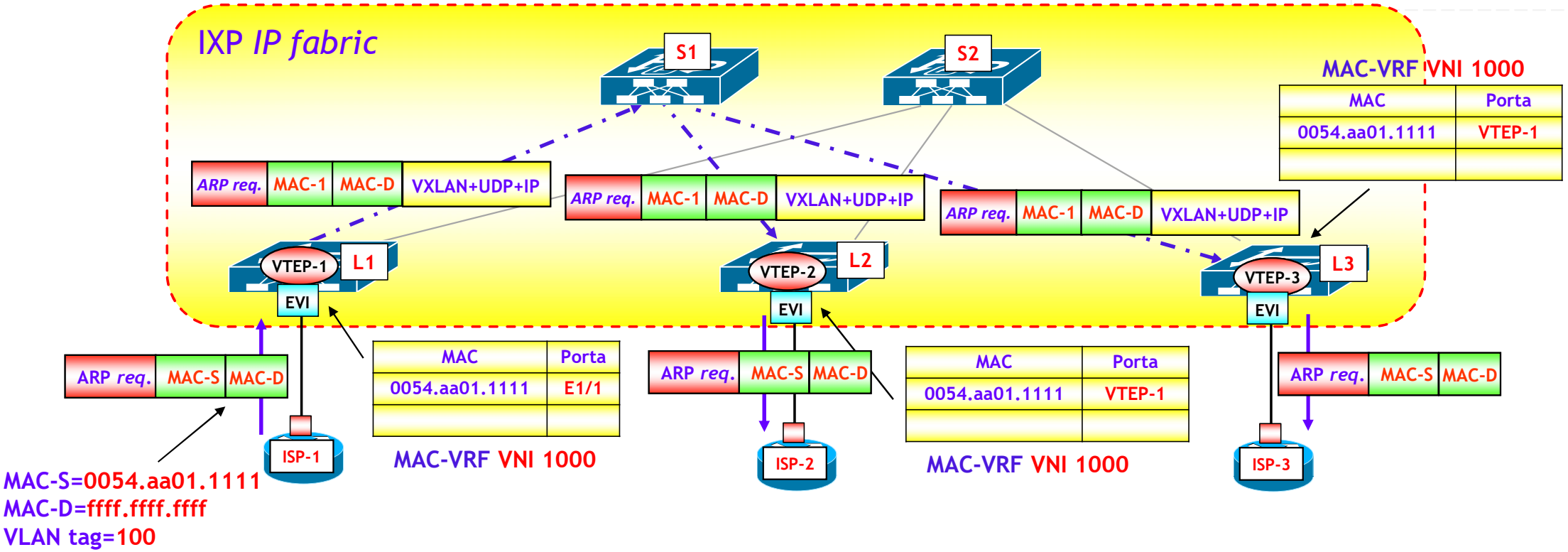
- Passo preliminare: ciascuno degli switch *Leaf* dove è presente una EVI della stessa EVPN genera un annuncio di tipo 3 (*Inclusive Multicast Ethernet Tag*)

Esempio: *IP fabric* di un IXP (2/5)



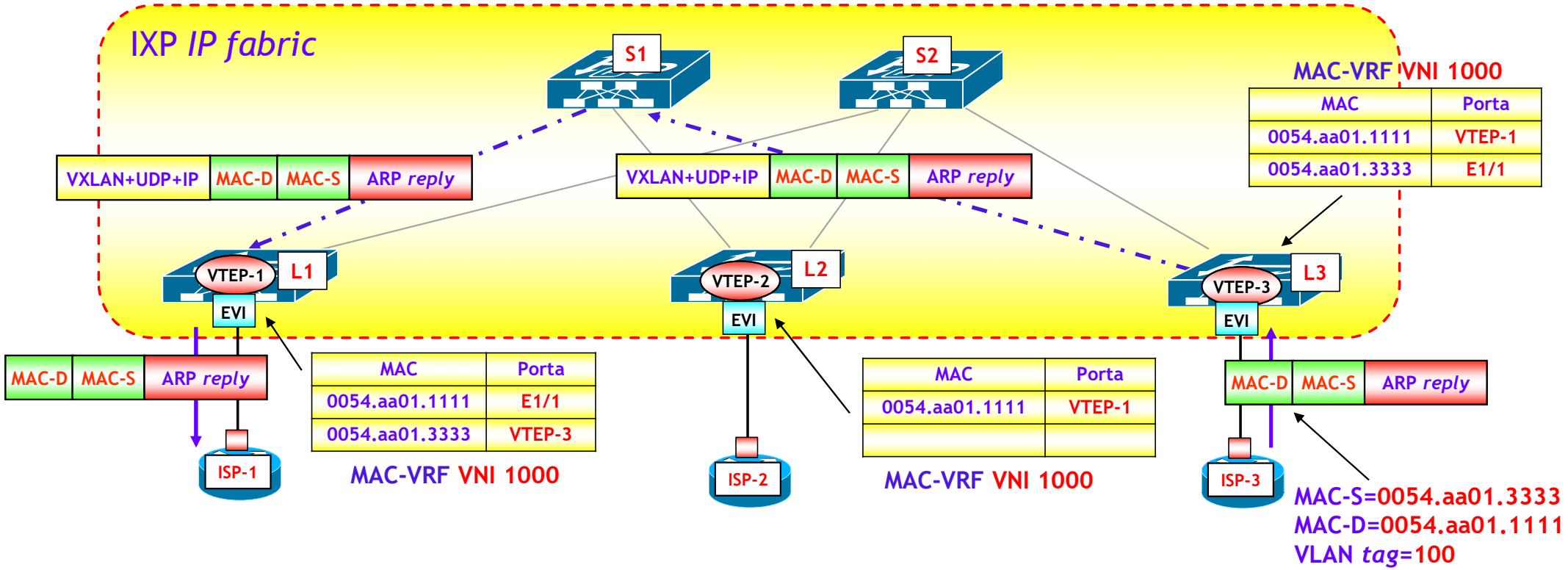
- Il router ISP-1 effettua un «**ping 10.1.1.3**»
  - Non conoscendo l'indirizzo MAC corrispondente a 10.1.1.3, invia un «*ARP request*»
  - Lo switch L1 apprende sul piano dati il MAC 0054.aa01.1111 e genera un annuncio BGP EVPN di tipo 2 per annunciarlo
  - Alla ricezione dell'annuncio BGP EVPN gli switch *Leaf* L1 e L2 aggiornano le loro MAC-VRF con l'entry MAC-to-VTEP

Esempio: *IP fabric* di un IXP (3/5)



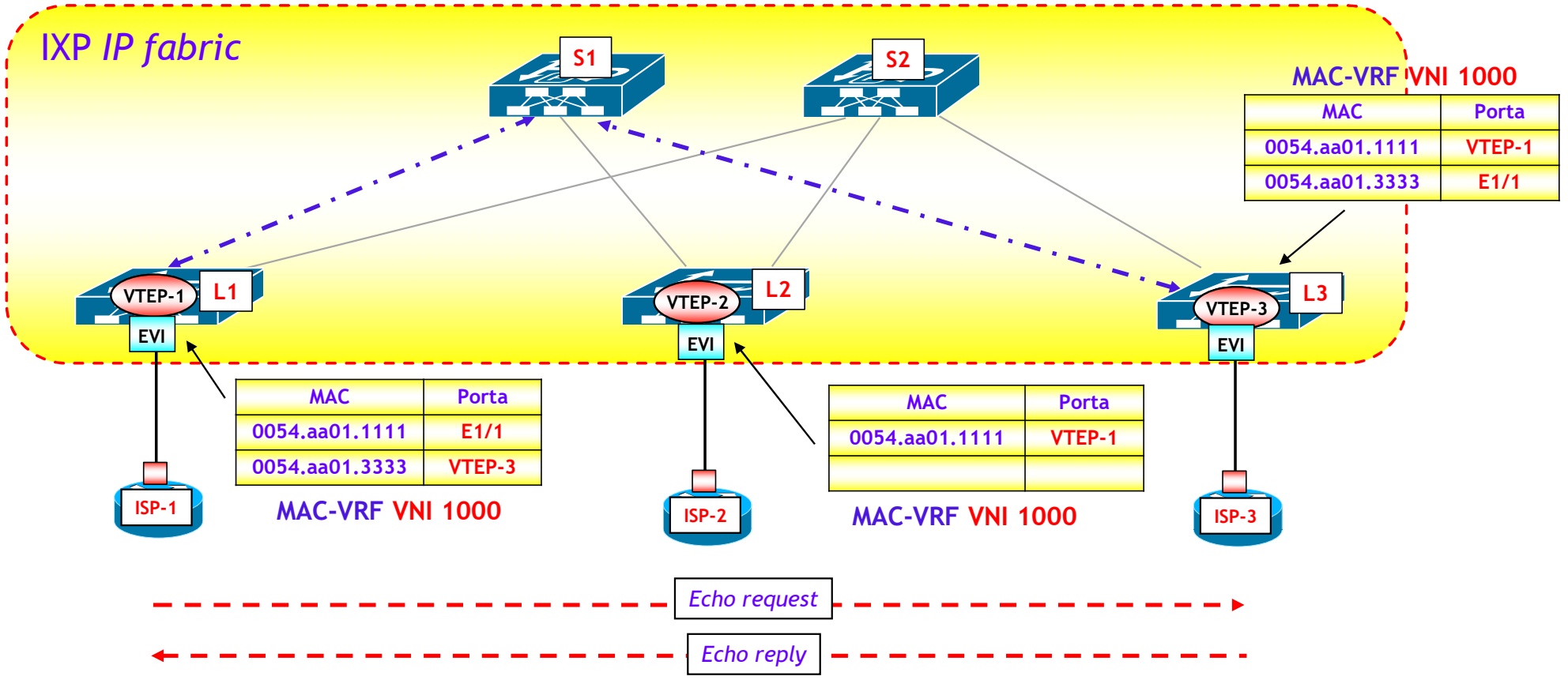
- Il messaggio *ARP request* raggiunge tutti gli altri CE dello stesso segmento VXLAN (VNI=1000) via *Head-end Replication* (*Ingress Replication*)

Esempio: *IP fabric* di un IXP (4/5)



- Il router ISP-3 risponde con un *ARP reply* con MAC destinazione unicast

Esempio: *IP fabric* di un IXP (5/5)



IV° modulo: Integrazione EVPN/VXLAN

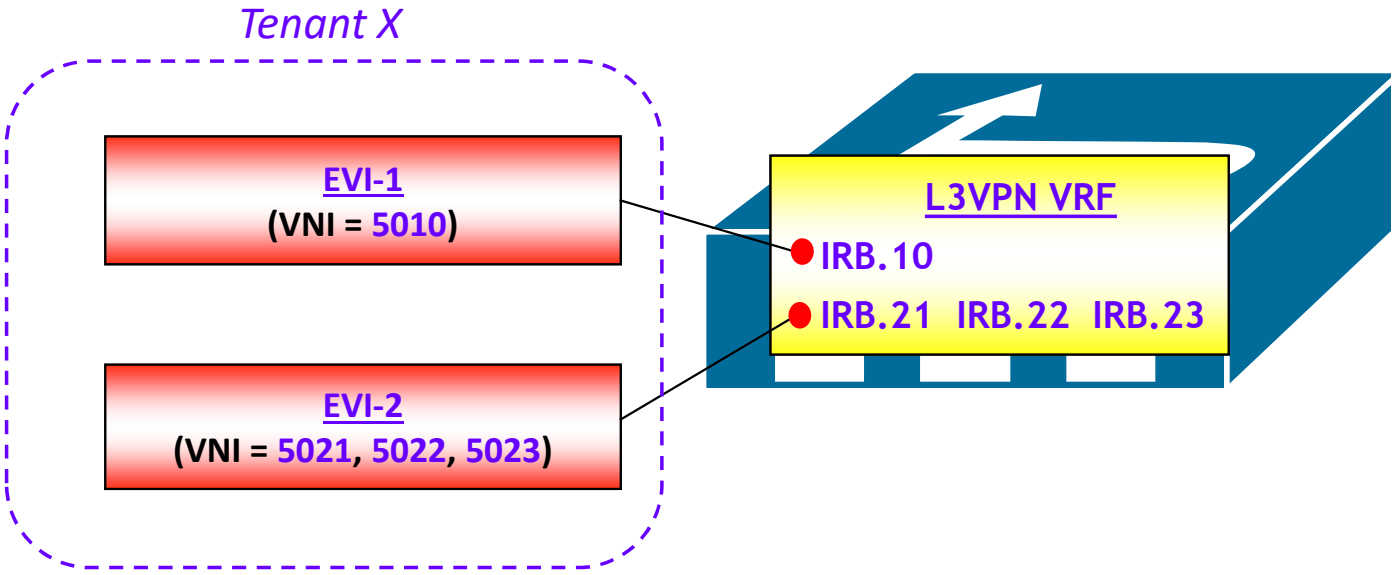
#1 Piano dati VXLAN con piano di controllo EVPN

#2 *Inter-VXLAN routing*

Copyright Tiziano Tofoni

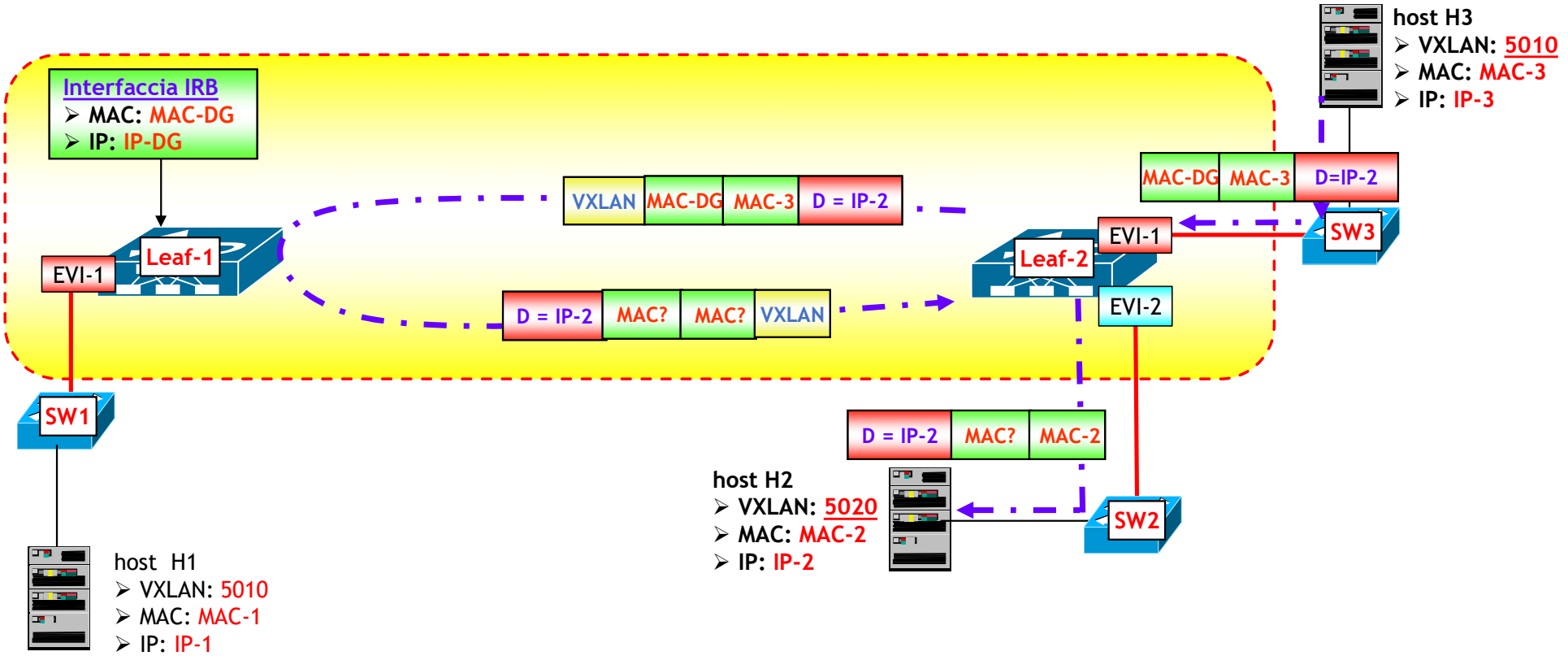
Integrazione *Routing* e *Bridging*

- Necessità di definire una interfaccia *IRB* che faccia da «ponte» tra le funzioni di Livello 2 e Livello 3
- Due funzioni fondamentali
  - Sincronizzazione delle associazioni <MAC; IP>: realizzata attraverso annunci BGP EVPN di tipo 2
  - Sincronizzazione dei *Default Gateway*



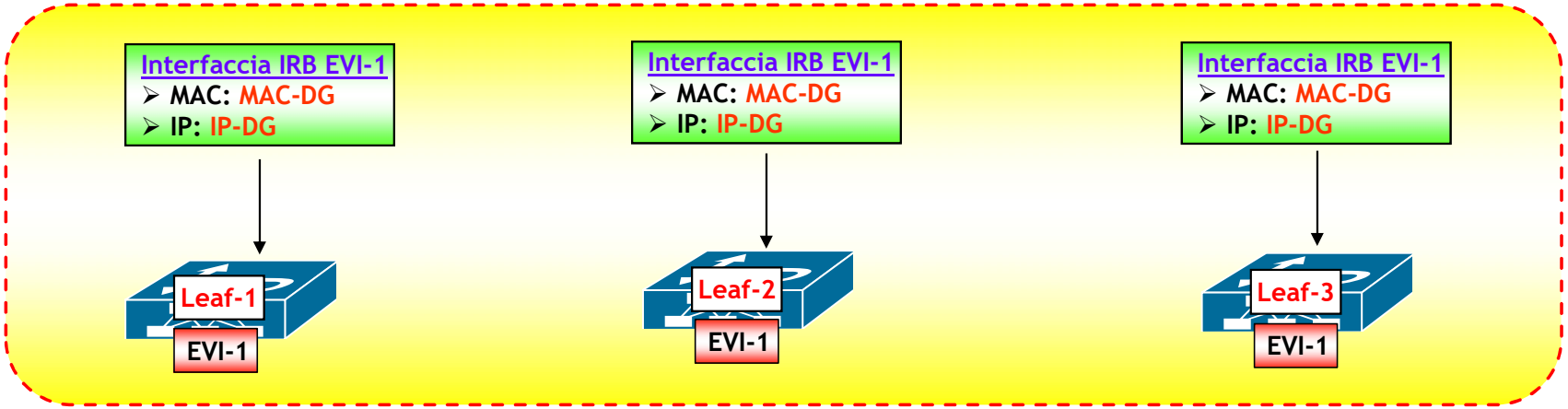
Sincronizzazione dei *default gateway* (1/2)

- Senza la sincronizzazione dei *default gateway*: l'effetto *trombooning* ...

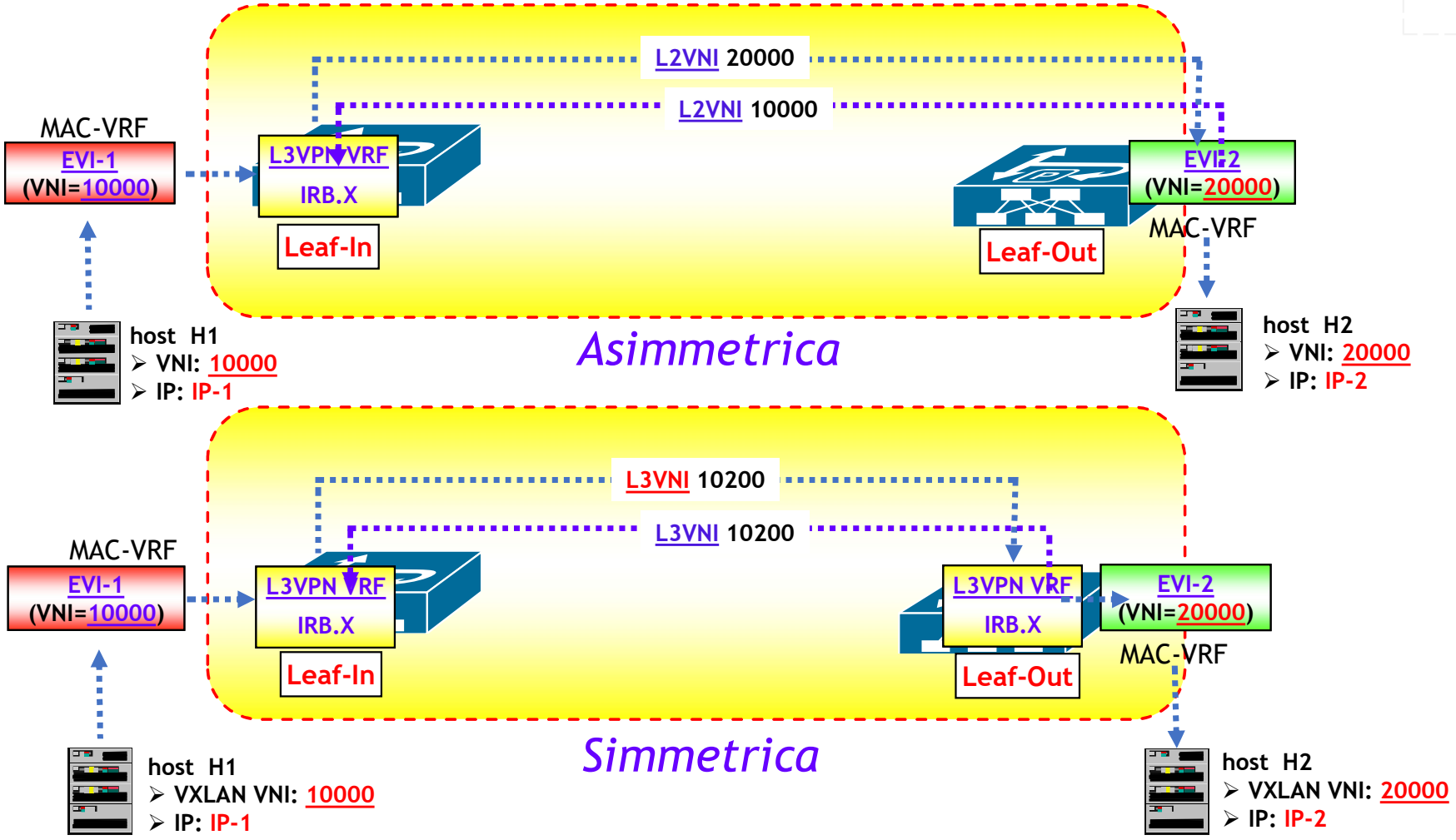


Sincronizzazione dei *default gateway* (2/2)

