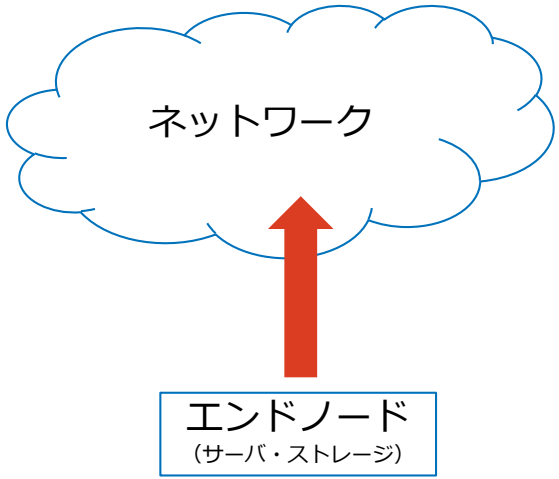
みんな First-Hop Redundancy どうしてるよ？

川上 雄也 (@yuyarin)
SDN Tech Lead, Cloud Services, NTT Ltd Japan

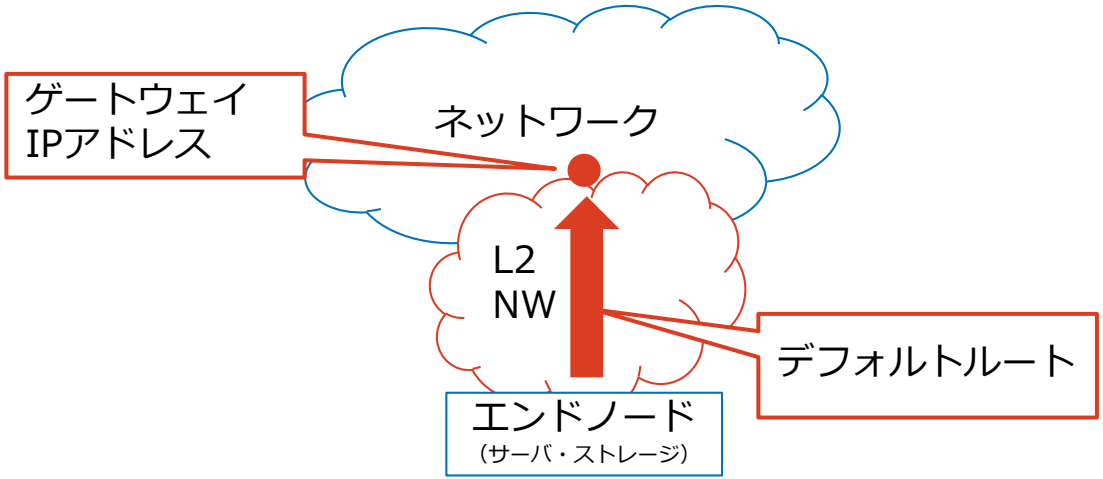
First-Hop Redundancyとは？

サーバやストレージなどの機器（エンドノード）がネットワークに接続する
最初の１歩目（First-Hop）の冗長性（Redundancy）



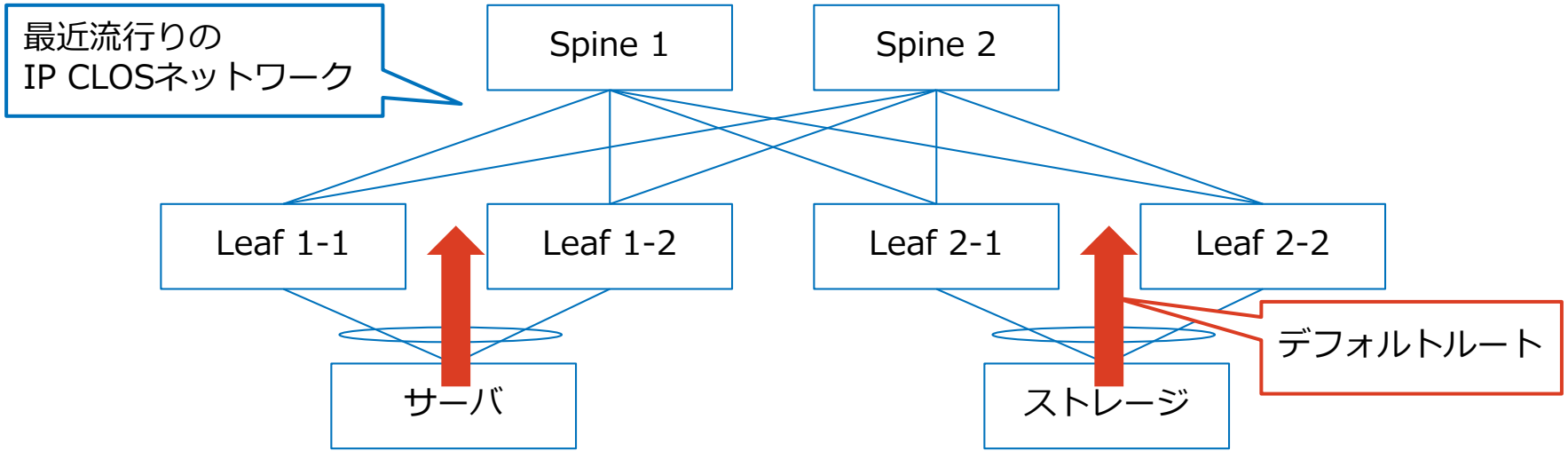
First-Hop Redundancy問題とは？

エンドノードのL3デフォルトルートのNext Hop（ゲートウェイIPアドレス）と
そこまでのL2ネットワークをどのように冗長するか、という問題



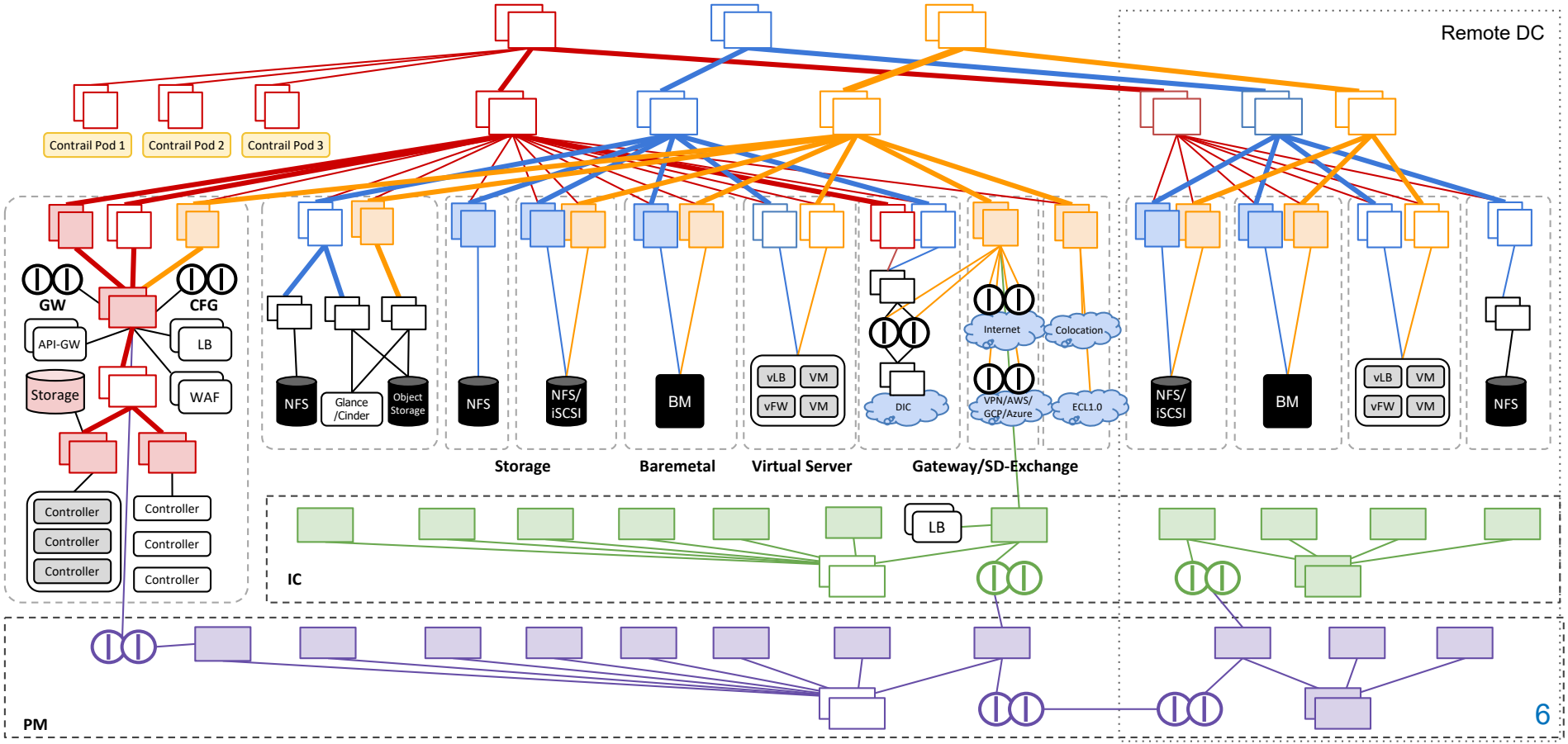
つまるところ

データセンターネットワークでサーバとかストレージをLeafにL3収容するときに