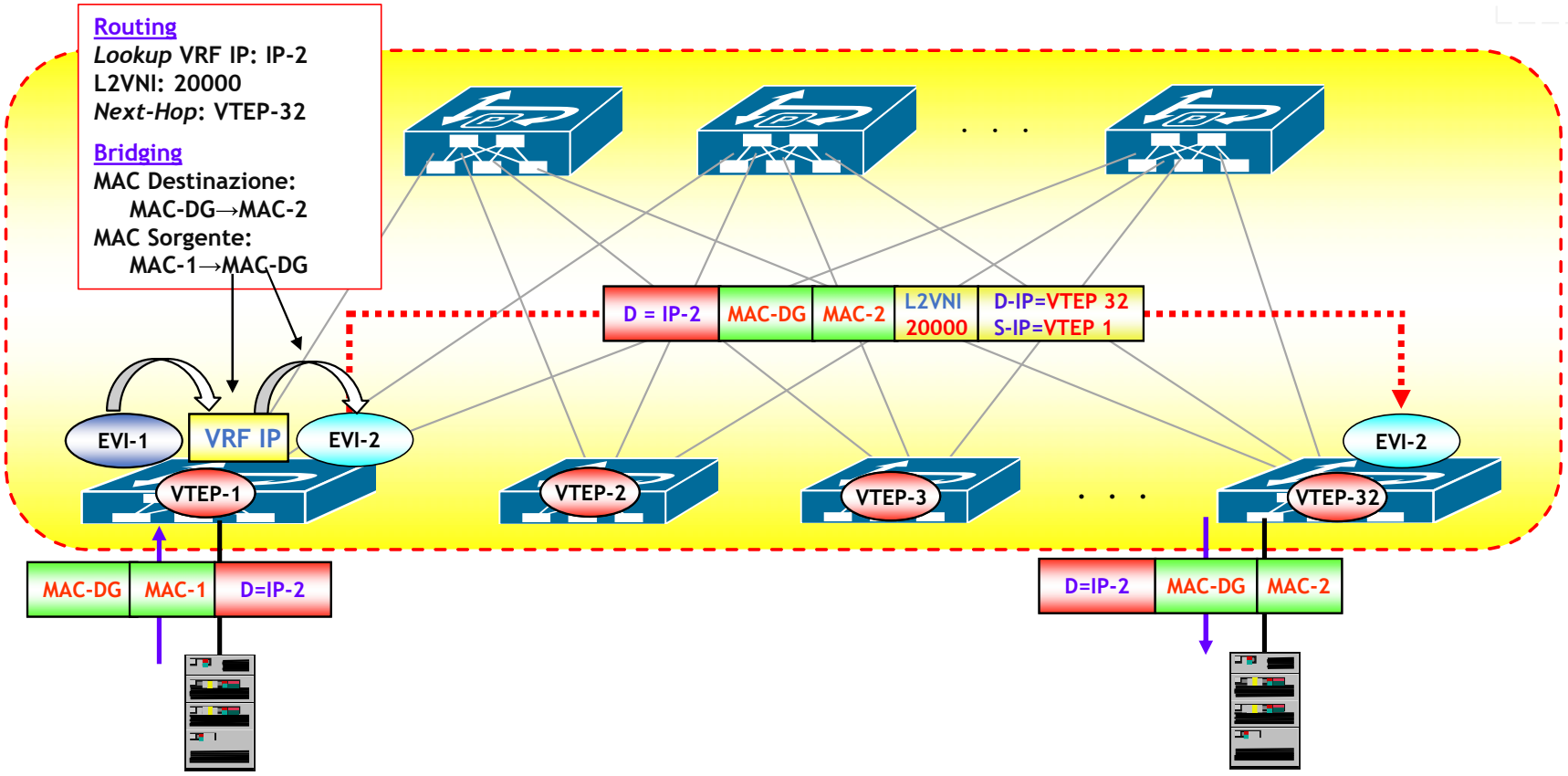
- Sincronizzazione dei *default gateway*: consente di distribuire su tutti gli switch *Leaf* che partecipano all'EVI, la funzione di *default gateway*
- Due modalità
  - Statica: si configura su ciascuno degli switch *Leaf* che partecipa all'EVI, per ciascun segmento VXLAN che fa parte dell'EVI, una interfaccia IRB con identico indirizzo MAC e identico indirizzo IP
  - Dinamica: si configurano MAC e IP del *default gateway* su un determinato switch *Leaf* e la sincronizzazione avviene attraverso annunci BGP EVPN di tipo 2, che contengono l'*Extended Community Default Gateway*.



Modalità di *forwarding*



Modalità asimmetrica



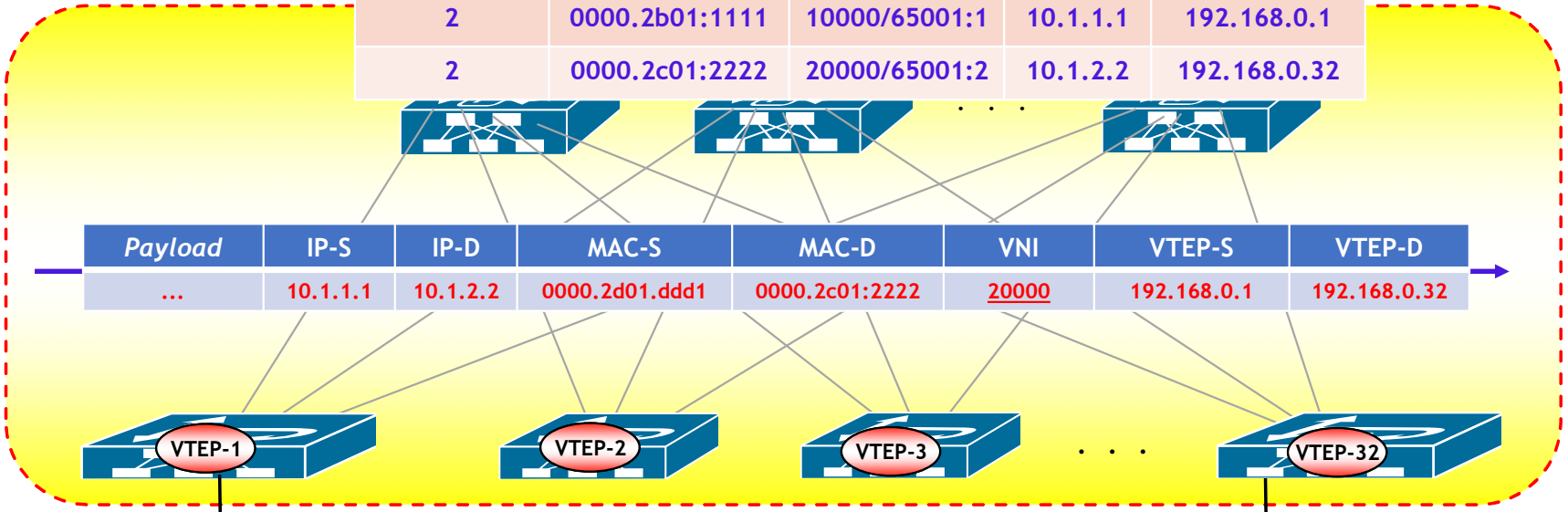
host H1
- MAC: MAC-1
- VLAN→VNI 10→10000
- IP: IP-1

host H2
- MAC: MAC-2
- VLAN→VNI 20→20000
- IP: IP-2

Modalità asimmetrica: esempio (1/2)

- *Host* H1→*Host* H2

BGP EVPN type	MAC	L2VNI/RT	IP	Next-Hop
2	0000.2b01:1111	10000/65001:1	10.1.1.1	192.168.0.1
2	0000.2c01:2222	20000/65001:2	10.1.2.2	192.168.0.32



MAC-D	MAC-S	VLAN-ID	IP-D	IP-S	Payload
0000.2d01:ddd1	0000.2b01:1111	10	10.1.2.2	10.1.1.1	....

MAC-DG

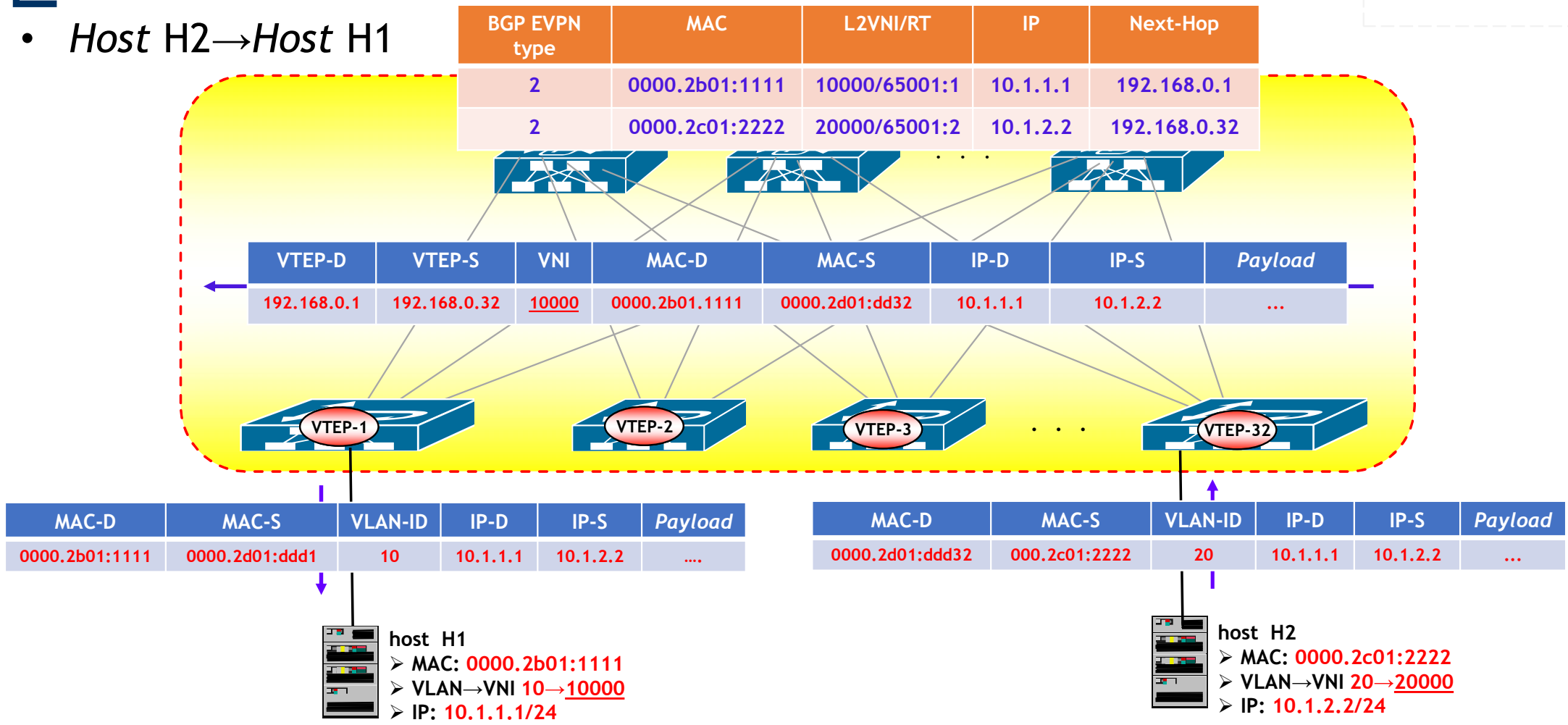
host H1
- MAC: 0000.2b01:1111
- VLAN→VNI 10→10000
- IP: 10.1.1.1/24

MAC-D	MAC-S	VLAN-ID	IP-D	IP-S	Payload
0000.2c01:2222	0000.2d01:ddd1	20	10.1.2.2	10.1.1.1	...

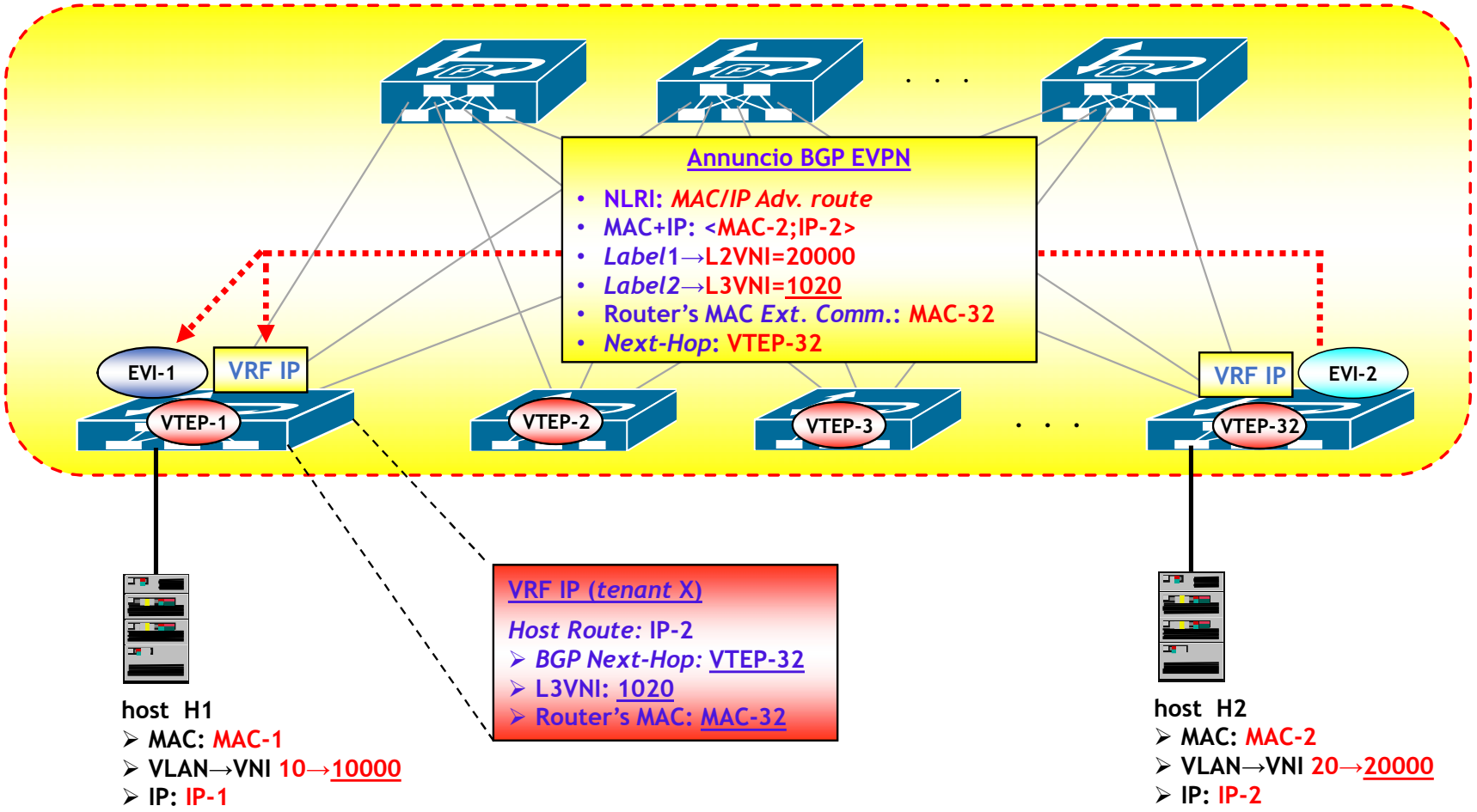
host H2
- MAC: 0000.2c01:2222
- VLAN→VNI 20→20000
- IP: 10.1.2.2/24

Modalità asimmetrica: esempio (2/2)

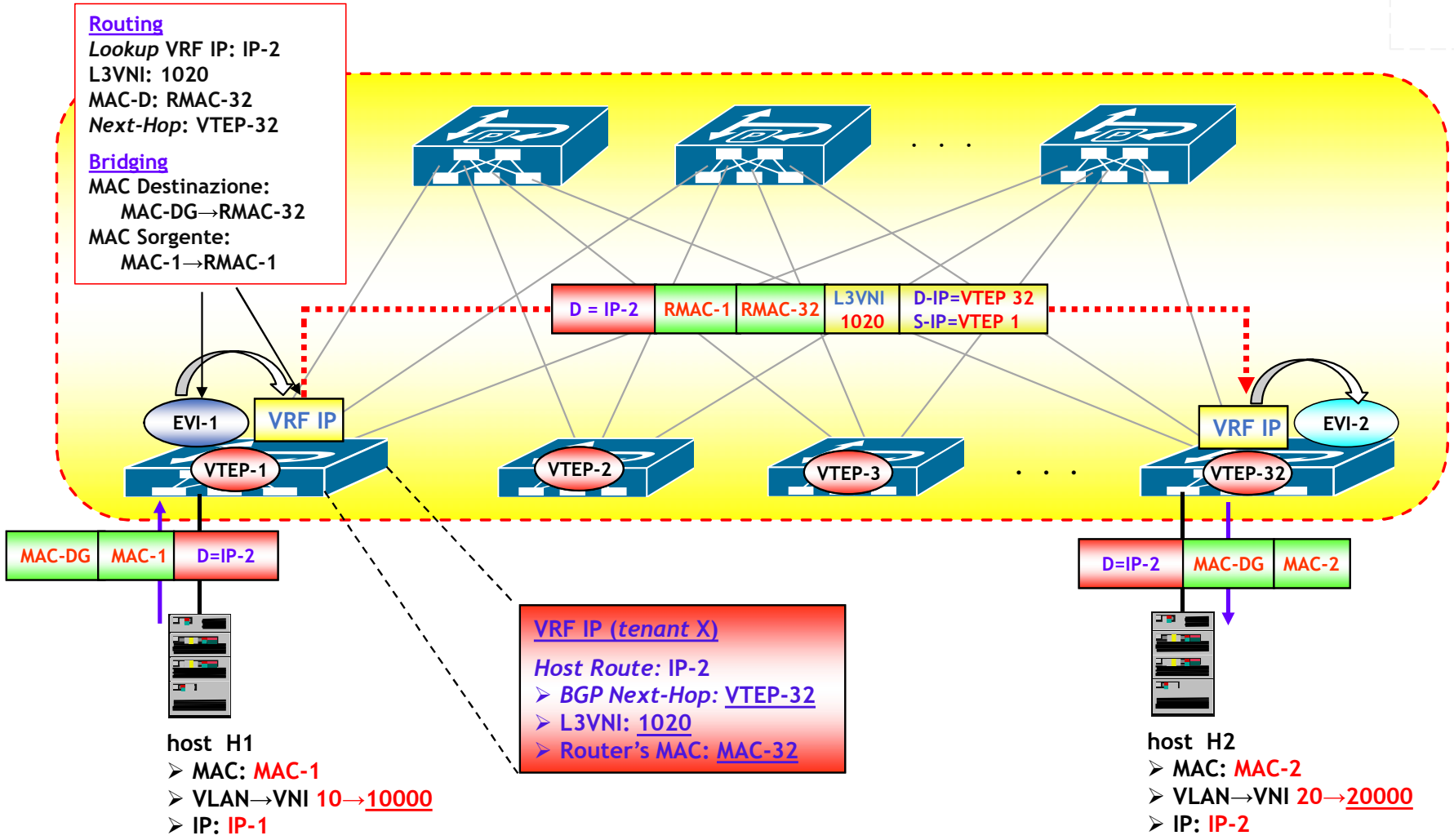
- *Host* H2→*Host* H1



Modalità simmetrica (1/2)

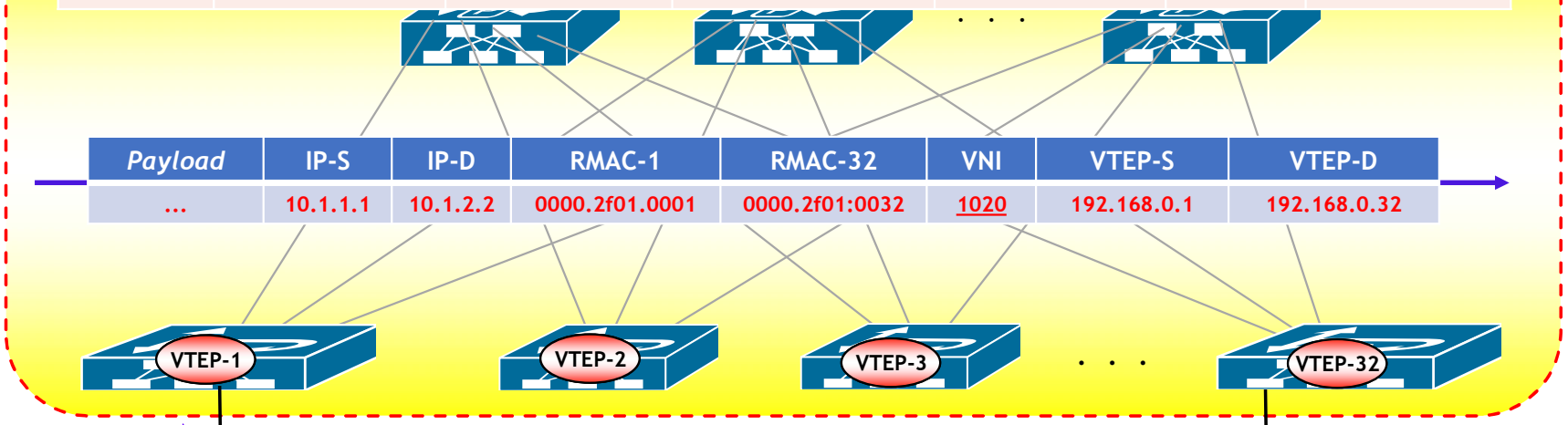


Modalità simmetrica (2/2)



Modalità simmetrica: esempio (1/2)

BGP EVPN type	MAC	L2VNI/RT	L3VNI/RT	RMAC	IP	Next-Hop
2	0000.2b01:1111	10000/65001:1	1020/65001:1020	0000.2f01.0001	10.1.1.1	192.168.0.1
2	0000.2c01:2222	20000/65001:2	1020/65001:1020	0000.2f01.0032	10.1.2.2	192.168.0.32



MAC-D	MAC-S	VLAN-ID	IP-D	IP-S	Payload
0000.2d01:ddd1	0000.2b01:1111	10	10.1.2.2	10.1.1.1	....

MAC-DG

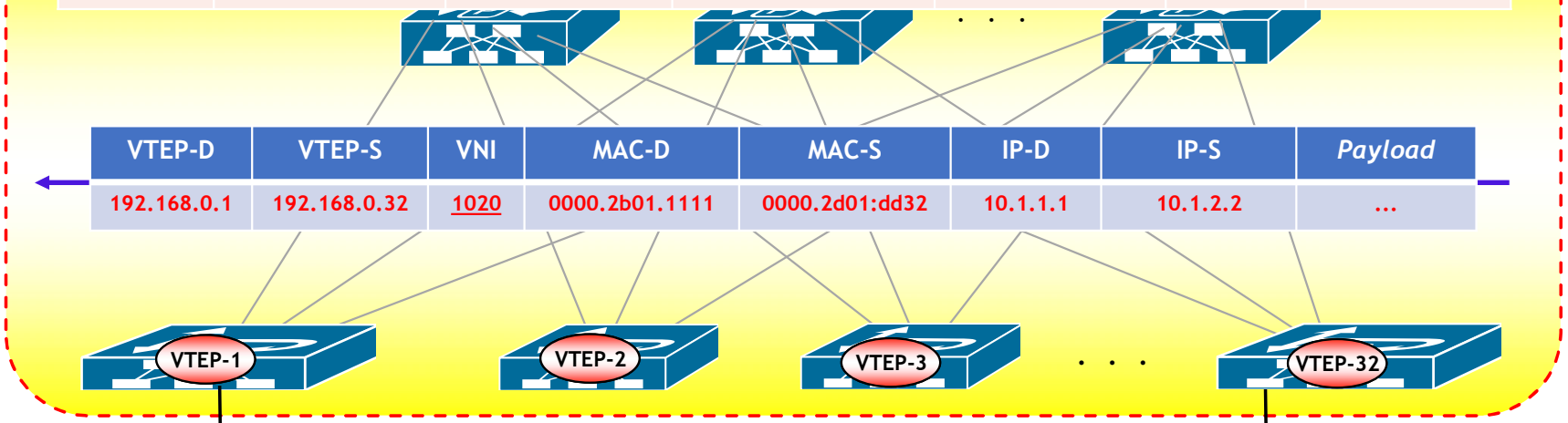
host H1
- MAC: 0000.2b01:1111
- VLAN→VNI 10→10000
- IP: 10.1.1.1/24

MAC-D	MAC-S	VLAN-ID	IP-D	IP-S	Payload
0000.2c01:2222	0000.2d01:dd32	20	10.1.2.2	10.1.1.1	...

host H2
- MAC: 0000.2c01:2222
- VLAN→VNI 20→20000
- IP: 10.1.2.2/24

Modalità simmetrica: esempio (2/2)

BGP EVPN type	MAC	L2VNI/RT	L3VNI/RT	RMAC	IP	Next-Hop
2	0000.2b01:1111	10000/65001:1	1020/65001:1020	0000.2f01.0001	10.1.1.1	192.168.0.1
2	0000.2c01:2222	20000/65001:2	1020/65001:1020	0000.2f01.0032	10.1.2.2	192.168.0.32



MAC-D	MAC-S	VLAN-ID	IP-D	IP-S	Payload
0000.2b01:1111	0000.2d01:ddd1	10	10.1.1.1	10.1.2.2	....

host H1
- MAC: 0000.2b01:1111
- VLAN→VNI 10→10000
- IP: 10.1.1.1/24

MAC-D	MAC-S	VLAN-ID	IP-D	IP-S	Payload
0000.2c01:1111	0000.2d01:dd32	20	10.1.1.1	10.1.2.2	...

host H2
- MAC: 0000.2c01:2222
- VLAN→VNI 20→20000
- IP: 10.1.2.2/24

V° modulo

Aspetti architetturali

Copyright Tiziano Tofoni

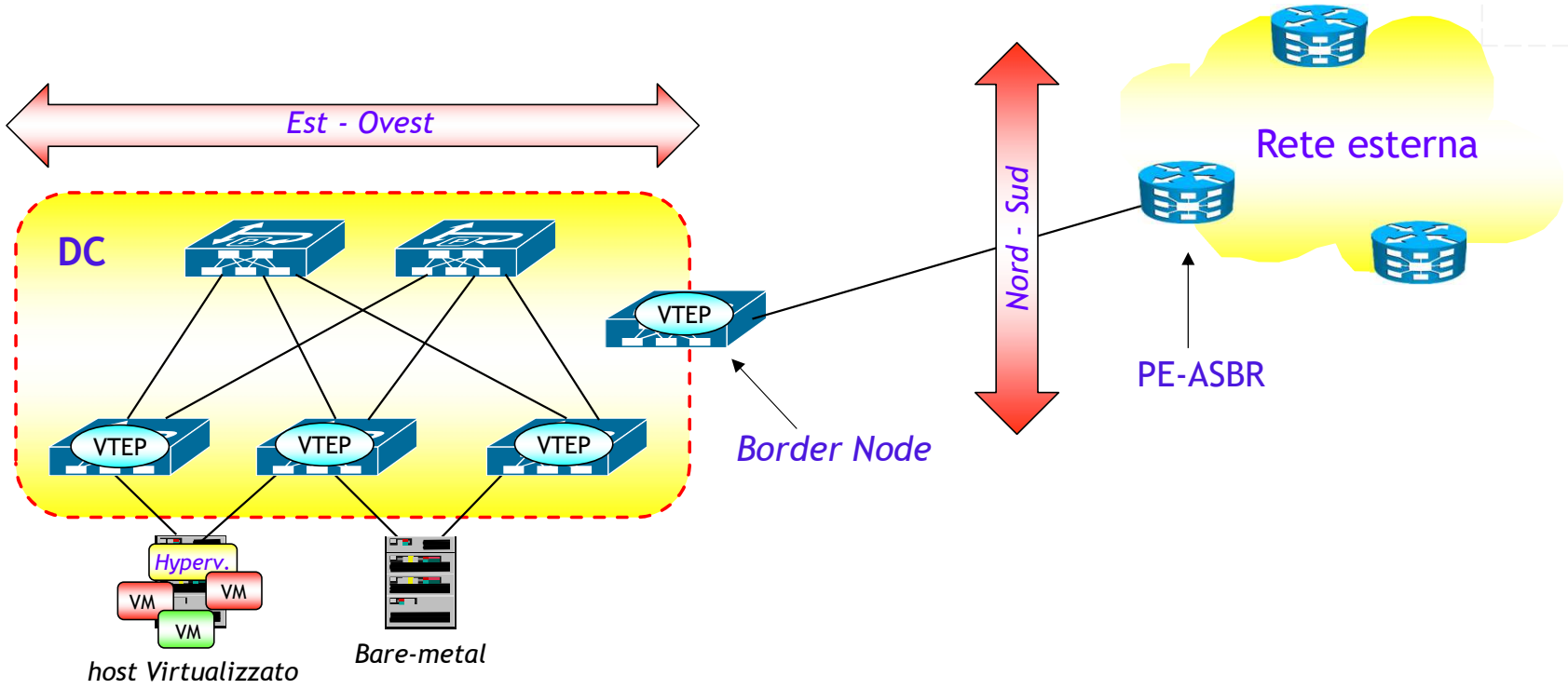
V° modulo: Aspetti architetturali

#1 Interconnessione DC con reti esterne

#2 Architetture *multi-POD* e *multi-site*

Copyright Tiziano Tofoni

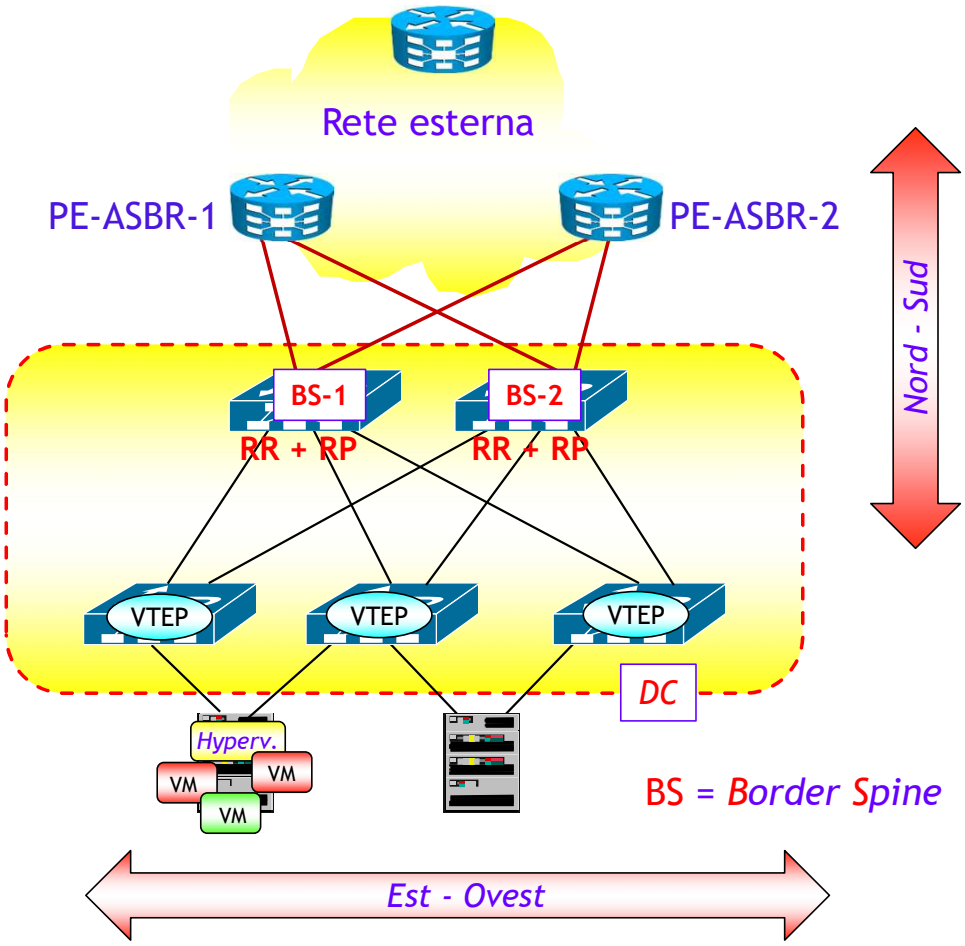
Architetture di Interconnessione



- Il *Border Node* (BN) è l'elemento di interconnessione con le reti esterne
  - Tipicamente ridondato
  - Funzione integrata negli *Spine* oppure in un *Leaf* separato (*Border leaf*)

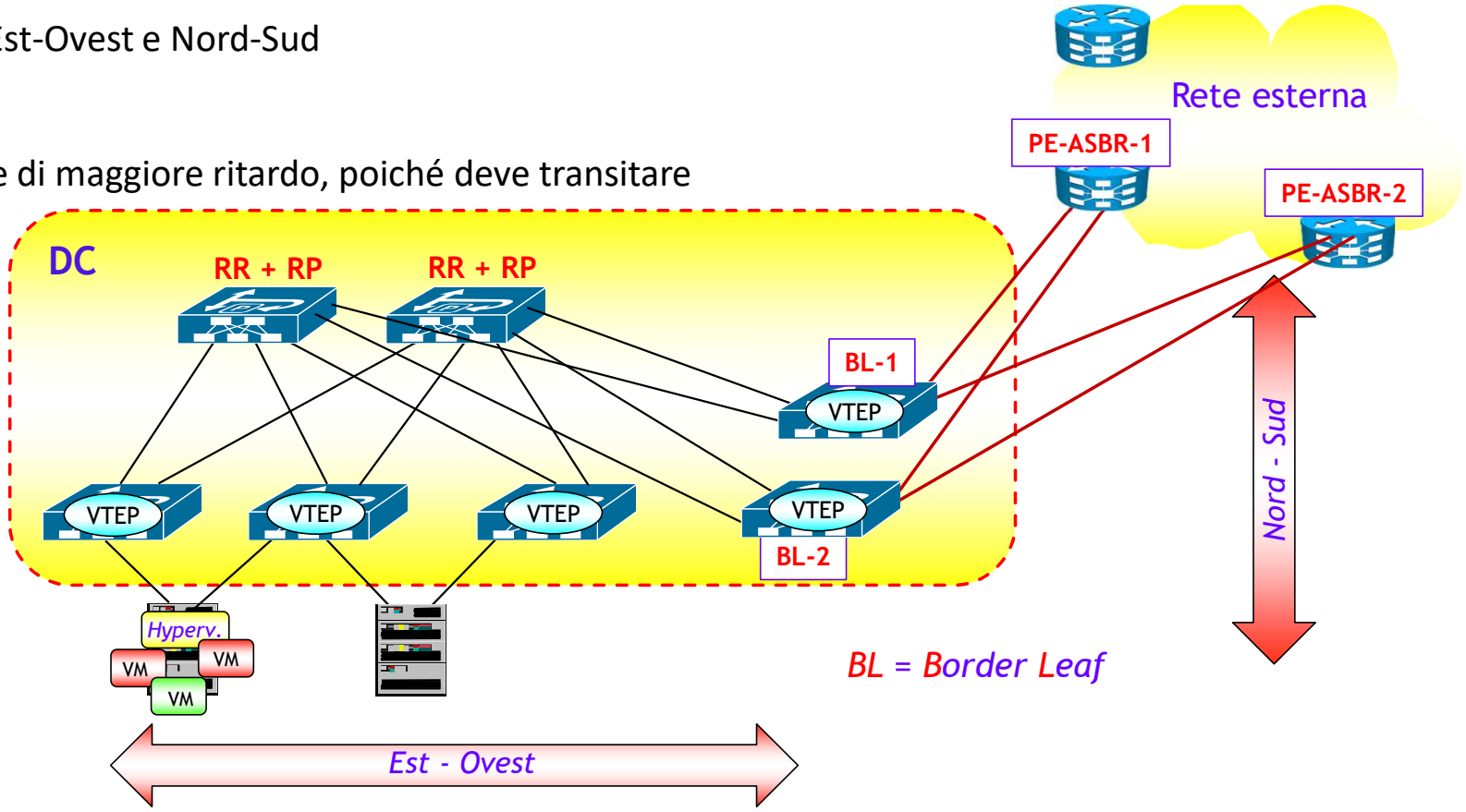
Border Node negli switch *Spine*

- Pro
  - Minore ritardo *end-to-end* per il traffico Nord-Sud
  - Vantaggioso quando il traffico Nord-Sud è preponderante rispetto al traffico Est-Ovest
- Contro
  - Gli *Spine* devono avere la funzione VTEP
  - Potenziale sovraccarico elaborativo a causa di altre funzioni che normalmente gli vengono assegnate (es. RR, RP)
  - In funzione del traffico Nord-Sud due o più interfacce aggiuntive per l'interconnessione verso l'esterno

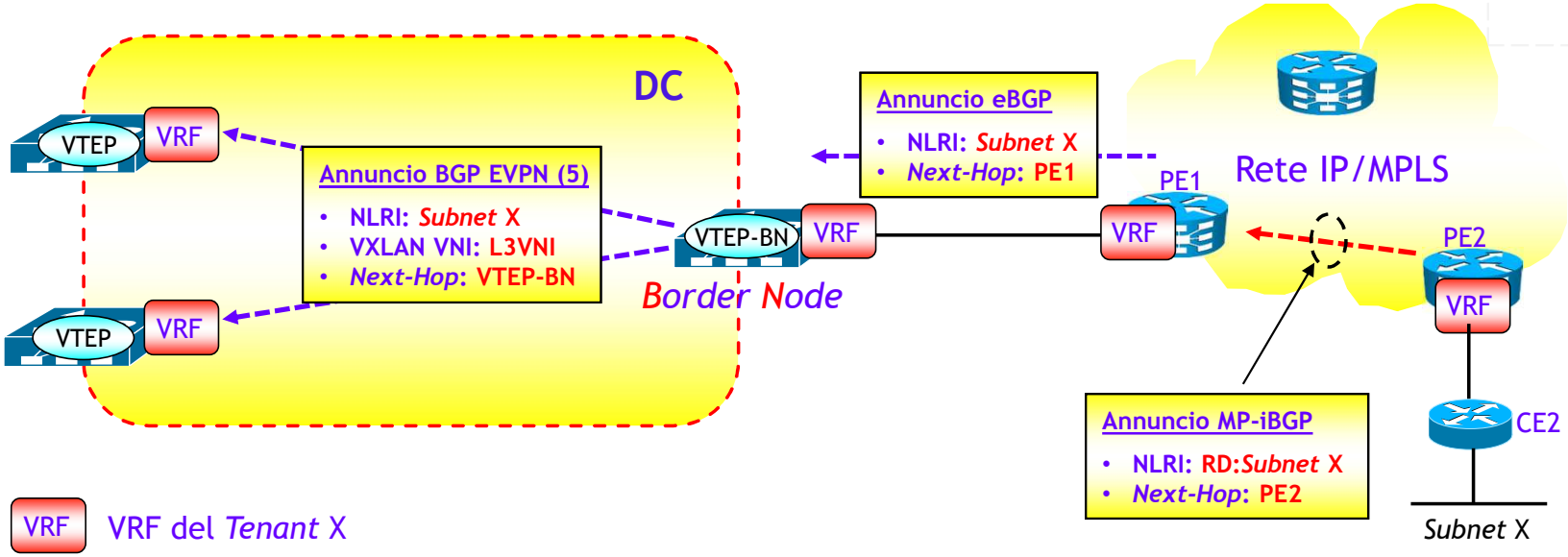


Border Node negli switch *Leaf*

- Pro
  - Maggiore scalabilità e banda disponibile,
  - Separazione tra traffico Est-Ovest e Nord-Sud
- Contro
  - Il traffico Nord-Sud soffre di maggiore ritardo, poiché deve transitare su un nodo in più
  - Soluzione più costosa

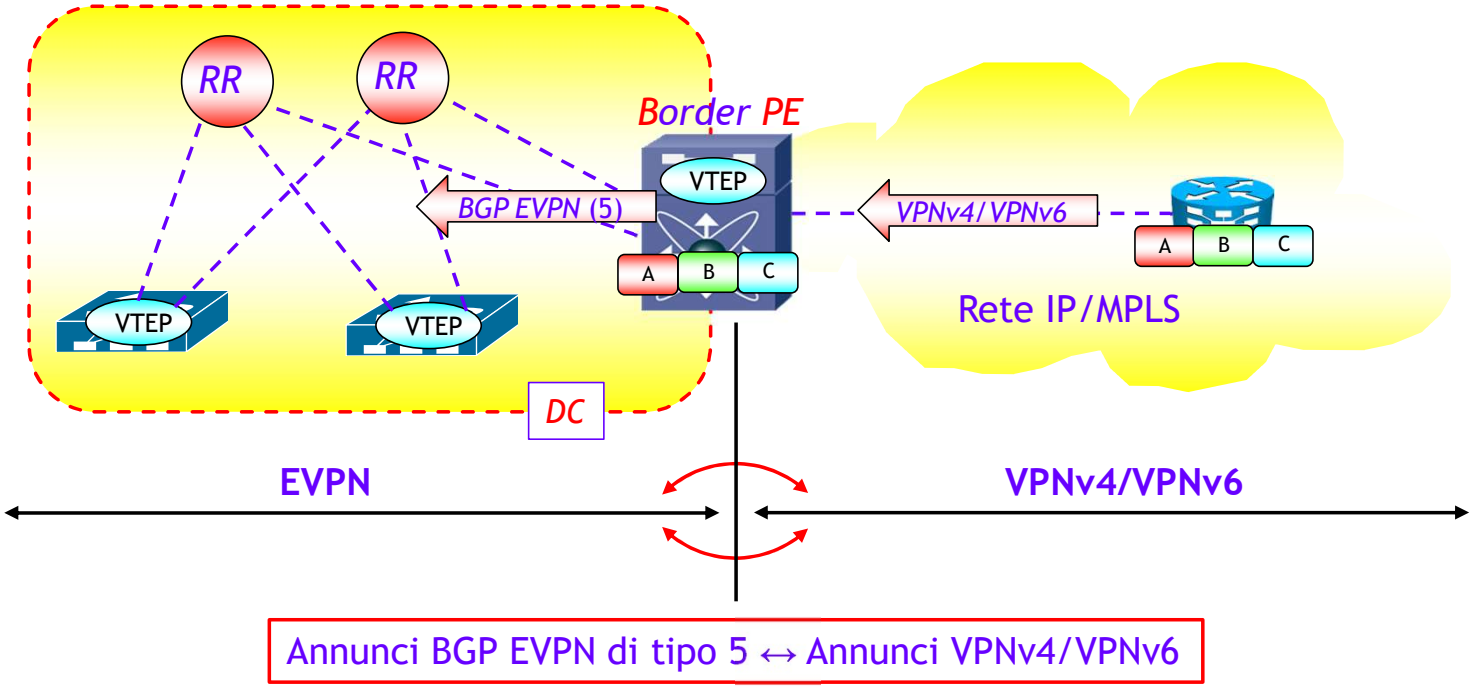


Modello *VRF-Lite* (*Inter-AS Option A*)



- Simile al modello «*Option A*» del servizio L3VPN *Inter-AS*
  - Tra BN e PE-ASBR della rete esterna viene attivato un collegamento associato a due VRF
  - Il BN vede il PE-ASBR come un CE e viceversa, permettendo lo scambio dei prefissi inter-sito utilizzando le regole classiche del routing IPv4
  - Soluzione molto semplice ma poco scalabile
    - Adatta per *Data Center* di piccole dimensioni con pochi *tenant*

Modello *L3VPN Hand-off*



- Il *Border PE* integra le funzioni di PE L3VPN e di *Border Node*
  - Gli annunci VPNv4/VPNv6 ricevuti dalla rete IP/MPLS vengono trasformati in annunci BGP EVPN di tipo 5 e viceversa