デフォゲをなんかいい感じに冗長化したい

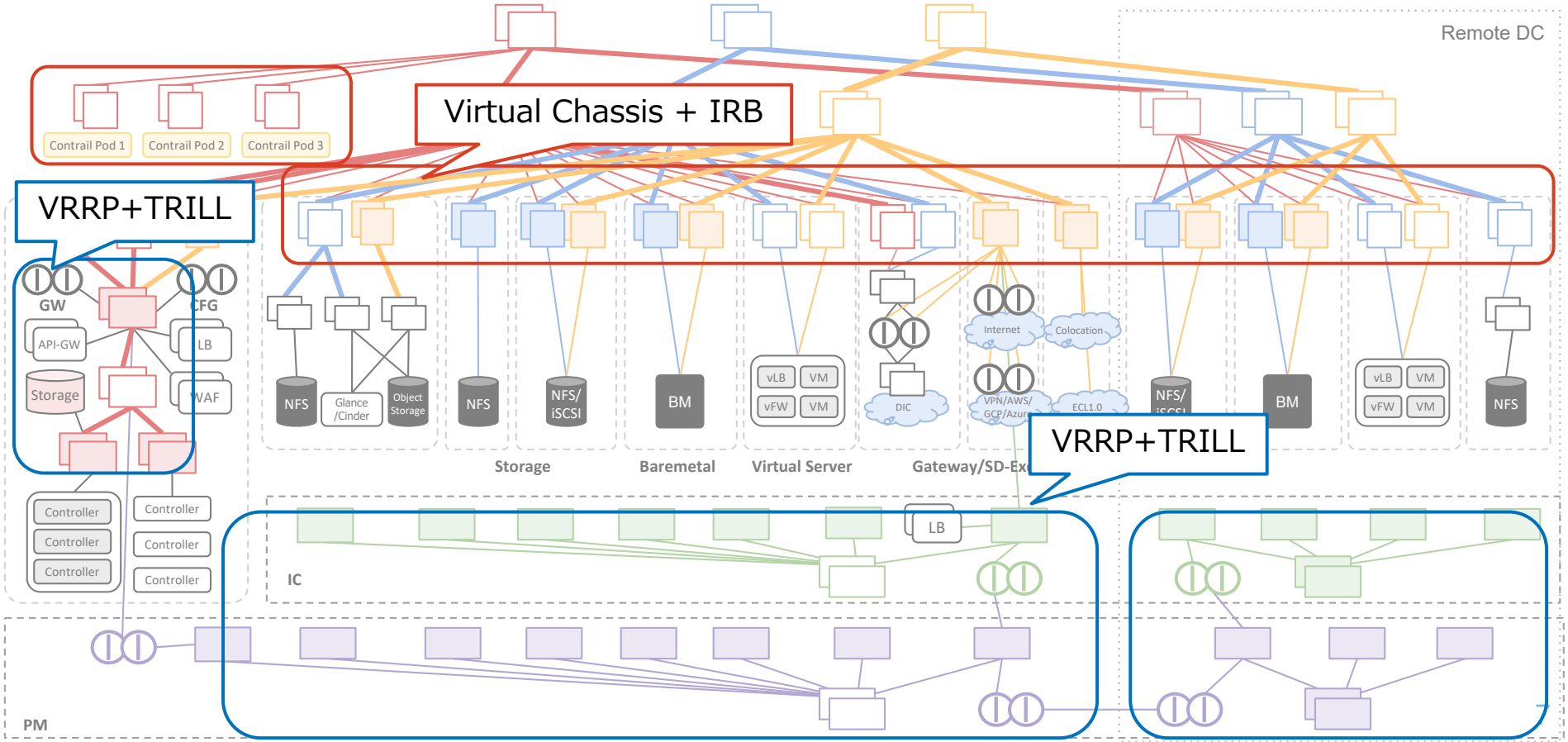


Enterprise Cloud 2.0 (ECL2.0)

ECL2.0 Network Overview



ECL2.0 Network Overview



このプログラムのモチベーション

これまでいろいろなサービスにおいてVRRPで痛い目を見てきた

ECL2.0では特にLeaf部分にVirtual Chassisを採用したがそれでも痛い目を見た

次のリージョンからSDN/NW基盤を新しい世代にするので、そのタイミングで
First-Hop Redundancyの実現手法を再考したい！！！

このプログラムの概要

First-Hop Redundancyを実現するこれまでの手法を整理するとともに、
これからの新しい手法としてのEVPN Anycast Gatewayの可能性を探ります

目次

1. First-Hop Redundancy概論
2. First-Hop Redundancy手法
    1. VRRP