V° modulo: Aspetti architetturali

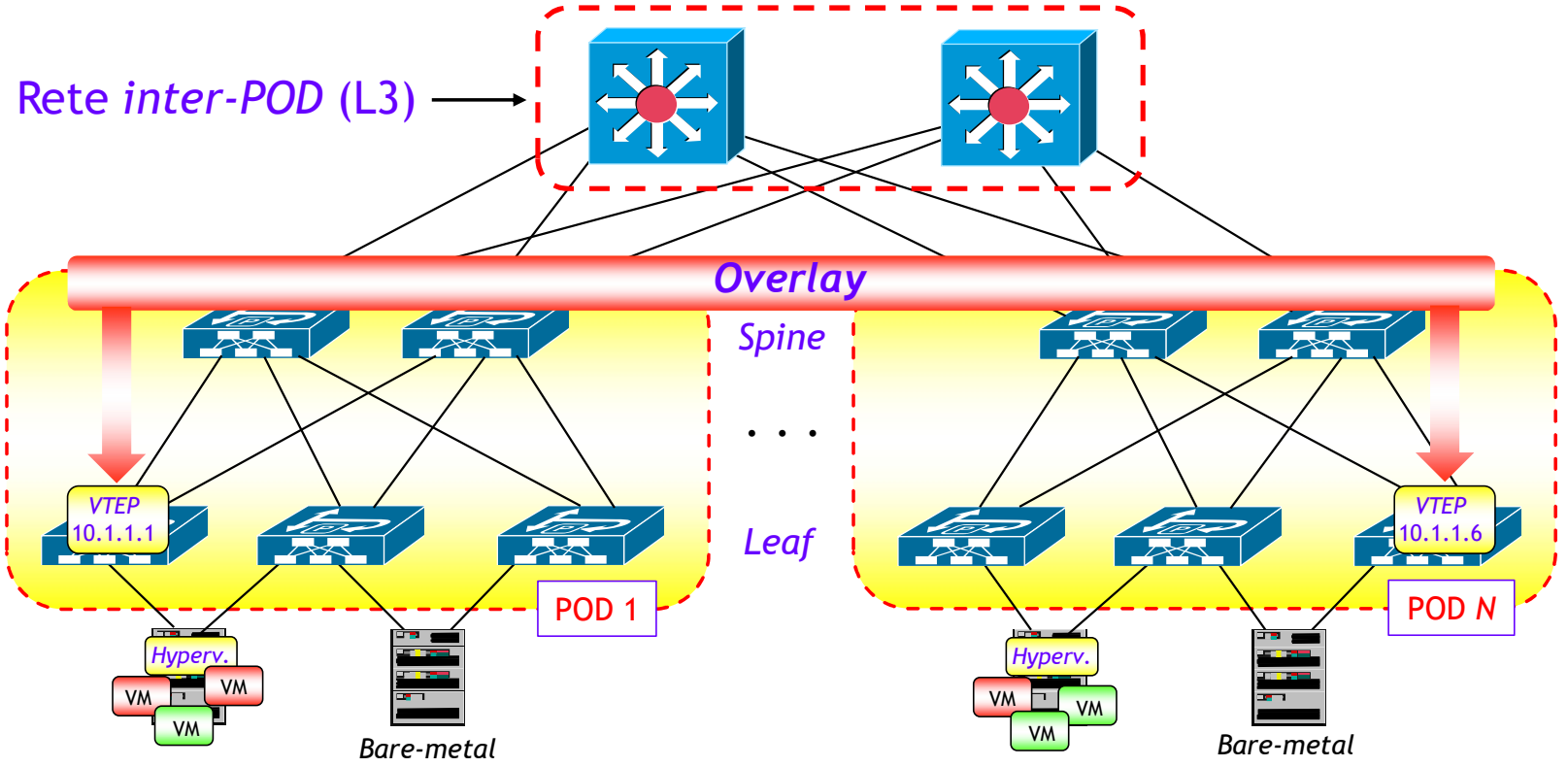
#1 Interconnessione DC con reti esterne

#2 Architetture *multi-POD* e *multi-site*

Copyright Tiziano Tofoni

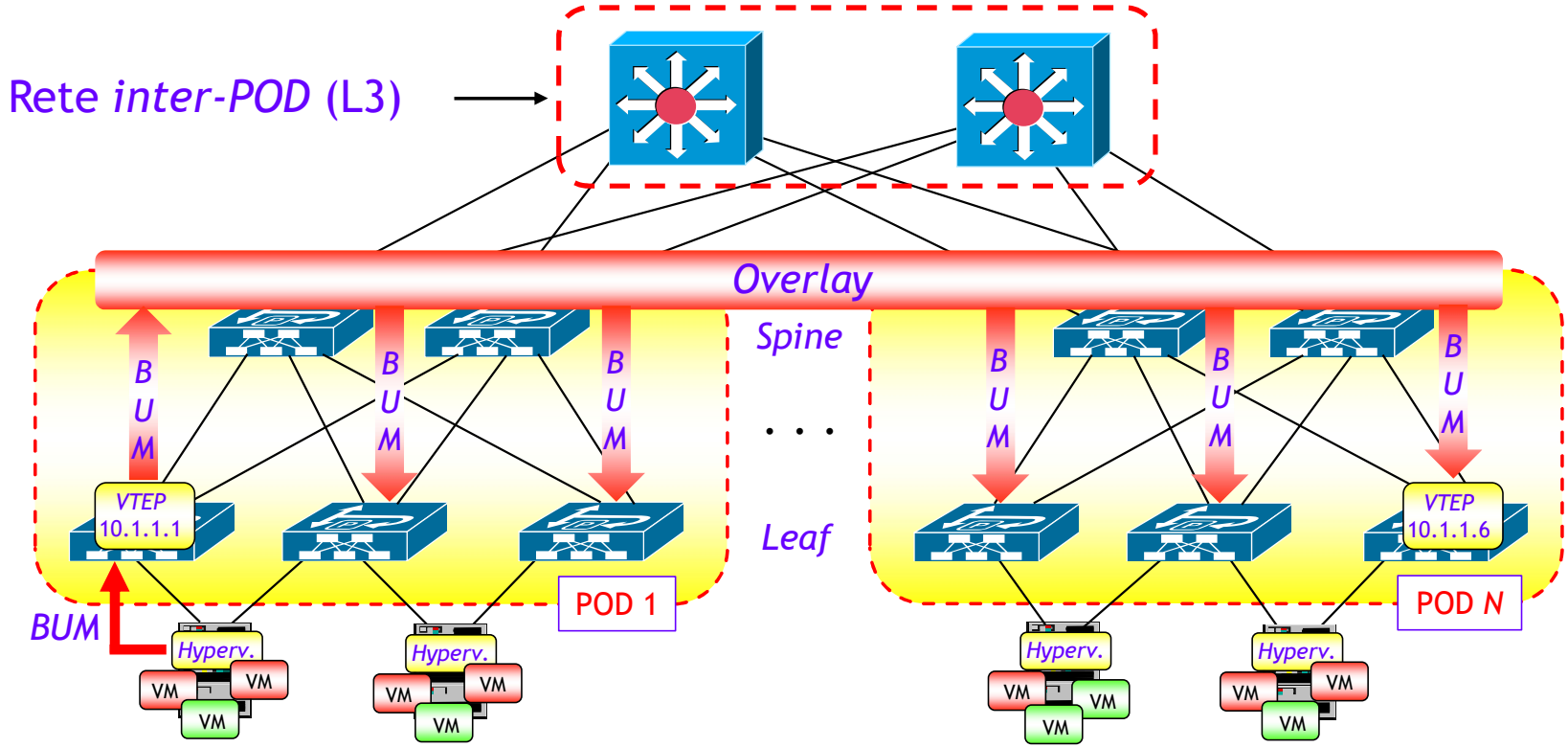
Architetture *multi-POD*

- Architettura di tipo *flat*: un segmento VXLAN vede la topologia *multi-POD* come fosse un singolo dominio *Overlay*
  - L'incapsulamento VXLAN avviene *end-to-end*

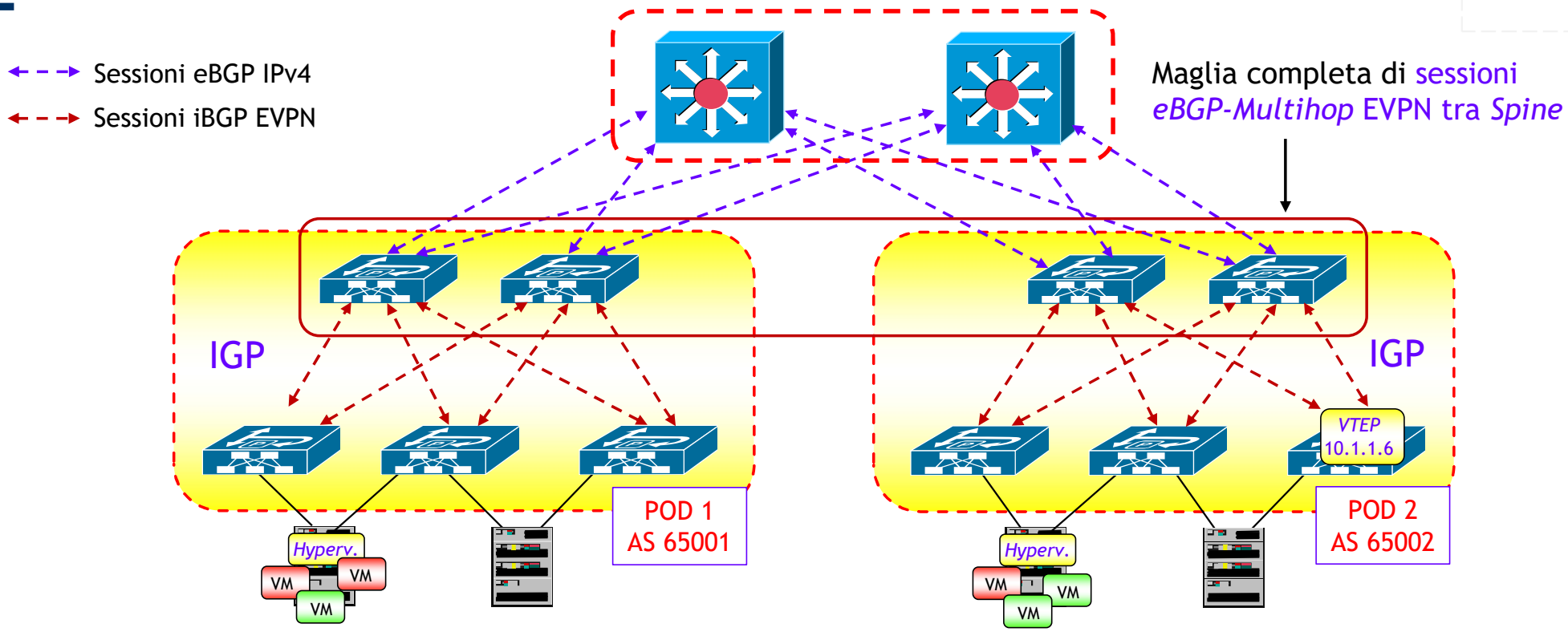


Architetture *multi-POD*: traffico BUM

- Singolo dominio di *broadcast*
  - Una trama BUM inviata da un host di un segmento VXLAN viene replicata (via routing multicast o via *head-end replication*) su tutte le VTEP che hanno almeno un *host* dello stesso segmento

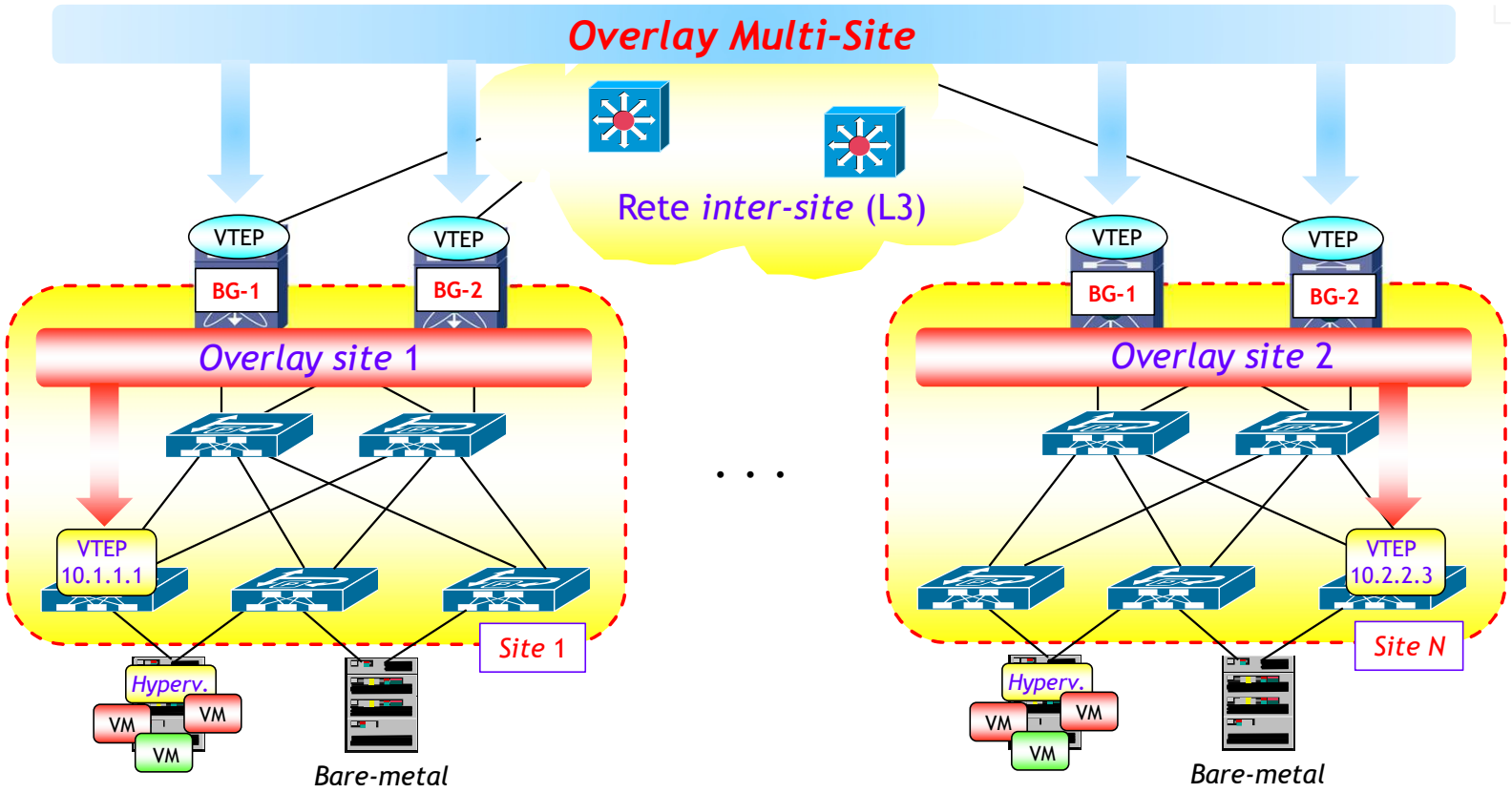


Architetture *multi-POD*: piano di controllo



- NOTA: le sessioni *eBGP-multihop* EVPN devono essere configurate in modo da non cambiare il *BGP Next-Hop*
  - Simile al modello L3VPN *inter-AS Option C*

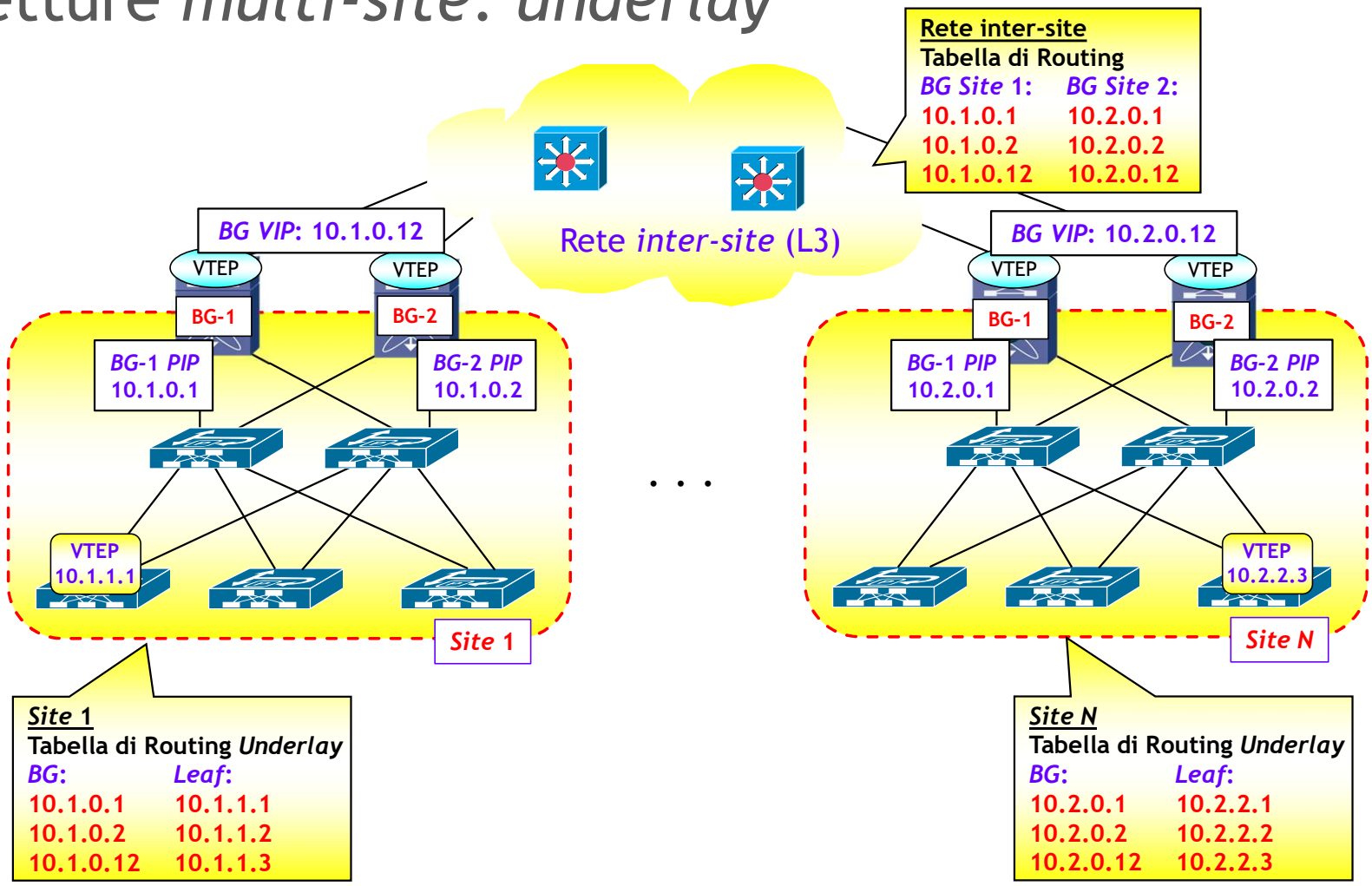
Architetture *multi-site*



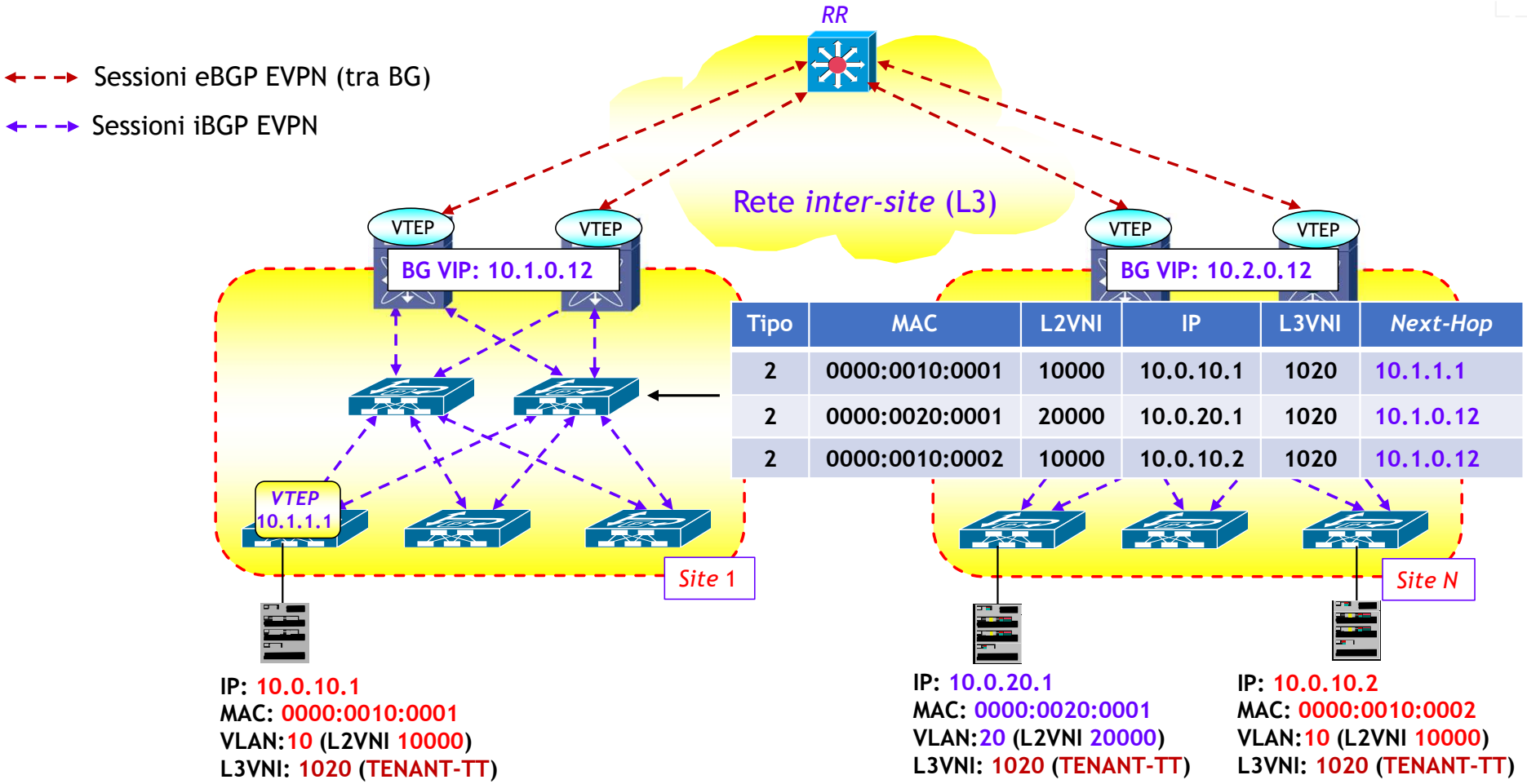
- Caratteristica: *underlay* isolati, *overlay* gerarchici
  - Simile al modello L3VPN *inter-AS Option B*

Border Gateway

Architetture *multi-site*: *underlay*

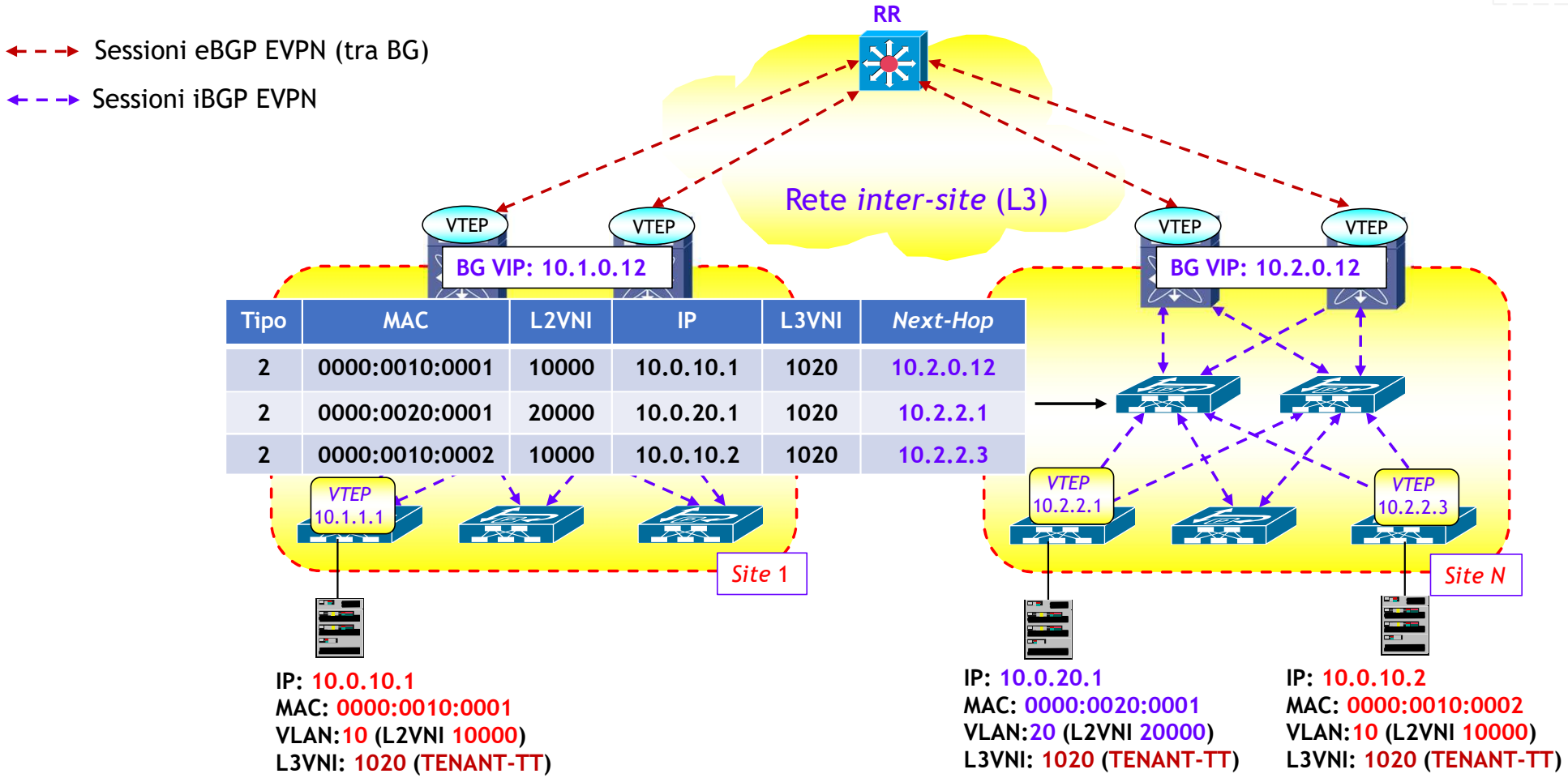


Architetture *multi-site*: *overlay* (1/3)

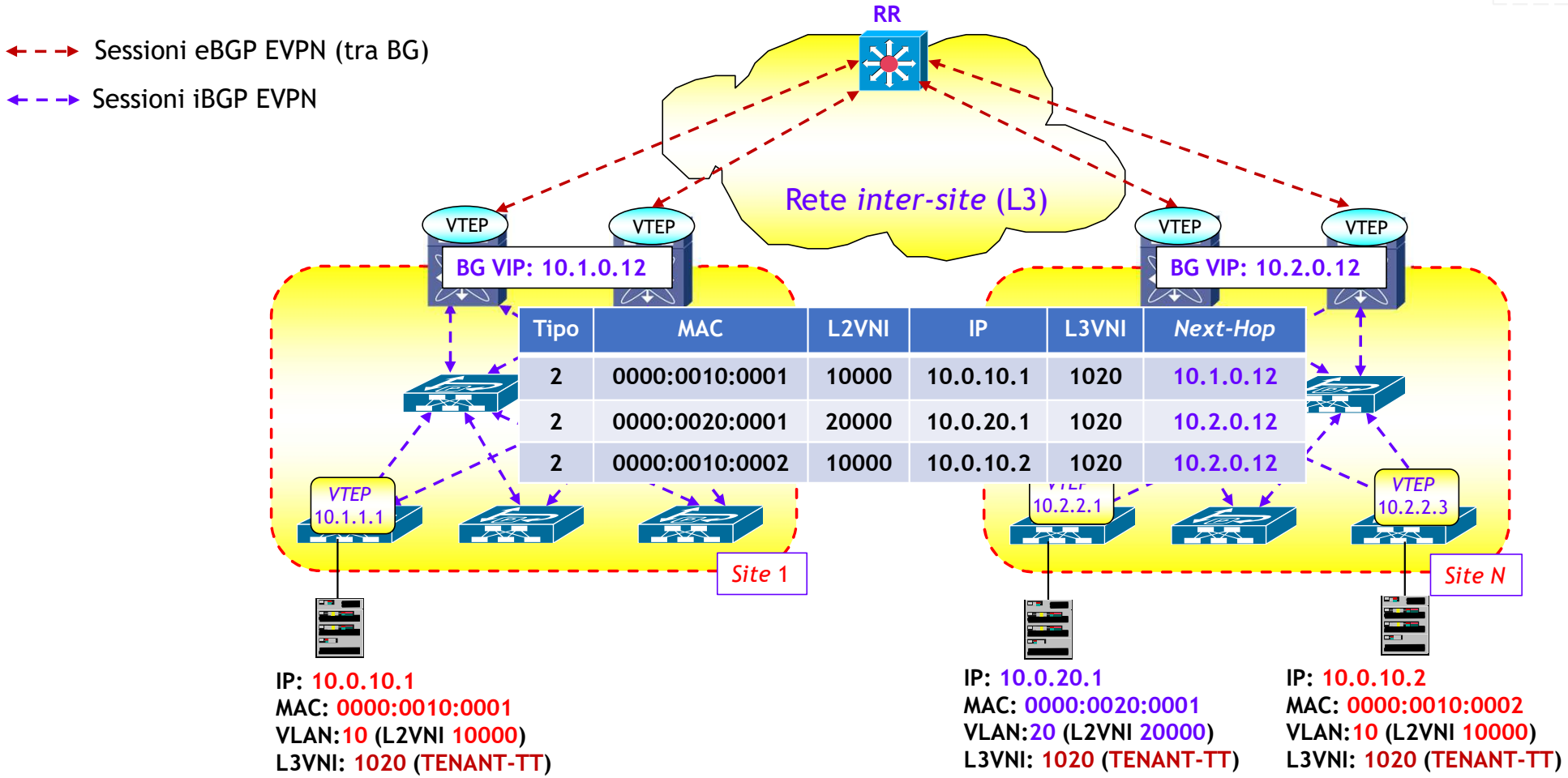


Tipo	MAC	L2VNI	IP	L3VNI	*Next-Hop*
2	0000:0010:0001	10000	10.0.10.1	1020	10.1.1.1
2	0000:0020:0001	20000	10.0.20.1	1020	10.1.0.12
2	0000:0010:0002	10000	10.0.10.2	1020	10.1.0.12

Architetture *multi-site*: *overlay* (2/3)



Architetture *multi-site*: *overlay* (3/3)



VI° modulo

DC per l'AI

Copyright Tiziano Tofoni

VI° modulo: DC per l'AI

#1	*AI cluster*
#2	Controllo della congestione: ECN e WRED
#3	*Lossless Ethernet*: *Priority Flow Control*
#4	PFC+ECN=DCQCN

Copyright Tiziano Tofoni

Cosa è un *AI cluster*?

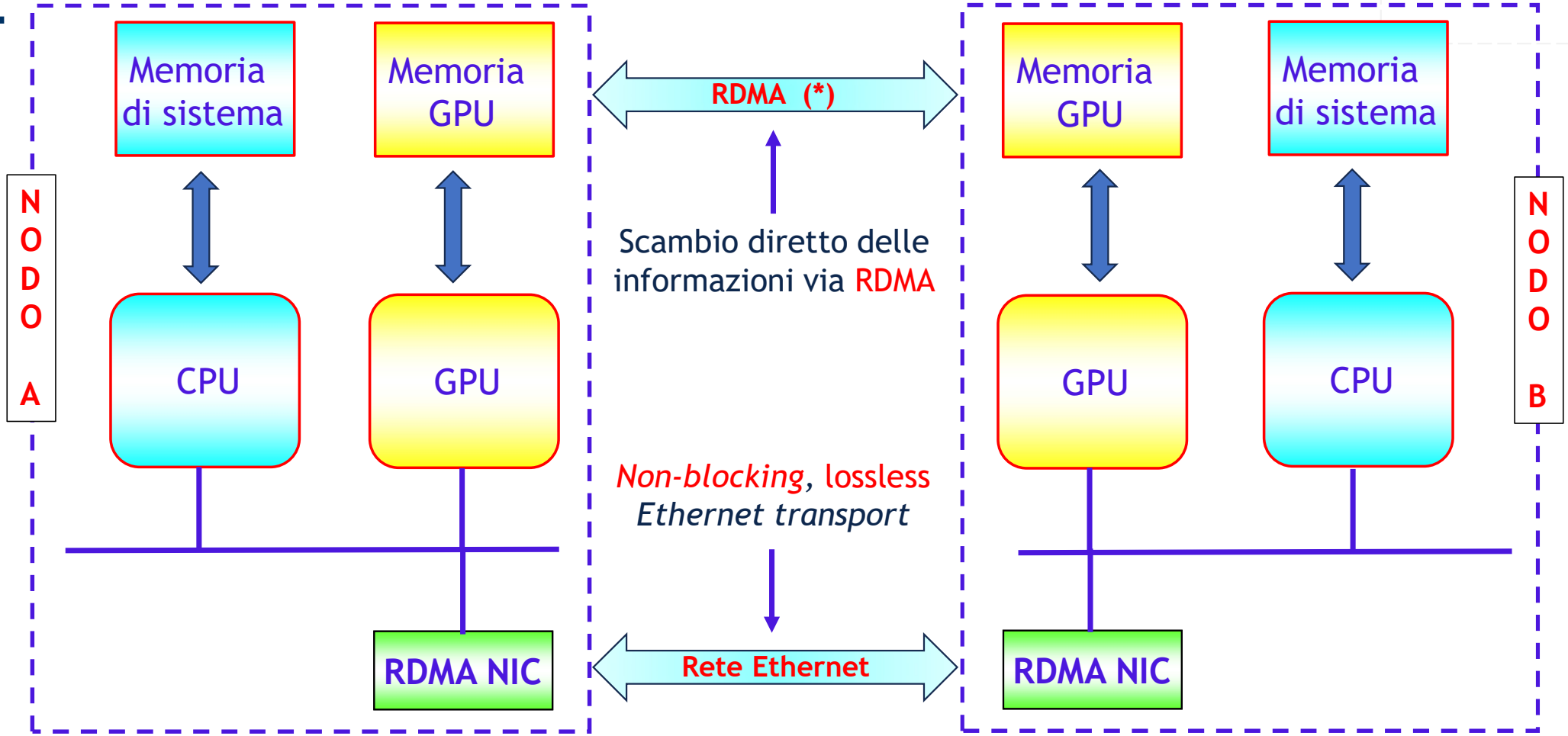
- Un *AI cluster* è un sistema di calcolo composto da più nodi (computer) interconnessi che lavorano insieme per eseguire carichi di lavoro intensivi di intelligenza artificiale (AI) e *machine learning* (ML)
- Progettati specificamente per accelerare i processi di *training* di modelli complessi e altre attività ad alta intensità computazionale tipiche dell'AI
- Caratteristiche principali di un *AI cluster*
  - Nodi di calcolo ad alte prestazioni (GPU, Acceleratori AI specializzati (es. TPU - Tensor Processing Units), CPU molto potenti
  - Interconnessione ad alta velocità e bassa latenza
  - Storage ad alte prestazioni
  - Software e framework ottimizzati per l'AI
  - Scalabilità

A cosa serve un *AI cluster*?

- Accelerare e rendere possibile l'esecuzione di compiti di AI complessi che sarebbero proibitivi su una singola macchina o su un'infrastruttura meno potente

- Esempi:
  - Training di modelli di *deep learning* di grandi dimensioni
  - Inferenza su larga scala
  - Esplorazione e analisi di grandi dataset
  - Ricerca e sviluppo in AI
  - Applicazioni AI avanzate

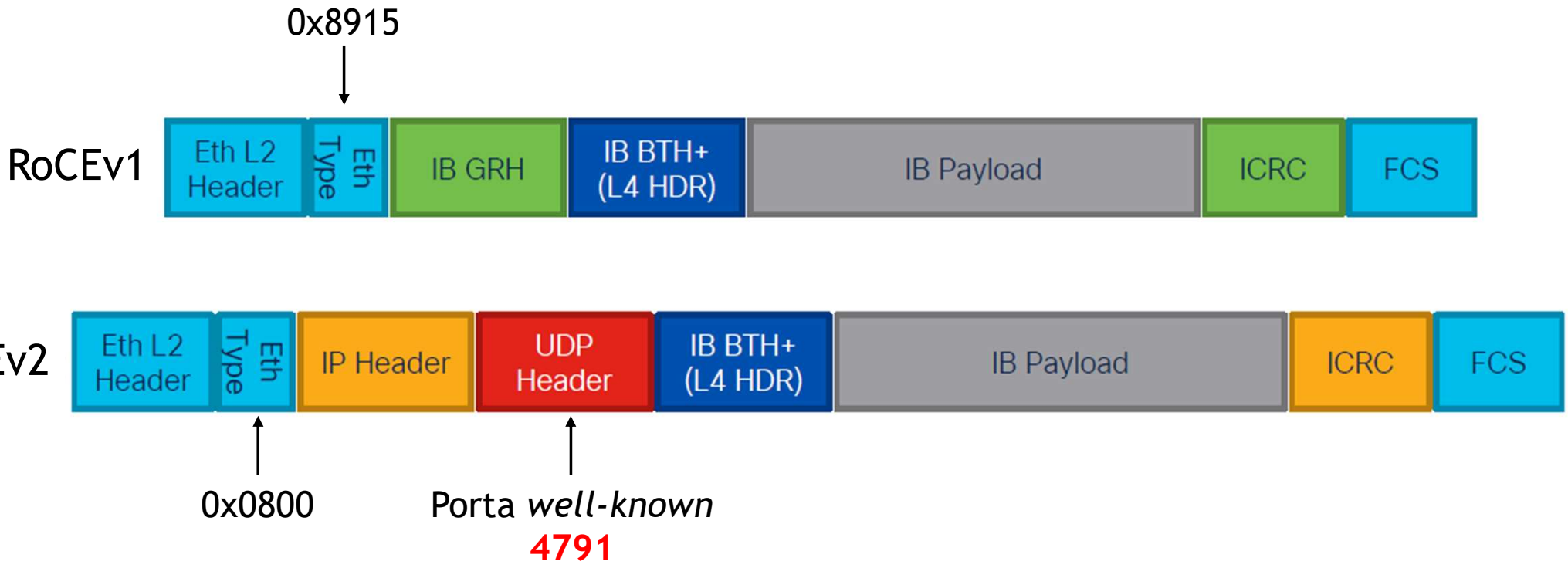
Comunicazione tra nodi



(*) **RDMA**= ***R****emote* ***D****irect* ***M****emory* ***A****ccess*

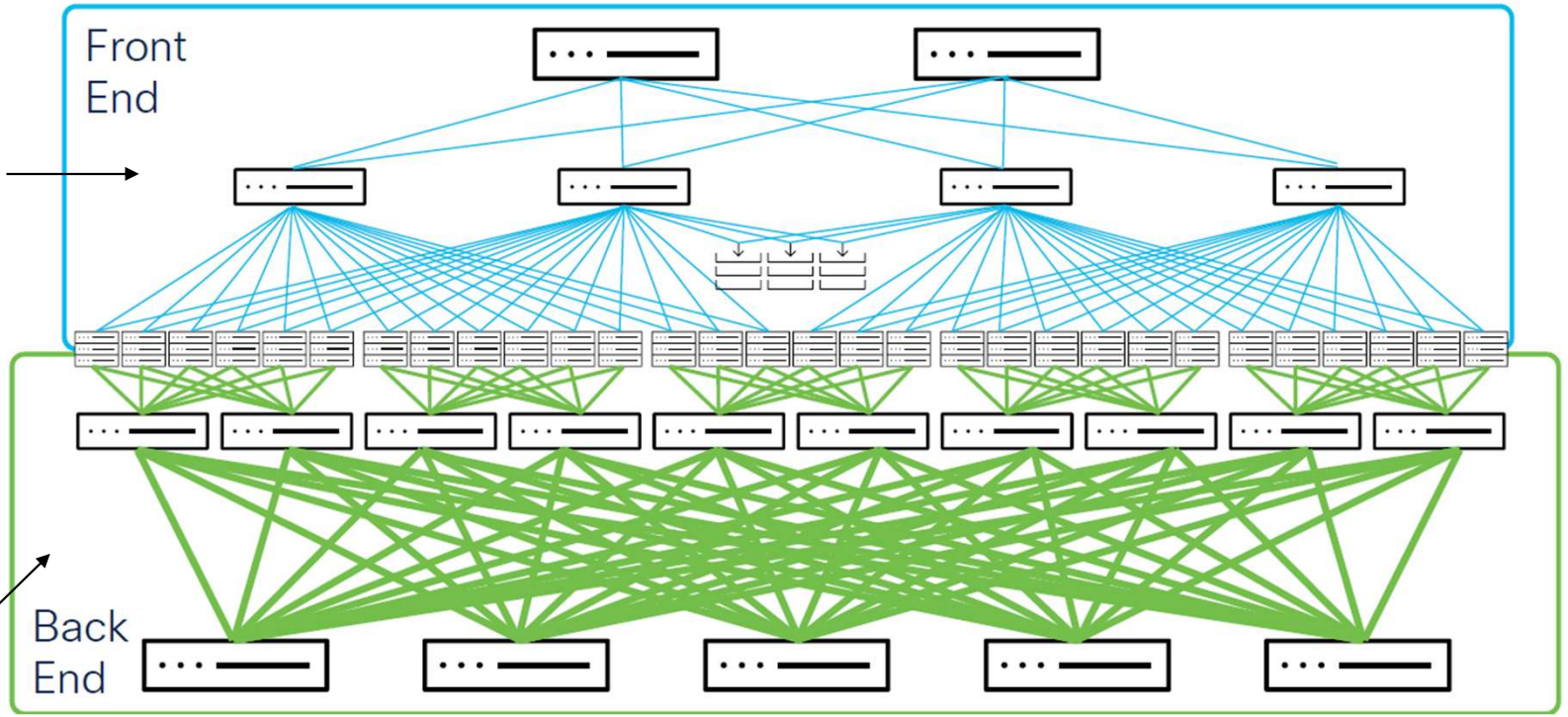
RoCE(v1/v2)

- **RoCE** (***R**DMA **o**ver **C**onverged **E**thernet*) è un protocollo di rete che consente di effettuare RDMA su una rete Ethernet utilizzando il protocollo *Infiniband*



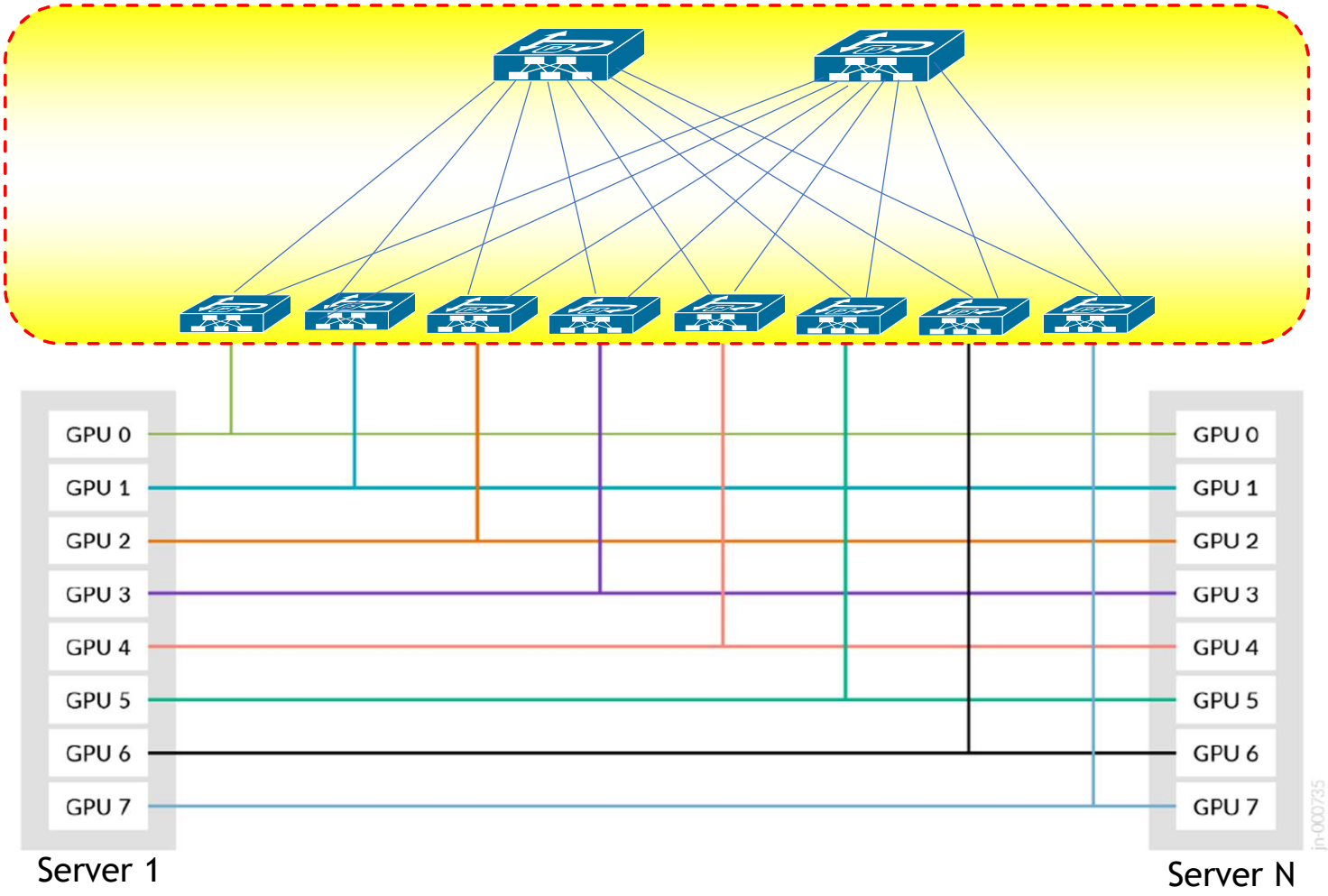
Componenti di un *AI cluster*

Frontend (Nord-Sud)
- Interconnessione con l'esterno
- Gestione degli host
- *Storage* (opzionale)



Backend (Est-Ovest)
- Infrastruttura di calcolo
- Elaborazione dei dati e addestramento dei modelli
- Comunicazioni GPU↔GPU
- *Non-blocking*

Design ottimizzato di una rete di *backend*

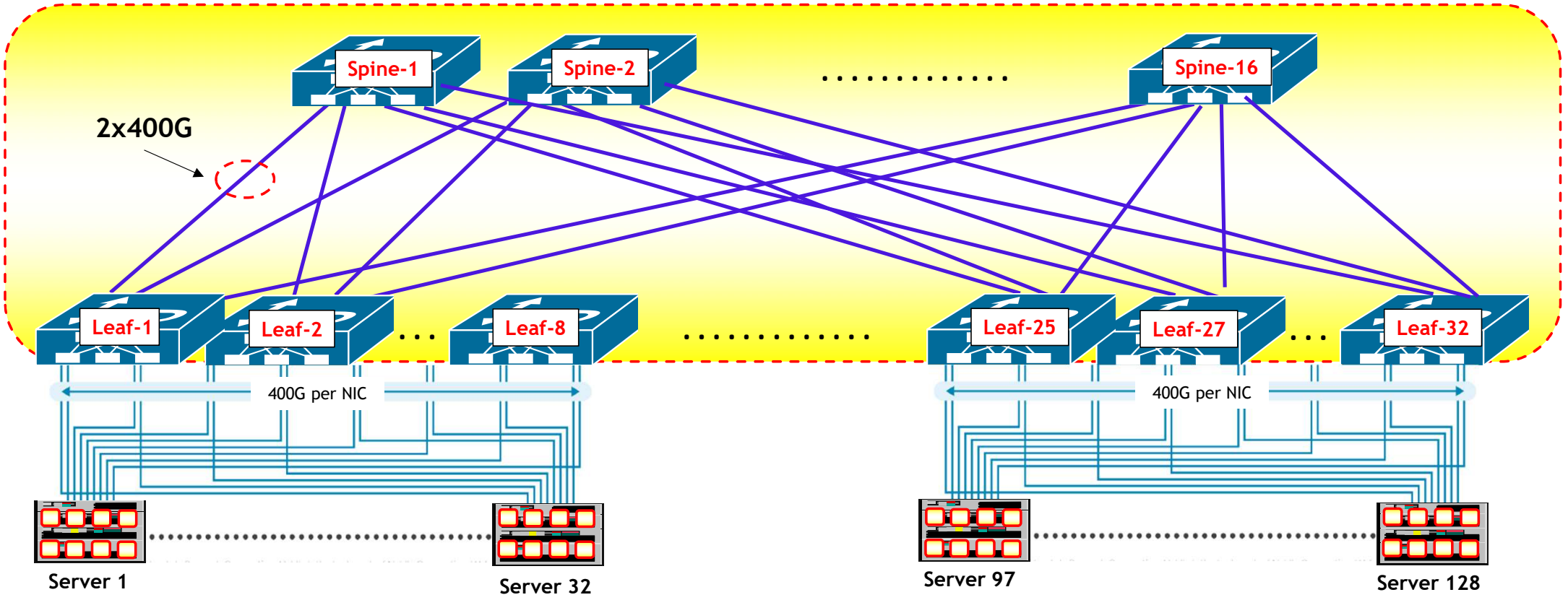


Esempio: dimensionamento di una rete di *backend* (1/2)