    2. Virtual Chassis
    3. MC-LAG
    4. BGP
3. EVPN Anycast Gateway
4. 議論

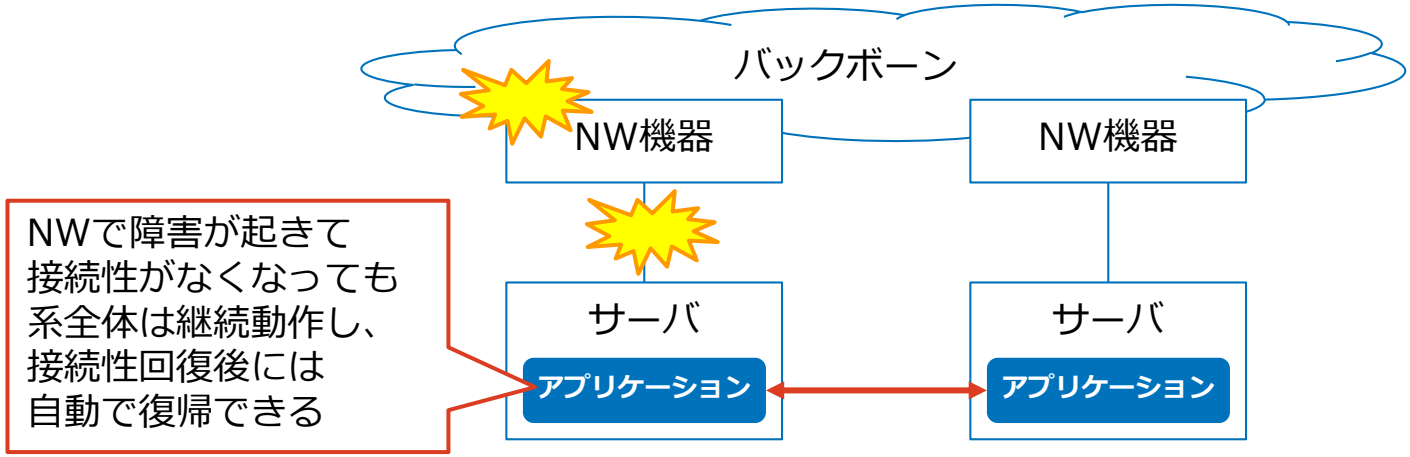
人はなぜFirst-Hop Redundancyを求めるのか…

人はなぜFirst-Hop Redundancyを求めるのか

求めないケースがあるとすれば…

- アプリケーションで完全に冗長性を取れていて各種の障害パターンにも自動で対応・回復できる場合（たとえばCloud Native）

➡ ストレージまで含めてこの境地に達している人はそんなに多くないと思う



人はなぜFirst-Hop Redundancyを求めるのか

多くの場合現実は

- 接続されている機器やその先のリソースまたは**アプリケーションを**
  ネットワークインフラ提供者から**制御できない**
  - ○ 顧客、他社、他組織、他部署、他チームに提供していてアプリケーションに口を出せない
  - ○ 口は出せるけどアプリケーションがそのように作られていない・作ることが出来ない