REISS ROMOLI

- IPOTESI
  - *Cluster* con 1.024 GPU, 8 GPU per server
  - Connessioni GPU↔*Leaf*: 400 Gbit/s
  - *Leaf* e *Spine* ciascuno con 64 porte a 400 Gbit/s
  - *Design* ottimizato
- DIMENSIONAMENTO
  - Numero totale di server=1.024/8=128
  - N.ro di porte utilizzabili per *Leaf*=64/2=32
    - Nota: le altre 32 sono per i collegamenti *uplink* verso gli switch *Spine*, per mantenere un rapporto di sovraccarico 1:1 (rete non bloccante)
  - N.ro totale di switch *Leaf*=1.024/32=32
  - N.ro totale di switch *Spine*=(32 *Leaf*x32 porte per *Leaf*)/64 porte per *Spine*=16
  - N.ro di connessioni da ogni *Leaf* a ogni *Spine*=32/16=2
    - Nota: i due collegamenti sono affasciati in LAG per una banda totale per LAG di 800 Gbit/s

Esempio: dimensionamento di una rete di *backend* (2/2)



Perché un *AI cluster* richiede reti Ethernet *lossless*? (1/2)

REISS ROMOLI

- Sensibilità alla latenza e *throughput*
  - Training distribuito: l'addestramento di modelli AI complessi spesso viene distribuito su più nodi del *cluster*
  - Inferenza su larga scala: in scenari *real-time* la perdita di pacchetti può causare ritardi inaccettabili nelle risposte e ridurre la capacità di elaborazione complessiva del *cluster*
- Efficienza computazionale
  - Ritrasmissioni e gestione della perdita dei pacchetti: il processo di ritrasmissione consuma risorse di CPU sui nodi del *cluster*, incrementa lo spreco di banda e introduce ulteriore latenza
  - Sincronizzazione efficiente: molti algoritmi di *machine learning* distribuito si basano su una sincronizzazione efficiente tra i nodi
- Scalabilità
  - All'aumentare del numero di nodi di un cluster, la probabilità che si verifichino perdite di pacchetti aumenta, soprattutto in reti non ottimizzate

Copyright Tiziano Tofoni

Perché un *AI cluster* richiede reti Ethernet *lossless*? (2/2)

REISS ROMOLI

- Utilizzo di RDMA
  - RDMA è particolarmente sensibile alla perdita di pacchetti. Se si verificano perdite, i benefici di RDMA (*bypass* del *kernel*, *zero-copy*) vengono vanificati dai meccanismi di recupero e ritrasmissione
- In sintesi, le reti *lossless* sono essenziali per gli AI *cluster* perché garantiscono
  - Comunicazione efficiente e veloce
  - Ottimizzazione delle risorse computazionali
  - Elevata scalabilità
  - Pieno sfruttamento di tecnologie come RDMA
- Per ottenere reti Ethernet *lossless* si utilizzano tecnologie come *Data Center Bridging* (DCB), in particolare *Priority Flow Control* (PFC)
  - PFC è definito dallo standard IEEE 802.1Qbb ed è una parte delle specifiche *Data Center Bridging* (DCB) definite dall'IEEE 802.1 *Working Group*

VI° modulo: DC per l'AI

#1 *AI cluster*

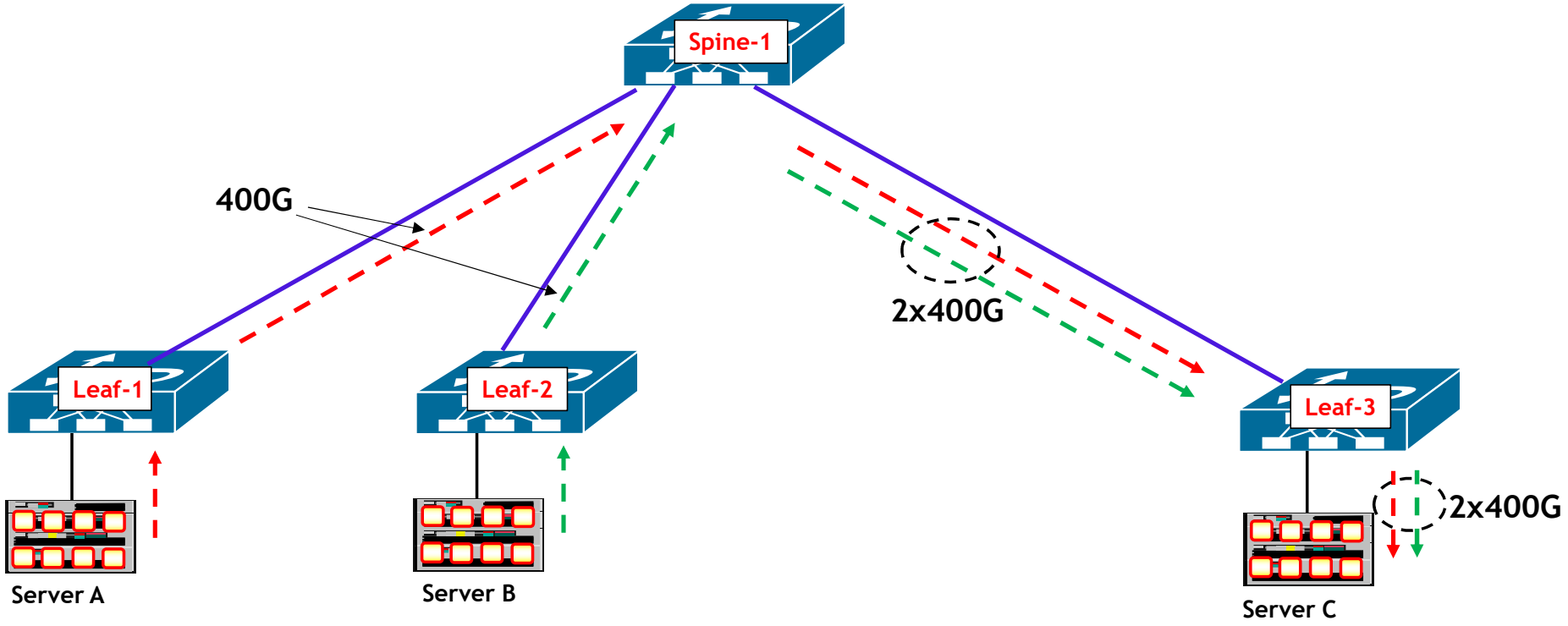
#2 Controllo della congestione: ECN e WRED

#3 *Lossless Ethernet*: *Priority Flow Control*

#4 PFC+ECN=DCQCN

Copyright Tiziano Tofoni

Non-blocking != lossless Ethernet



Traffico GPU-0 Server A→GPU-0 Server C

Traffico GPU-0 Server B→GPU-0 Server C

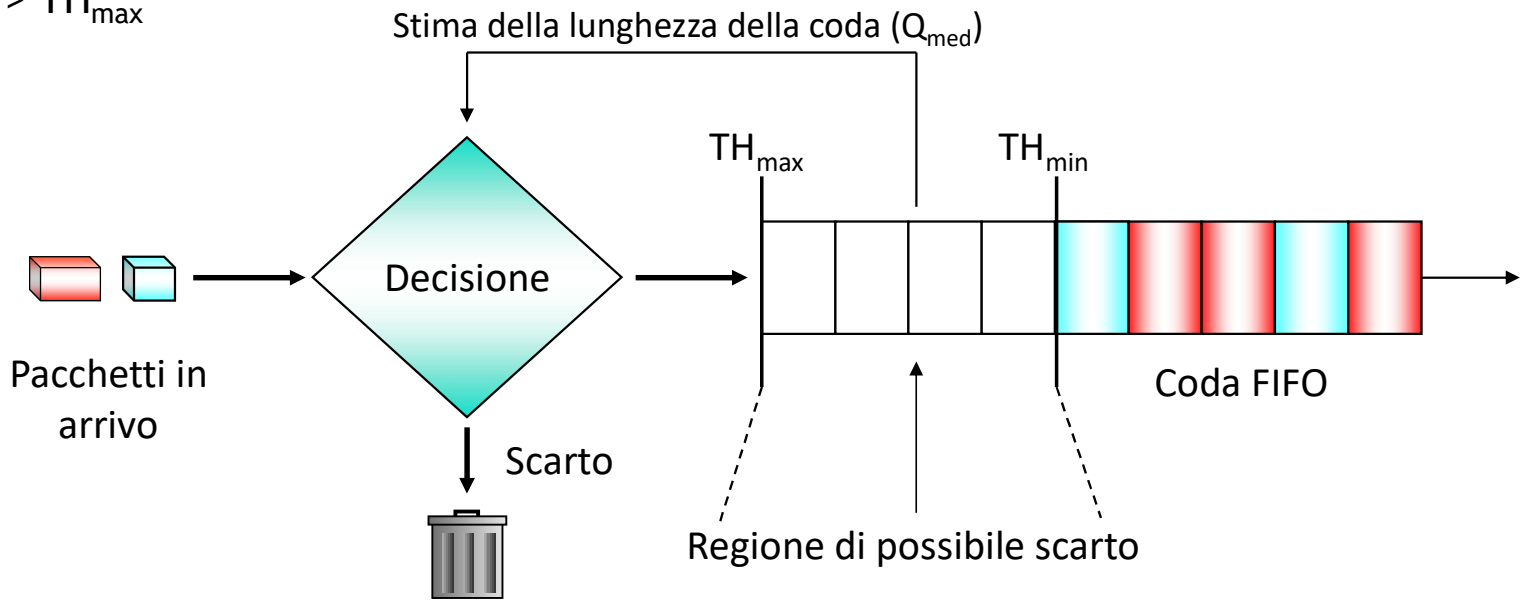
Come controllare la congestione

- Scarto proattivo dei pacchetti
  - Algoritmo **RED** (***R**andom **E**arly **D**etection*) o **WRED** (***W**eighted* **RED**) in presenza di più classi di servizio
  - Funzionamento
    - Scarta pacchetti in modo preventivo, ai primi sintomi di congestione
    - La congestione viene valutata attraverso opportune soglie sull'occupazione (stimata) del buffer

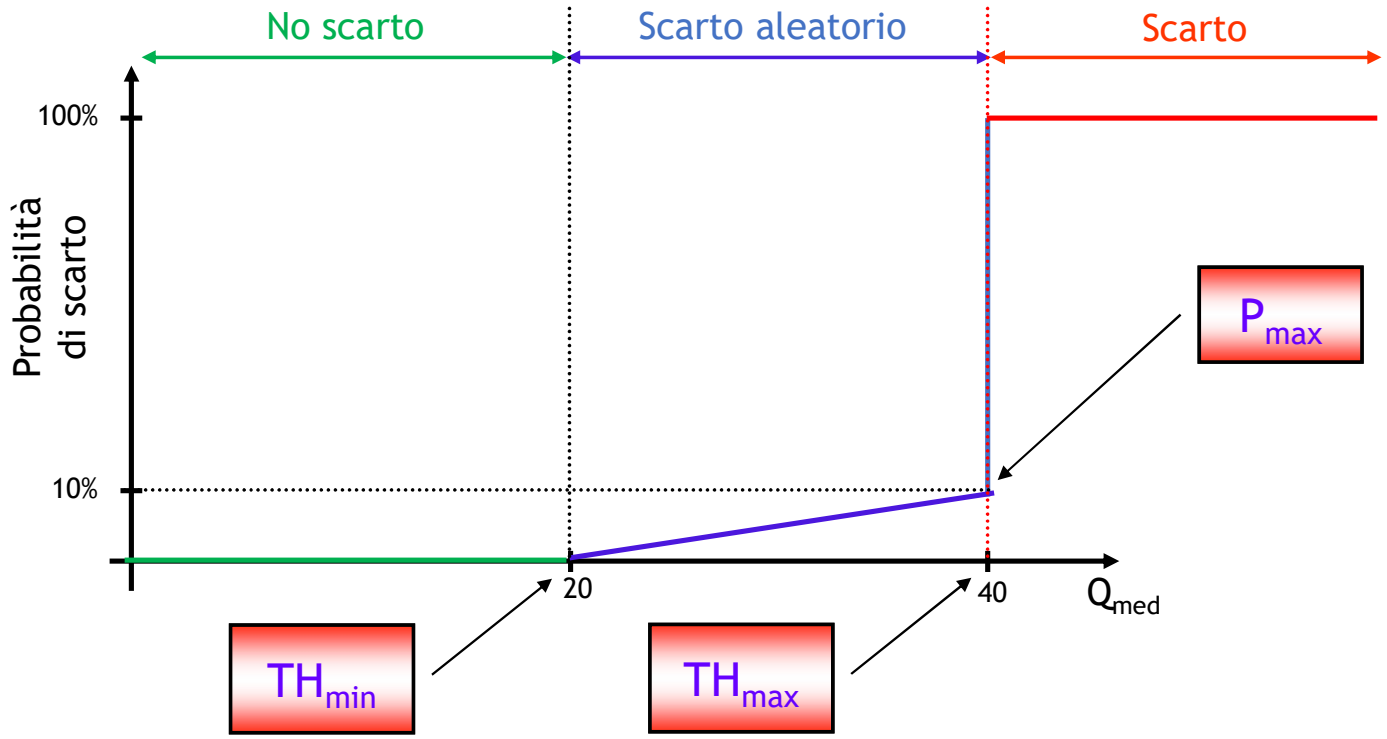
- Notifica della congestione
  - Via **ECN** (***E**xplicit **C**ongestion **D**etection*)
  - Funzionamento
    - Il dispositivo che rileva la congestione lo segnala nella direzione verso la destinazione, che risponde notificando la congestione alla sorgente
    - Sulla base della notifica dello stato di congestione, la sorgente esegue opportune operazioni di controllo del traffico

L’algoritmo RED: come funziona

- RED funziona sulla base di due soglie, TH_{min} e TH_{max}, e di uno stimatore dell’occupazione del buffer
- Tre regioni di funzionamento (Q_{med}=occupazione stimata)
  - No scarto: se $Q_{med} < TH_{min}$
  - Scarto aleatorio: se $TH_{min} \leq Q_{med} \leq TH_{max}$
  - Scarto: se $Q_{med} > TH_{max}$

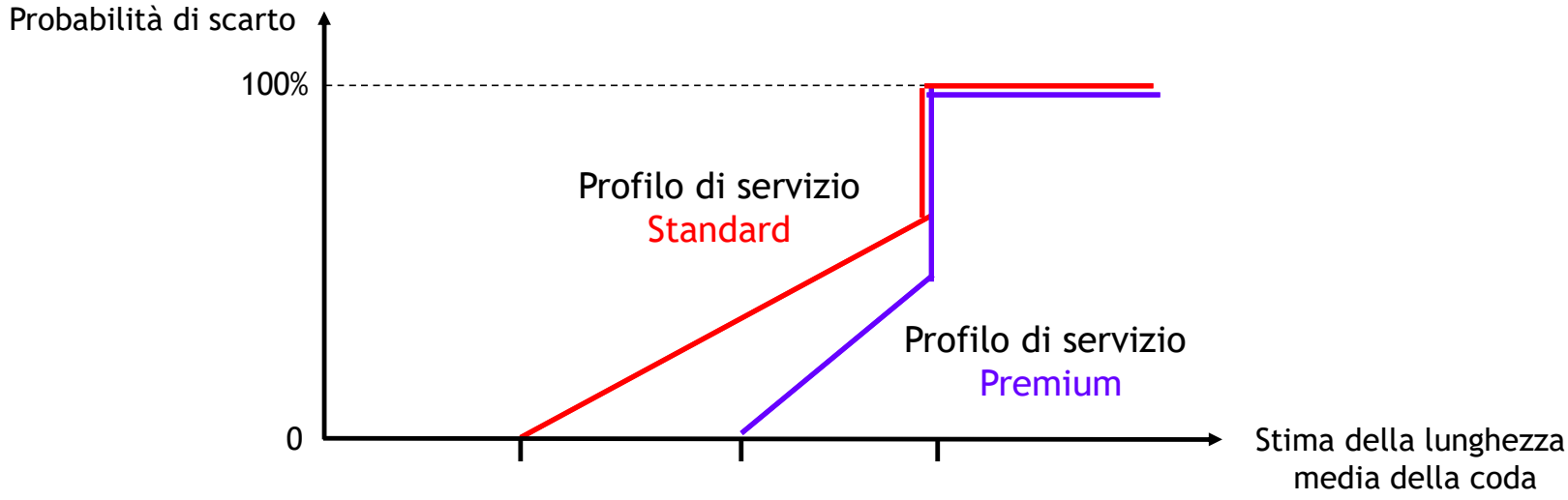


Profilo della probabilità di scarto



Weighted RED (WRED)

- Consente di impostare diversi profili di funzionamento, differenziati per soglie e/o probabilità di scarto
- Ogni profilo è caratterizzato da tre parametri
  - Soglia minima (TH_{min})
  - Soglia massima (TH_{max})
  - Massima probabilità di scarto (P_{max})

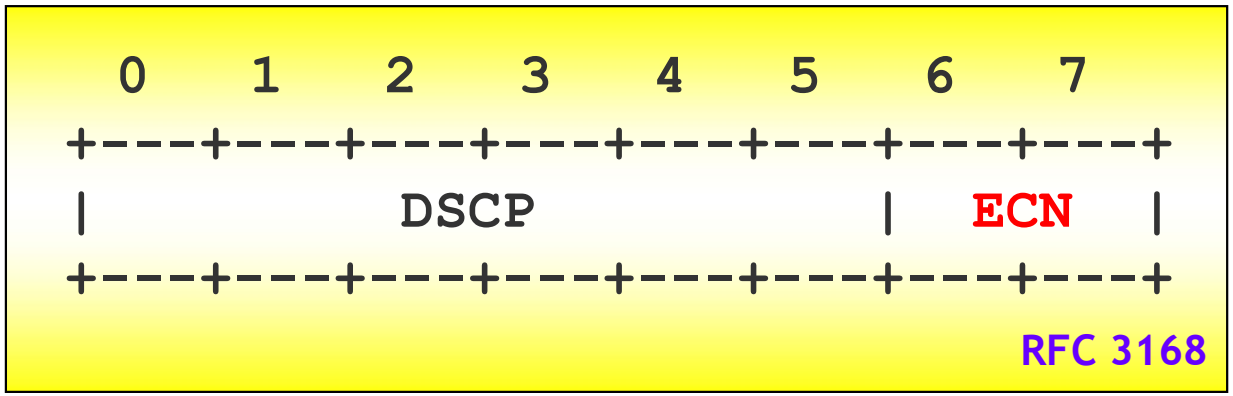


Explicit Congestion Notification (ECN)

REISS ROMOLI

- ECN è un meccanismo esplicito che consente di notificare lo stato di congestione di un buffer
  - È basato sulla scrittura di due opportuni bit nel campo ToS dell'intestazione IP
  - Simile al meccanismo basato sui bit FECN e BECN nel *Frame Relay*
  - Può essere utilizzato come indicatore di congestione alternativo alla perdita dei pacchetti nelle connessioni TCP
  - Specificato nella RFC 3168
- L'indicazione di congestione può essere utilizzata dagli *end-point* per rallentare la frequenza di emissione dei pacchetti
  - Nel caso di connessioni TCP vengono utilizzati due dei bit riservati nell'intestazione TCP
  - Nel caso del protocollo RoCEv2 si utilizzano pacchetti **CNP** (**C***ongestion* **N***otification* **P***acket*)

Definizione del campo ECN

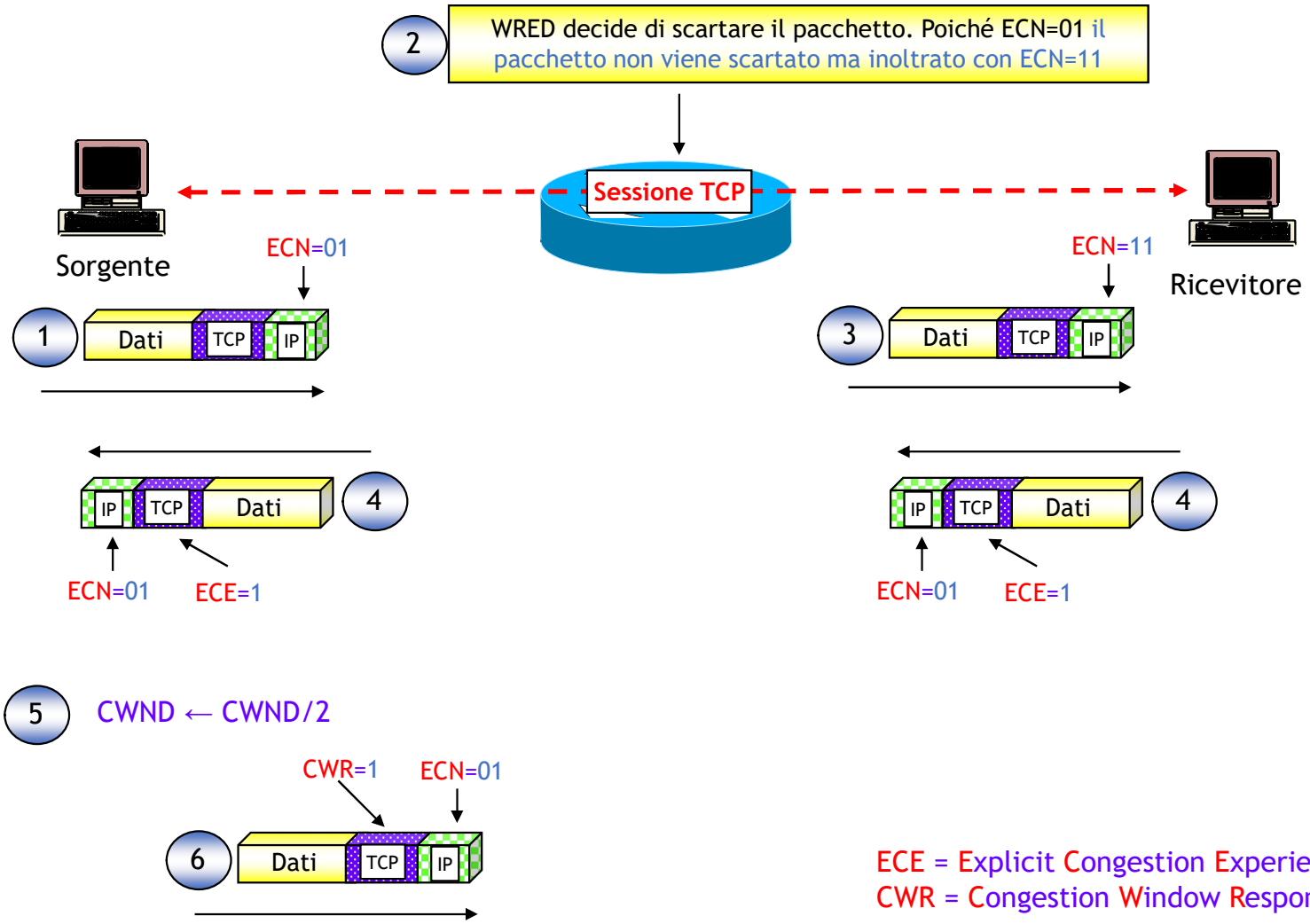


- ECN = **00b** – *Not ECN-capable Transport* (Not-ECT)
- ECN = **01b** – *ECN-capable Transport* (1) (ECT(1))
- ECN = **10b** – *ECN-capable Transport* (0) (ECT(0))
- ECN = **11b** – *Congestion Experienced* (CE)

ECN e WRED

- ECN è una estensione del WRED che ne migliora le prestazioni
- Se WRED decide di scartare un pacchetto, si possono verificare i seguenti eventi
  - Se la combinazione dei bit ECN è “01b” o “10b”, invece di scartarlo il pacchetto viene inoltrato marcando i bit ECN con la combinazione “11b”
  - Se la combinazione dei bit ECN è “00b” il pacchetto viene scartato (WRED classico)
  - Se la combinazione dei bit ECN è “11b” il pacchetto viene inoltrato senza alcuna modifica

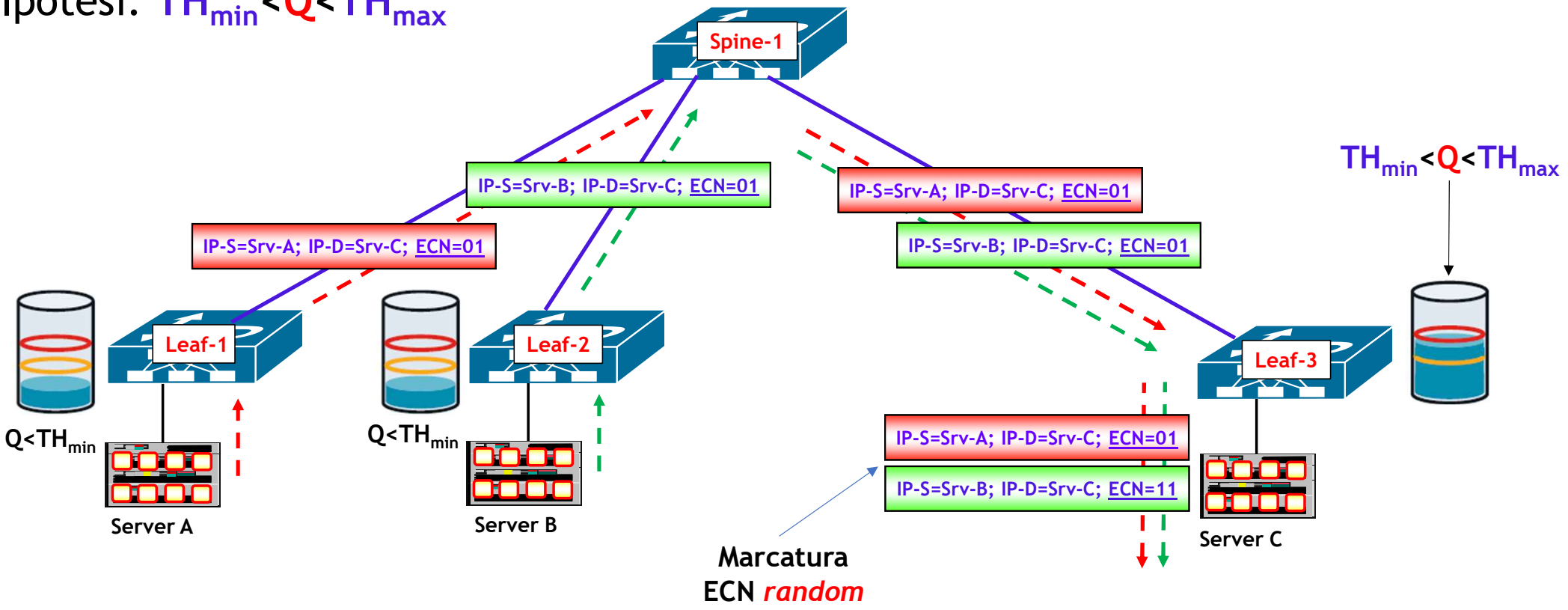
ECN e TCP



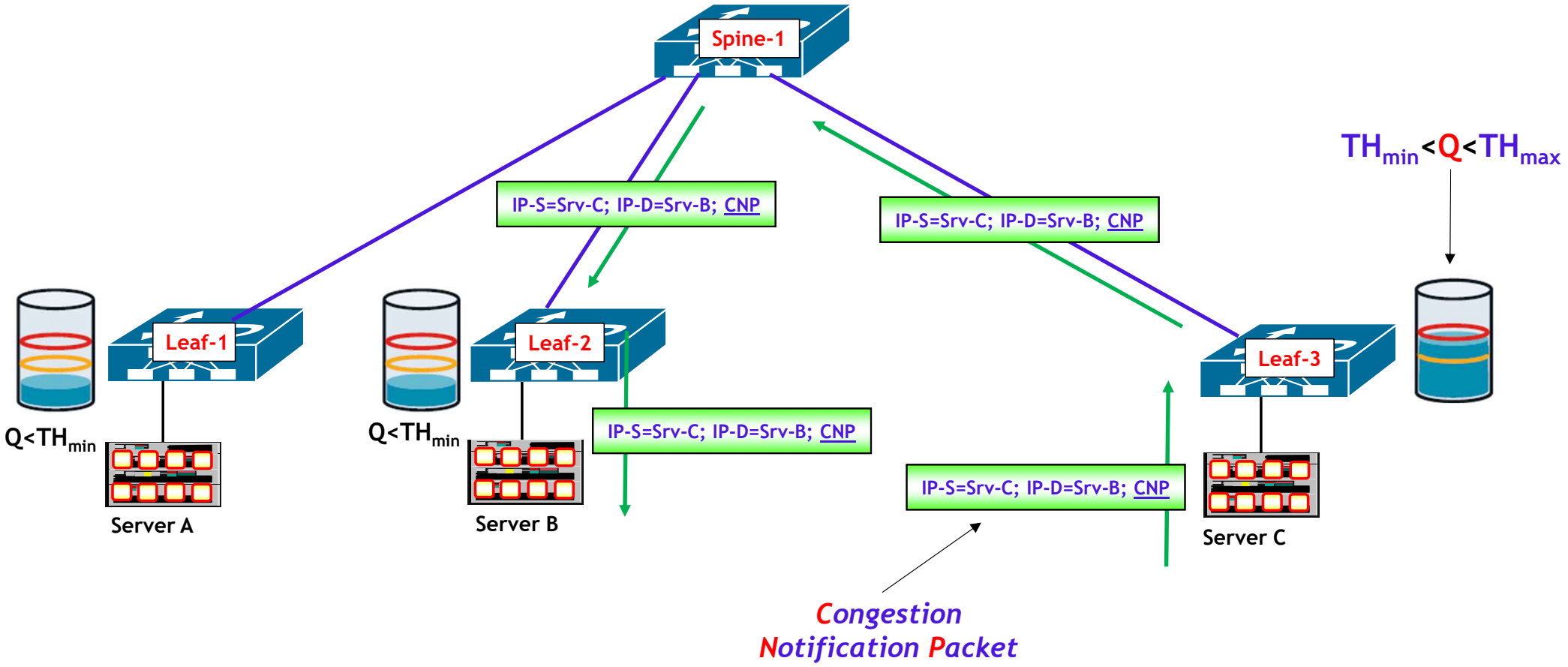
ECE = Explicit Congestion Experience
CWR = Congestion Window Response

ECN e RoCEv2 (1/4)

- Ipotesi: $TH_{min} < Q < TH_{max}$

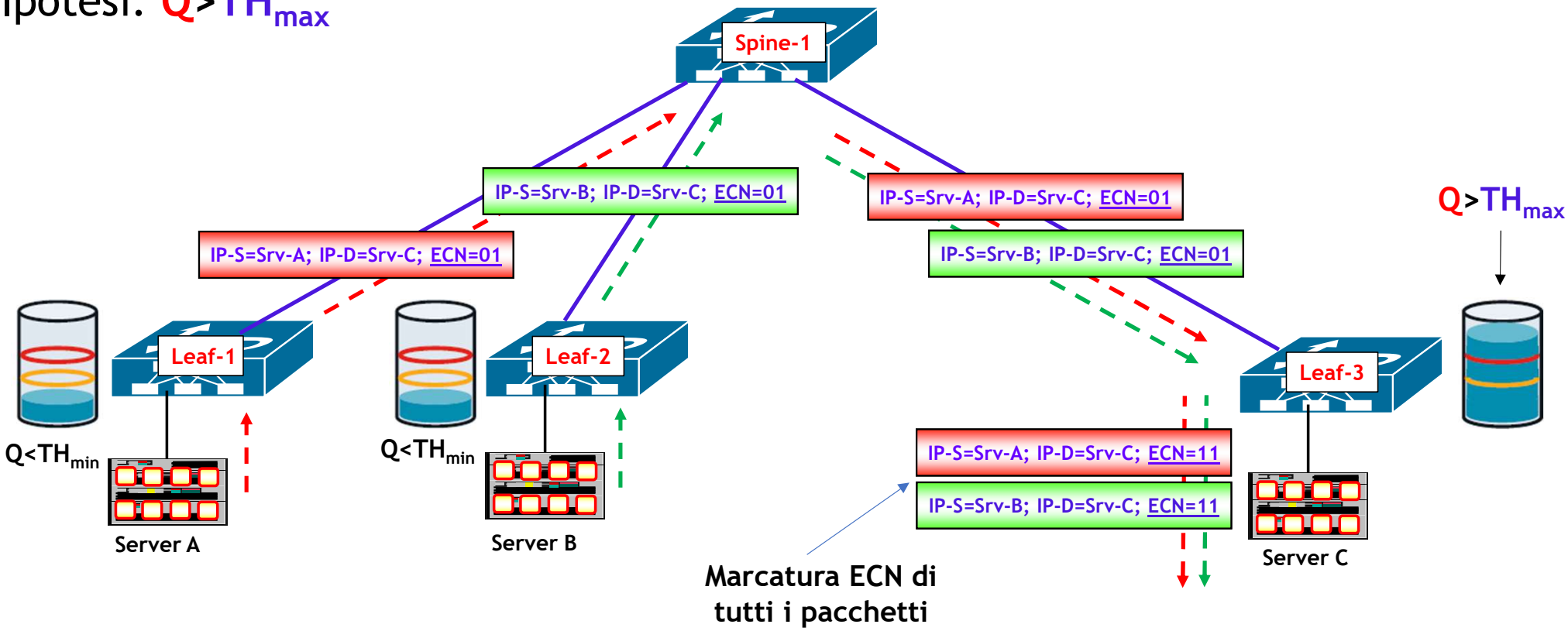


ECN e RoCEv2 (2/4)

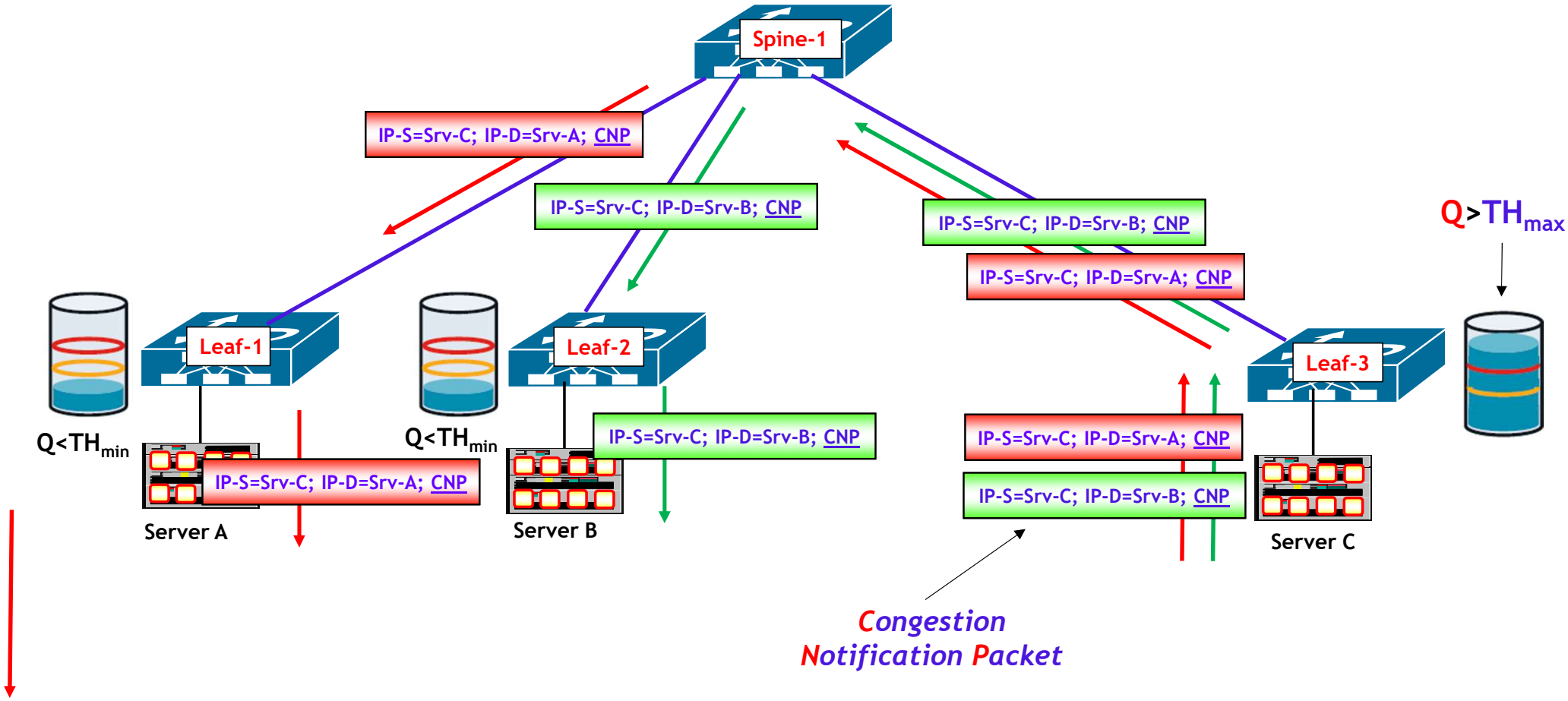


ECN e RoCEv2 (3/4)

- Ipotesi: Q>TH_{max}



ECN e RoCEv2 (4/4)



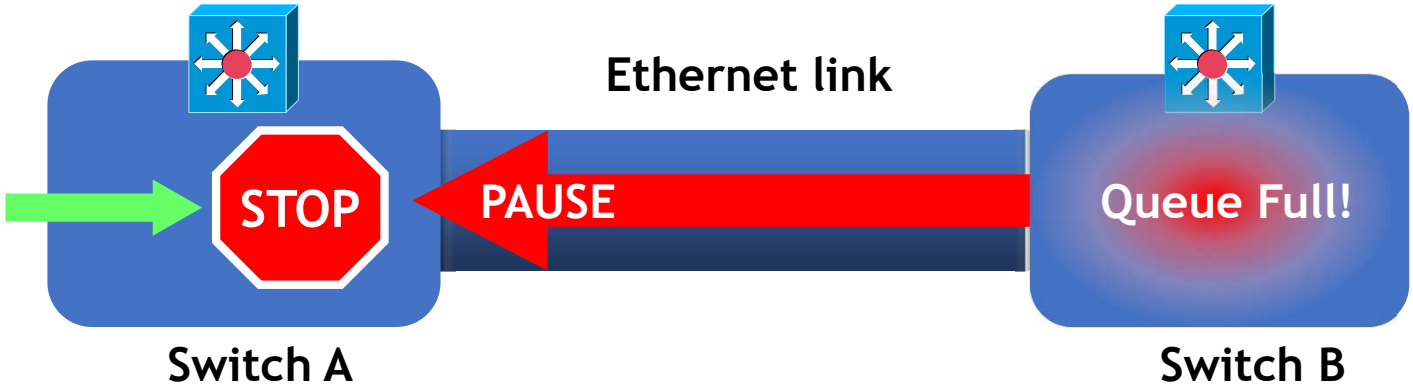
VI° modulo: DC per l'AI

#1	*AI cluster*
#2	Controllo della congestione: ECN e WRED
#3	*Lossless Ethernet*: *Priority Flow Control*
#4	PFC+ECN=DCQCN

Copyright Tiziano Tofoni

Può esistere una rete Ethernet *lossless*?

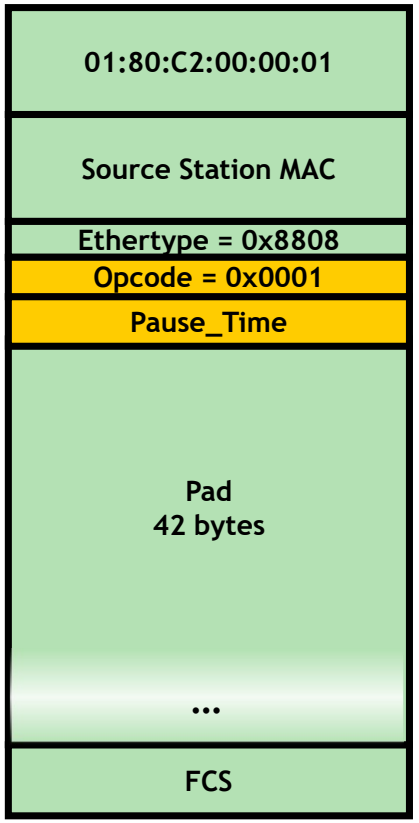
- Si, utilizzando trame di controllo Ethernet PAUSE



- Mediante una trama PAUSE è possible bloccare la trasmissione di trame Ethernet per un determinato periodo di tempo
  - Definito in IEEE 802.3 – Annex 31B
- Le trame Ethernet PAUSE trasformano una rete Ethernet in una rete *lossless*

Il formato delle trame PAUSE

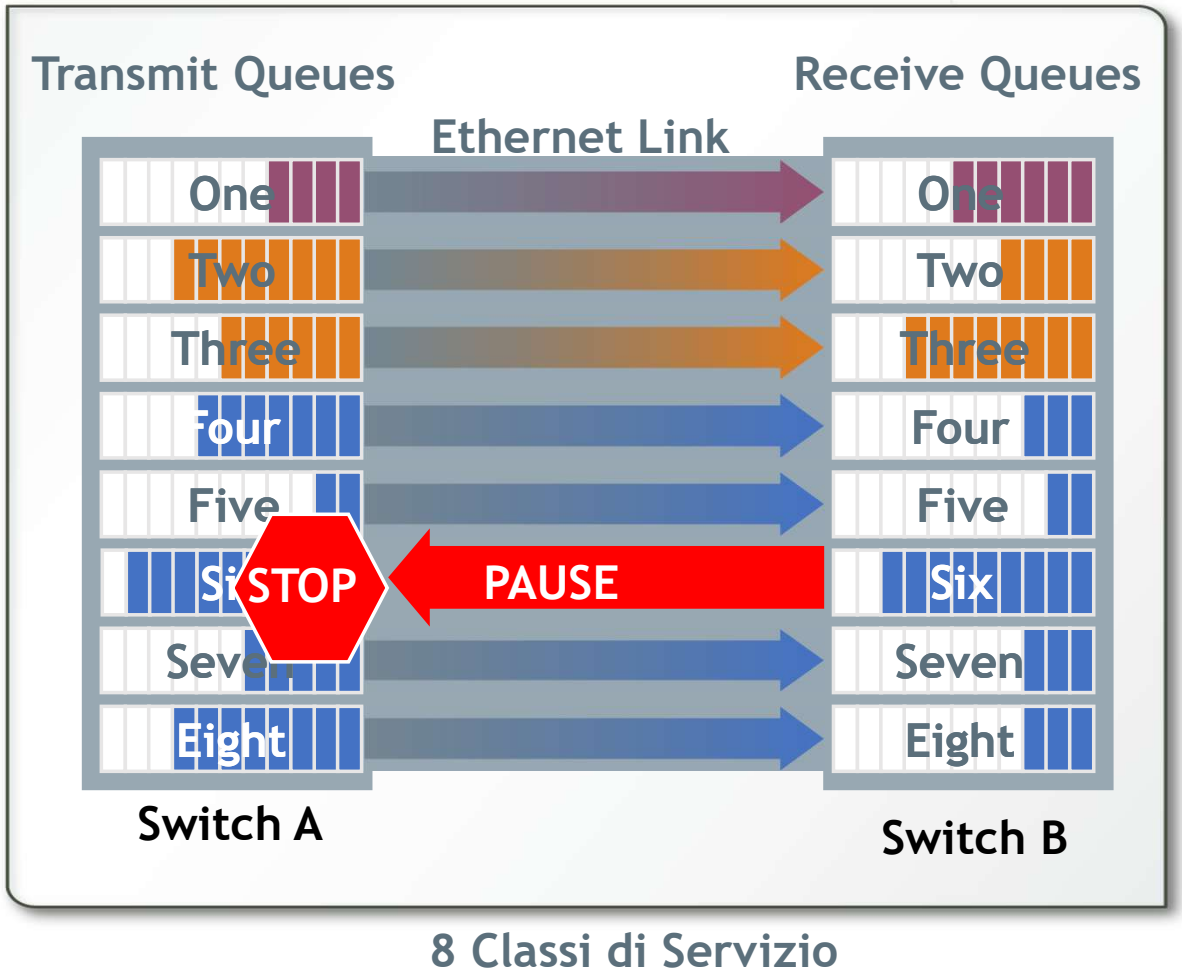
- Sono trame Ethernet standard, *untagged*
- MAC destinazione multicast (01:80:C2:00:00:01)
- **Ethertype** = 0x8808 (=trame di controllo)
- **Opcode** = 0x0001 (tipo di trama di controllo = PAUSE)
- **Pause_Time** è il tempo per cui il trasmettitore deve rimanere in pausa (misurato in *Pause Quanta* (=512 bits time))
  - Unico valore di Pause_time per l'intero collegamento