- メンテナンス性のために冗長化しておきたい
  - ○ 利用者側で何らかの対応が必要になるためにメンテナンスの通知や日程調整などが必要

従来の代表的なFirst-Hop Redanduncyの実現手法

大きく分けて４パターン

#	ネットワーク機器	エンドノード
1	VRRP(HSRP)	bonding
2	Virtual Chassis	bonding
3	MC-LAG	bonding (LACP)
4	BGP	BGP

First-Hop Redundancyの難しさ

それぞれの状況で最適な方法が違う

- 何の機器をつなぐのか
- 誰の機器をつなぐのか
- どのようなネットワークにつなぐのか
- どのような回線を使ってつなぐのか
- どのような通信要件・設計要件があるのか
- どの技術に対応しているか

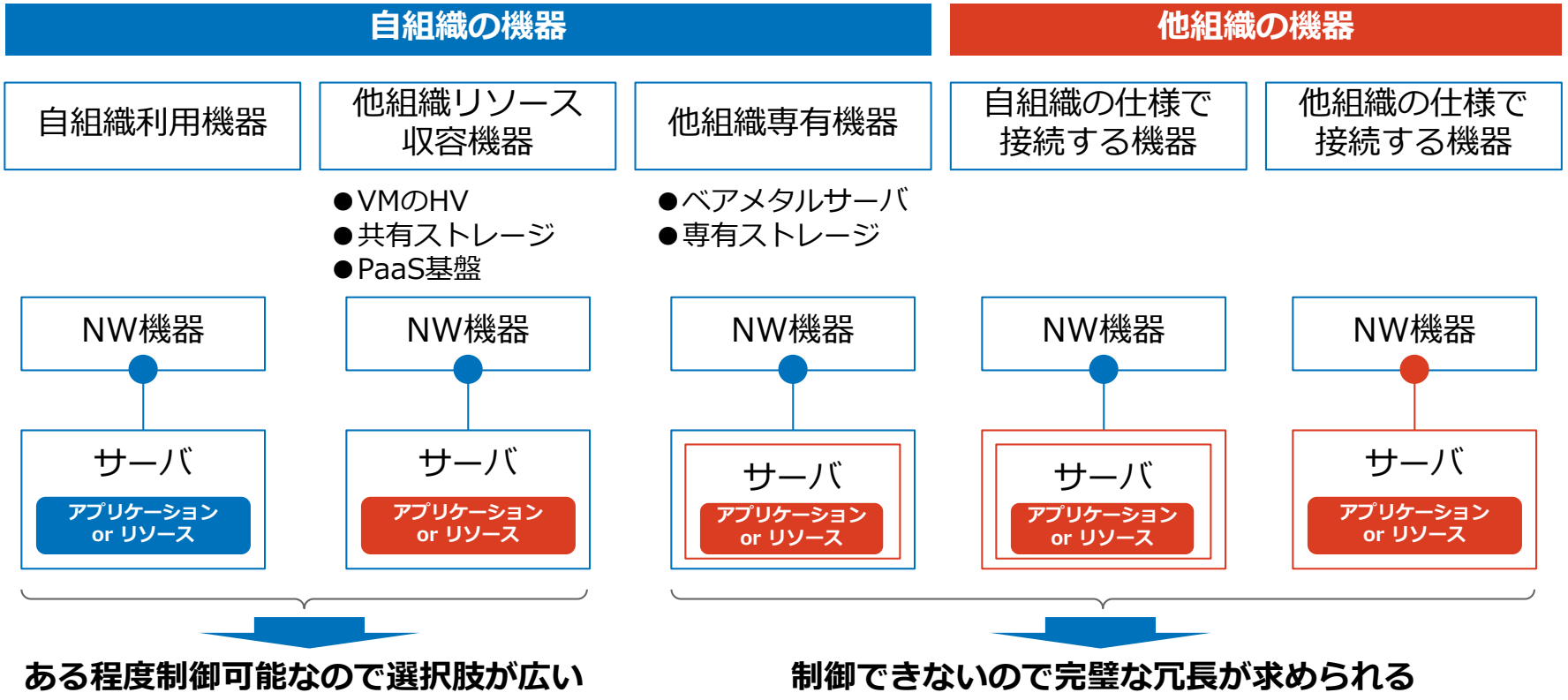
何の機器をつなぐのか

サーバ	ストレージ	L4ネットワーク機器
Linux/BSD	NFS	ファイアウォール
Windows	iSCSI	ロードバランサ
メインフレーム	オブジェクト	NAT
		WAF/UTM
		IPS/IDS

汎用Linuxサーバ以外は使える技術が限られたり、使えても機能が限定的だったりする

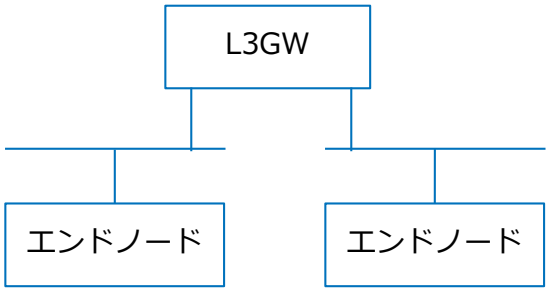
できればルーティングプロトコルを喋って欲しいが、構成上できない場合があるので考慮対象になりえる

誰の機器をつなぐのか


自組織の機器
他組織の機器
自組織利用機器
他組織リソース収容機器
他組織専有機器
自組織の仕様で接続する機器
他組織の仕様で接続する機器
●VMのHV
●共有ストレージ
●PaaS基盤
●ベアメタルサーバ
●専有ストレージ
NW機器
NW機器
NW機器
NW機器
NW機器
サーバ
アプリケーション or リソース
サーバ
アプリケーション or リソース
サーバ
アプリケーション or リソース
サーバ
アプリケーション or リソース
サーバ
アプリケーション or リソース
ある程度制御可能なので選択肢が広い
制御できないので完璧な冗長が求められる

どのようなネットワークにつなぐのか

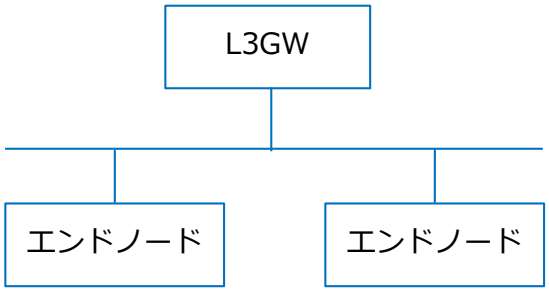
P2P L3



●安定性を重視してL2を他のエンドノードと共有したくないとき

L2が必要になる冗長技術は使えない

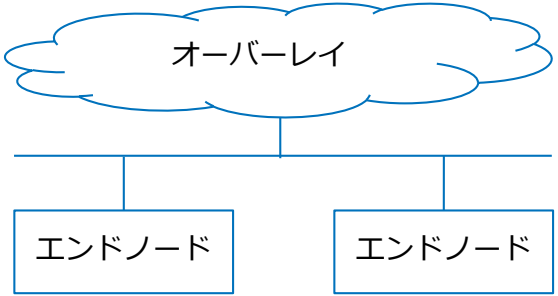
P2MP L3



●IPアドレスを節約したいとき
●１ペアのL3機器に多数のエンドノードを収容したいとき
●エンドノード間のHA構成等でL2疎通性が必要なとき（VRRPやcorosync）

選択肢が広い

L2/L3オーバーレイ

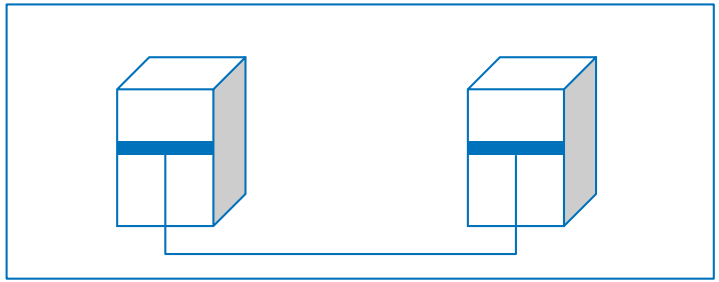


●オーバーレイサービスを提供しているとき

どのみちオーバーレイの先や内部でL3GWの冗長を考える必要がある

どのような回線でつなぐのか

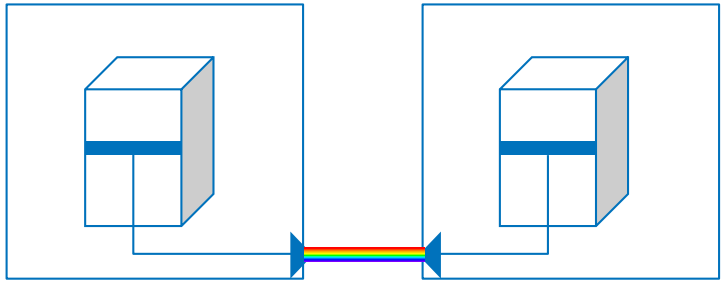
DC内配線



●ラック内やラック間などでSMF/MMF/UTPで届く範囲

特に考慮することはない

DC間～メトロ～遠距離



●伝送機器や専用線サービスを使う場合

LACPなどの特別なL2フレームの透過やリンクダウン転送の有無を考慮しておく

どのような通信要件・設計要件があるのか

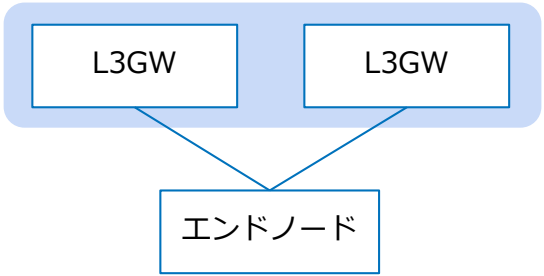
コスト	GWで必要な機能	許容通信断時間
設備投資コスト	NAT	冗長系の切り替わり時間
検証コスト	QoS	
保守運用コスト	Firewall	