Copyright Tiziano Tofoni

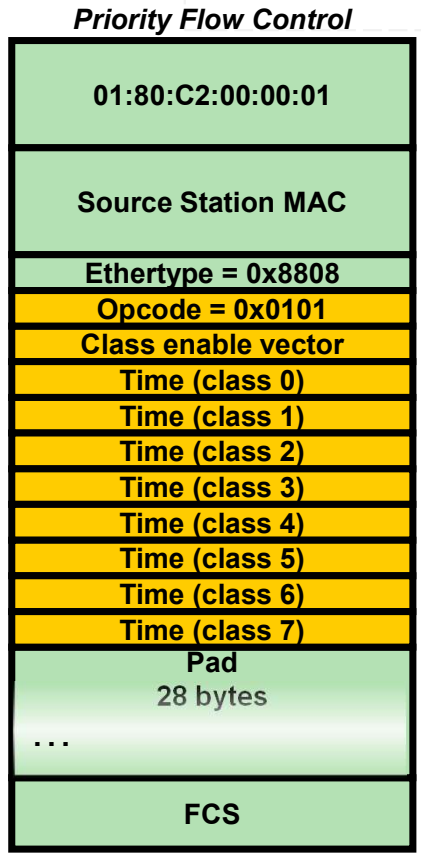
Priority Flow Control (PFC)

- Meccanismo di controllo di flusso standard (IEEE 802.1Qbb)
- Il controllo di flusso è applicato su base classe di servizio
  - Massimo 8 Classi di Servizio
- Utilizza lo stesso meccanismo *lossless* Ethernet basato sulle trame di controllo PAUSE



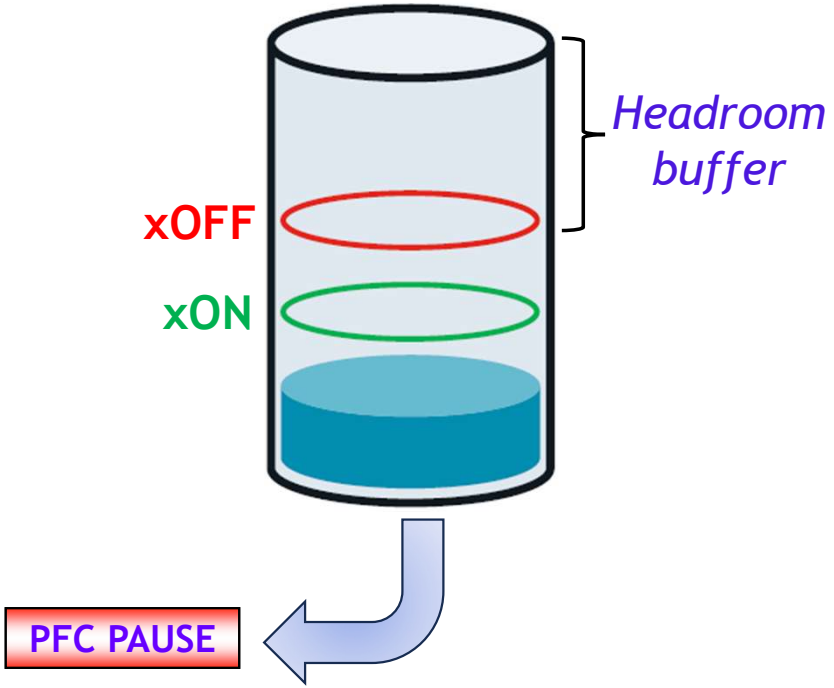
Il formato delle trame PAUSE per il PFC

- Formato simile alle trame di controllo PAUSE
- **Opcode** = 0x0101 per distinguerle dalle trame PAUSE ordinarie
- **Class Vector**: vettore di 8 *bit*; ogni *bit* indica per quale Classe di Servizio la trama trasporta le informazioni per bloccare il traffic
- 8 Pause_Time, uno per ciascuna Classe di Servizio



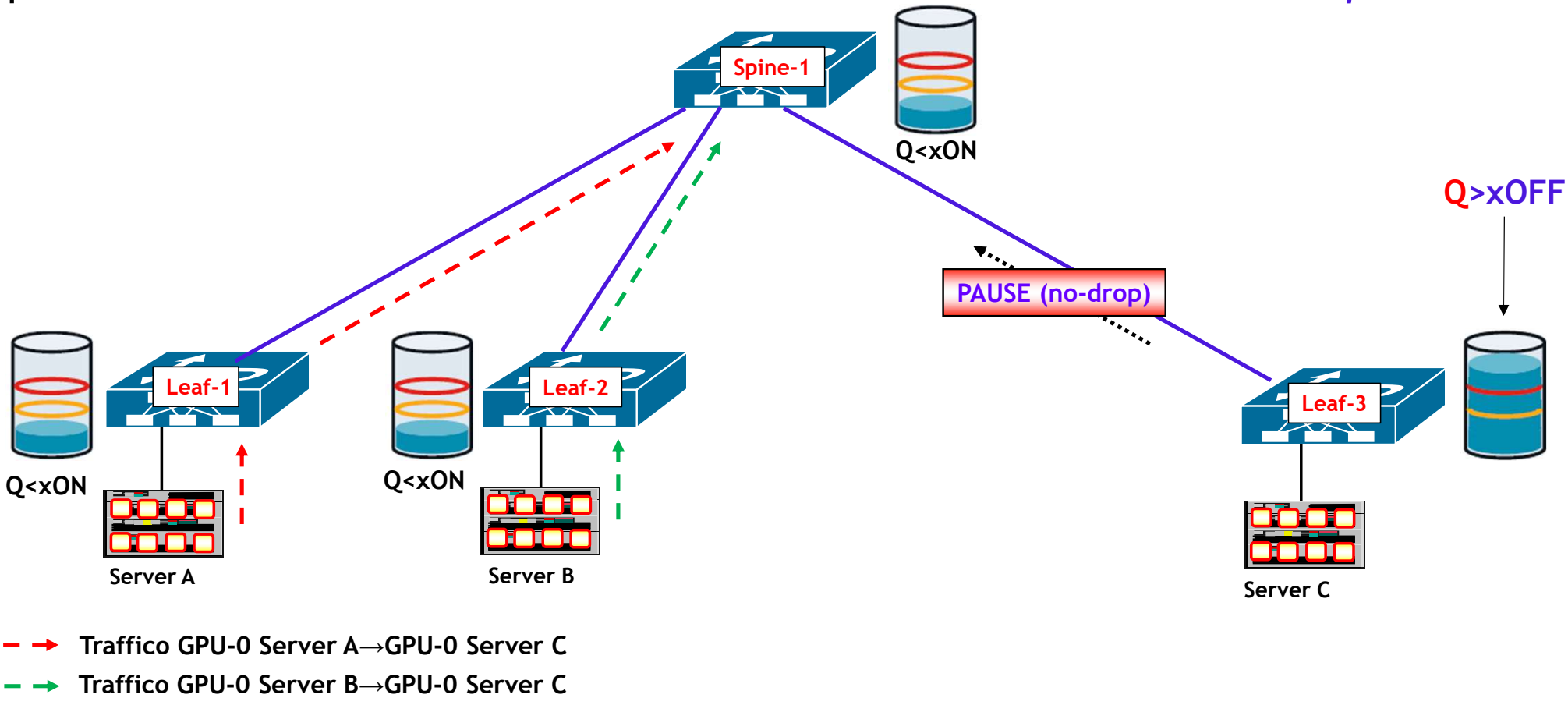
Funzionamento (1/4)

- Due soglie sulla lunghezza del *buffer* per Classe di Servizio
  - **xOFF**: il superamento di questa soglia comporta l'invio di trame PAUSE per bloccare il traffico della Classe di Servizio
  - **xON**: quando la lunghezza del buffer scende sotto questo valore viene ripreso l'invio del traffico della Classe di Servizio
- Presenza di una «riserva» (*Headroom buffer*): necessaria per il traffico *in-flight* mentre la trama PAUSE arriva a destinazione



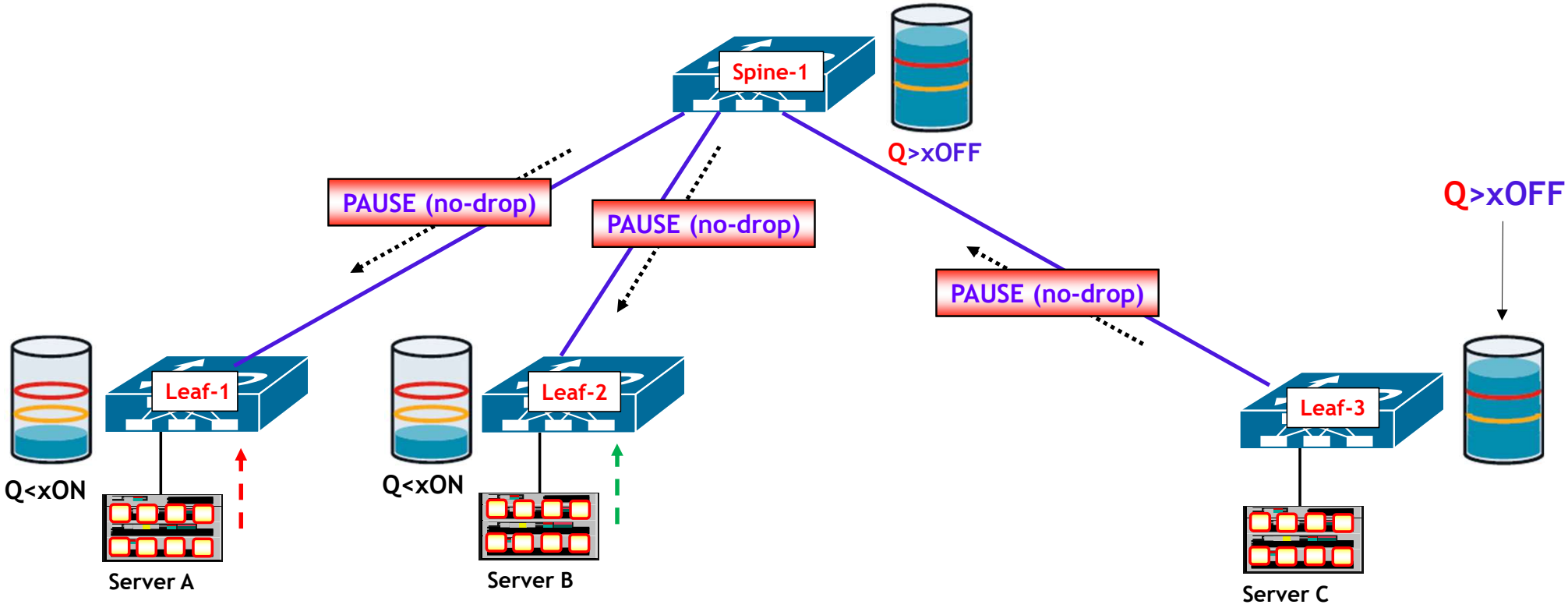
Funzionamento (2/4)

- Ipotesi: tutto il traffico inter-GPU viene inserito in una coda «*no-drop*»



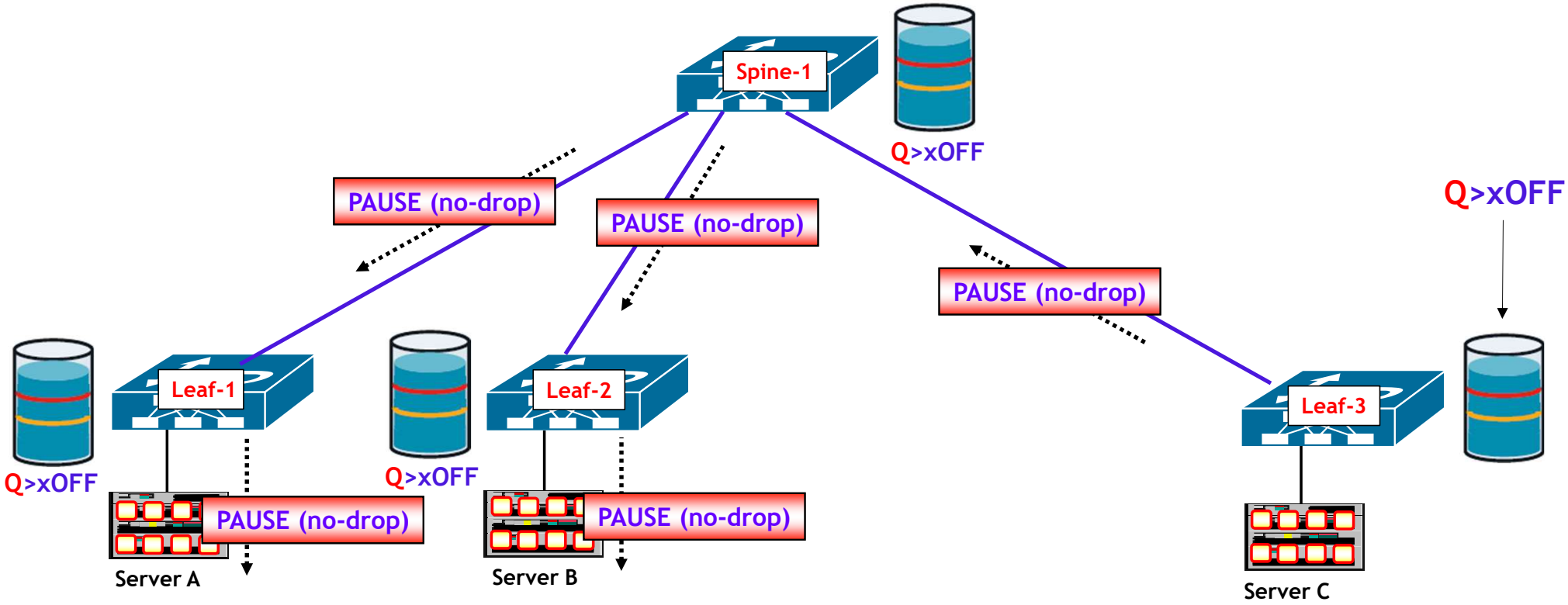
Funzionamento (3/4)

- Quando viene superata la soglia xOFF su Spine-1 questi invia una trama PAUSE *downstream* a Leaf-1 e Leaf-2



Funzionamento (4/4)

- Quando viene superata la soglia xOFF su Leaf-1 e Leaf-2 questi inviano una trama PAUSE *downstream* ai server A e B



VI° modulo: DC per l'AI

#1 *AI cluster*

#2 Controllo della congestione: ECN e WRED

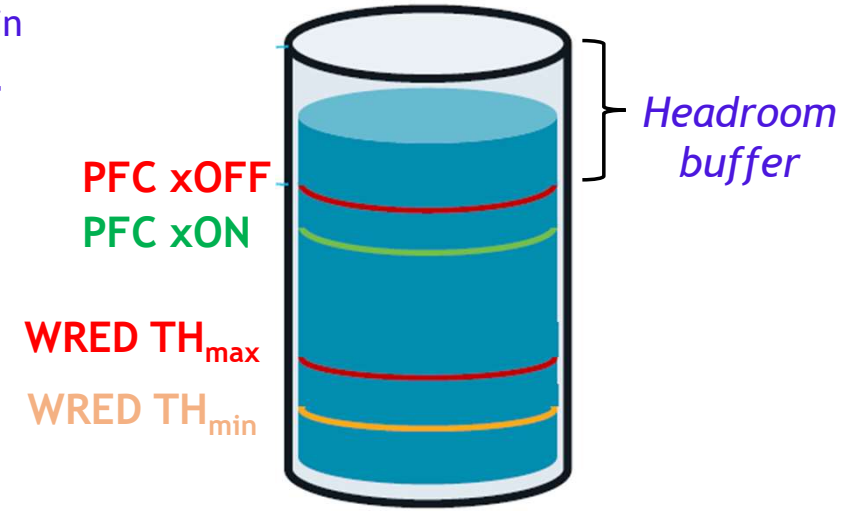
#3 *Lossless Ethernet*: *Priority Flow Control*

#4 PFC+ECN=DCQCN

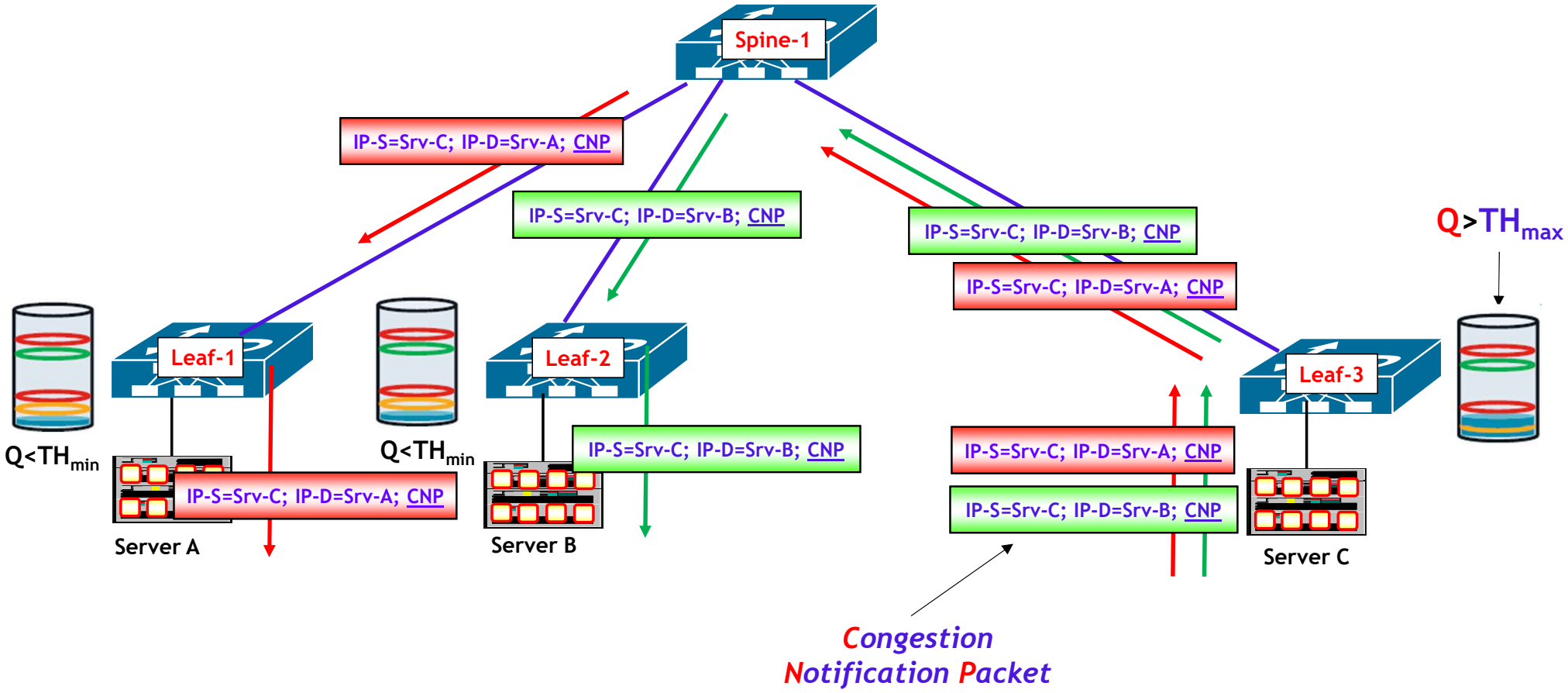
Copyright Tiziano Tofoni

Data Center Quantized Congestion Notification (DCQCN)

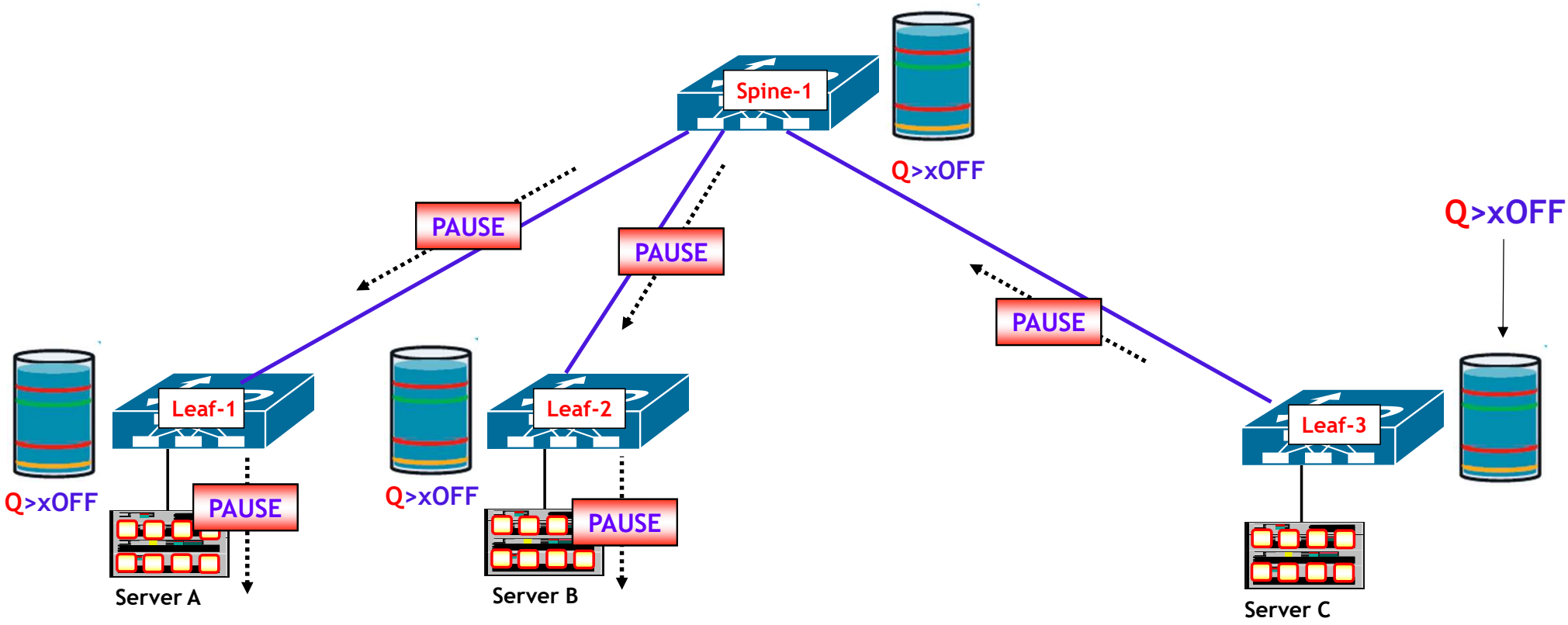
- DCQCN è la combinazione di PFC ed ECN+WRED
  - Consente di ottenere *low latency*, *high throughput* e *zero loss*
- Utilizza un approccio a due livelli
  - Applica l'ECN in anticipo per ridurre gradualmente la velocità di di trasmissione
  - Utilizza il PFC come backup per interrompere rapidamente il traffico sugli switch *upstream* (*hop-by-hop*) quando la congestione diventa grave
- Per il corretto funzionamento le soglie WRED (TH_{min} e TH_{max}) sono poste a valori inferiori alle soglie del PFC xON e xOFF



Prima azione: ECN



Seconda azione: PFC



Ultima pagina (finalmente …)

Grazie per l'attenzione

Copyright Tiziano Tofoni