収容機器数	収容リソース数
物理インターフェイス数	論理インターフェイス数
帯域	VRF数

どの技術に対応しているか

NW機器だけで冗長できる

VRRP | Virtual Chassis | MC-LAG

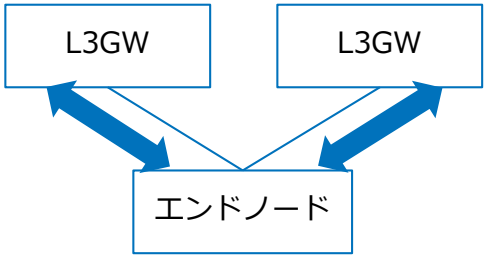


- エンドノードはbond mode 1 (Act-Stb)ができればOK
- MC-LAGのLACPは厳密には右側だけど、もうそろそろこっち側でもいいかも

ネットワーク機器側の技術要件だけを考慮すれば十分

エンドノードと連携する

BGP



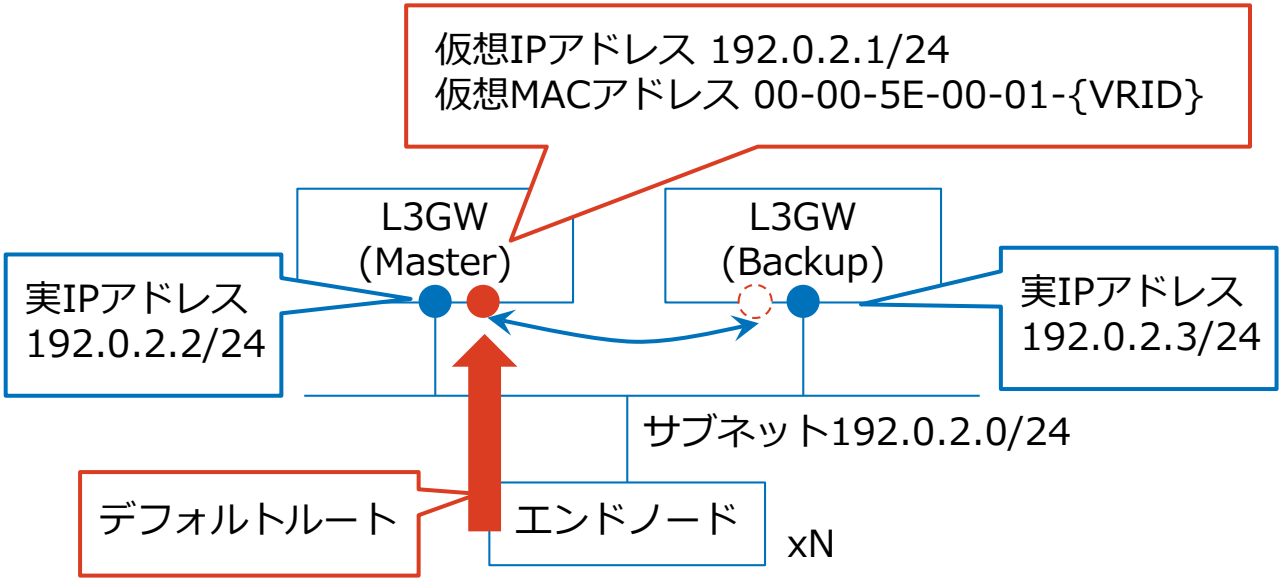
- エンドノードがなんらかのプロトコルをL3GWとしゃべる必要がある

エンドノードがその技術に対応していないと使えない

VRRP

VRRP(HSRP)

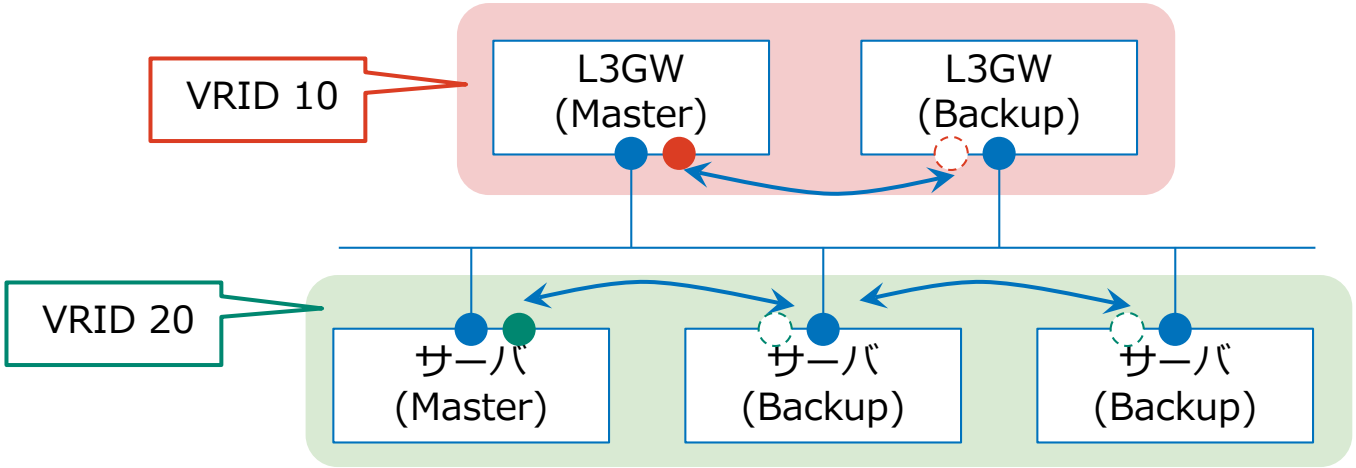
Masterに選出された機器が仮想ルータとして振る舞い、エンドノードは
仮想ルータの仮想IPアドレス/MACアドレスをゲートウェイにする方式



VRRPの使われどころ

ルータ側だけでなくサーバ側でアプリケーションのHA(High availability)を行うためにkeepalived (VRRP) + haproxy (LB) の構成で使われることもある

同じサブネットで別々の仮想ルータが動作していることを区別するために VRID (virtual router identifier) を使う



（参考）VRRPとHSRP

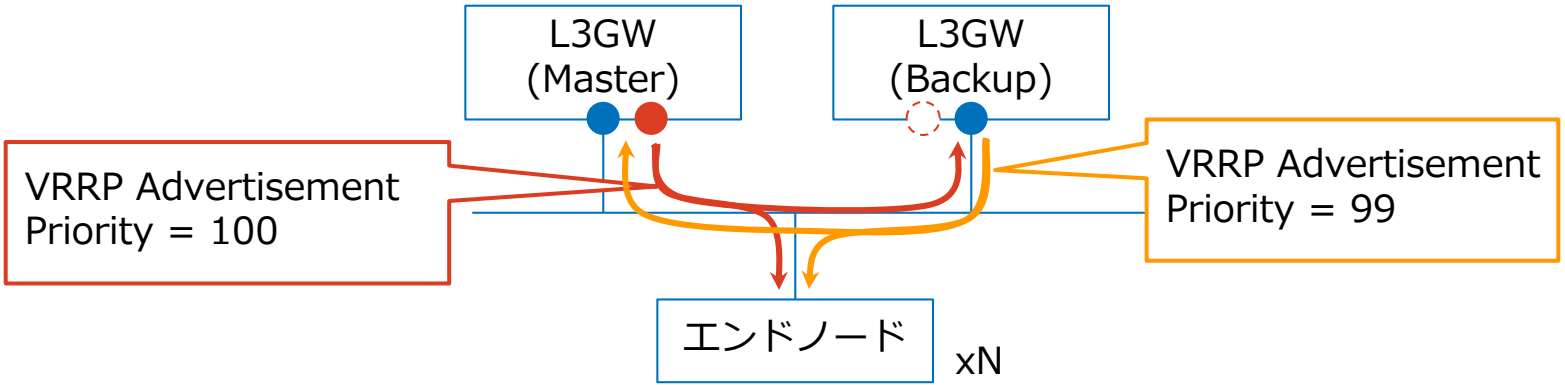
HSRPとVRRPは動作原理は基本的に同じであるが、利用するパラメータや細かい挙動が異なる

- HSRP(Hot Standby Router Protocol)はCisco社の実装
  - RFC2281にInformationalとして仕様が公開されている
- VRRP(Virtual Router Redundancy Protocol)はIETFで標準化された仕様(RFC5798)に基づく実装

Cisco社の機器でもVRRPは利用可能であるので、VRRPを覚えておけば大丈夫

VRRPの動作原理

1. VRRPでは各ルータが異なる優先度(Priority)のパラメータを持っている
2. VRRP AdvertisementにPriorityを載せ、他のルータに届くようにマルチキャストでサブネットに一定間隔で送信する
3. 受信したVRRP AdvertisementのPriorityが自身のものより高ければAdvertisementの送信を中止してBackupとして待機する
4. サブネットで一番高い優先度を持つルータがMasterとして動作する
    a. Master昇格時にGARPを送信してL2NWに仮想MACアドレスの居場所を学習させる



VRRPの特徴

超疎結合で単純な動作

- MasterからAdvertisementを投げるだけ
  - inbandのハートビート送信だけ
- 冗長グループの他のノードの状態に関知しない自律した動作
  - 自分より高いPriorityのAdvertisementが来るかどうかしか興味がない

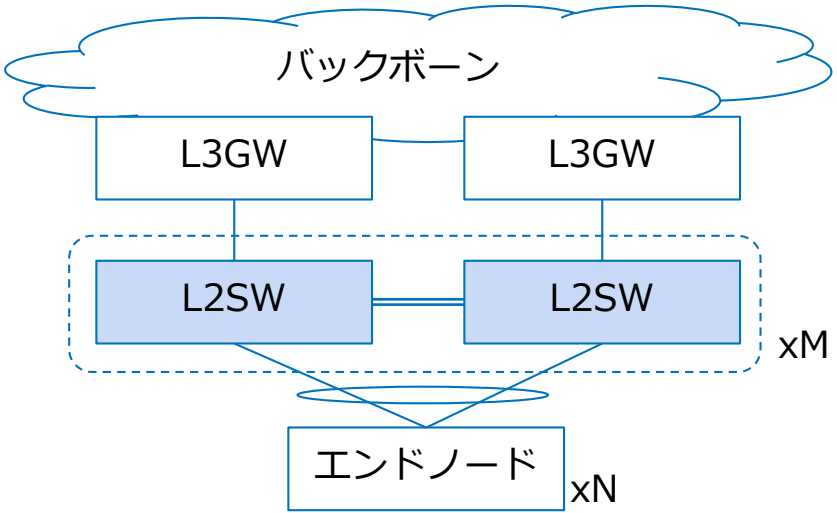
Active-Standby型

- データプレーンがActive-Standbyなのでスケールアウト出来ない
- 切り替わり時にFIBの書き込み・削除が発生する

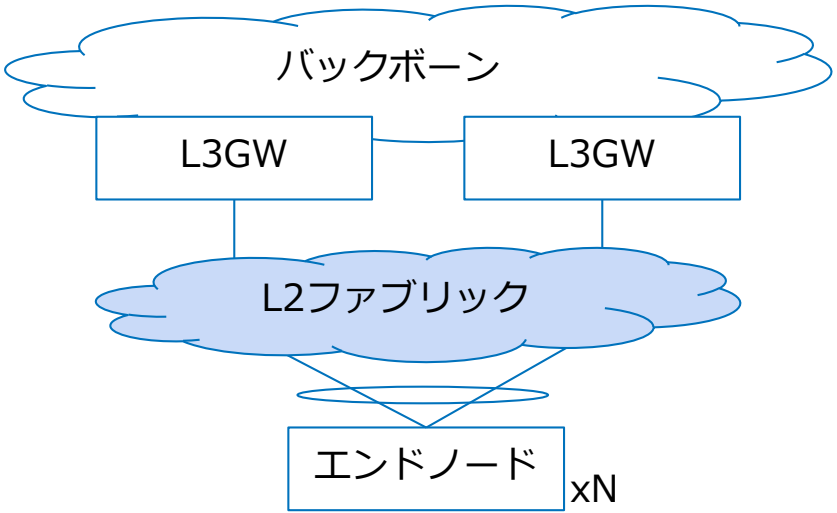
VRRPを使う場合の物理ネットワーク構成

必ずL2ネットワークが必要になるためL3GWとエンドノードの間にL2スイッチやL2ファブリックを挟む必要がある

➡ L2NWの冗長性を考える必要がある



ペアのL2スイッチを並べていく
Virtual Chassis / MC-LAG



L2ファブリックに接続する
STP / TRILL / OpenFlow / EVPN

VRRPの問題点

必ずL2が必要になる

- L2NWの冗長構成を併せて考える必要がある
- L2NWの分断やサイレント障害によりスプリットブレイン(Master-Master)状態になることが原因で長時間の通信障害が起きやすい

マルチキャスト&ブロードキャストを使う

- L2NW部分でループするとメルトダウンする
- Master↔Backup状態のフラップが起こるとGARPとAdvertisementを大量に投げることで他のL3機器に高負荷を与える
  - ○ Routing Engineが死んでGWがARP学習できなくなる場合も (JANOG44の発表参照)

L2NWでマルチキャストが「正しく」届くことを保証するのが難しい

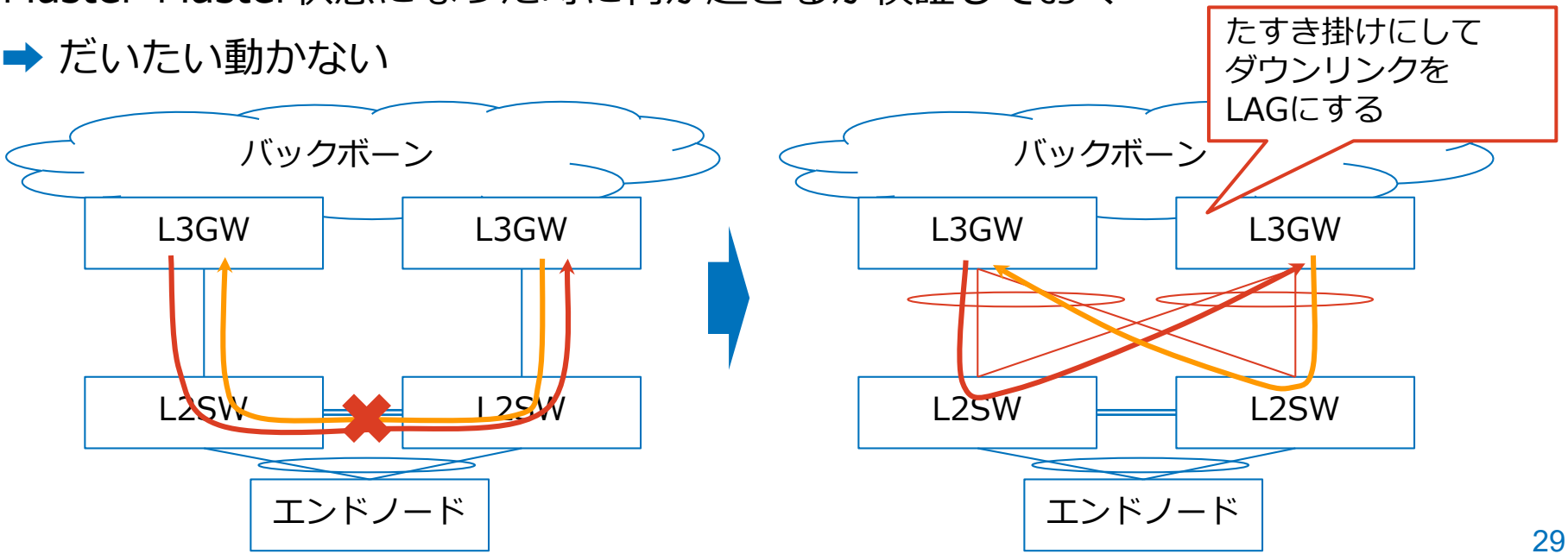
VRRPを使う場合

AdvertisementがL2ファブリックに入らずにローカルで折り返せる構成がベスト

➡ Virtual ChassisやLocal Biasが有効なMC-LAGなど

Master-Master状態になった時に何が起きるか検証しておく

➡ だいたい動かない



従来の代表的なFirst-Hop Redanduncyの実現手法

VRRPはやめたい

#	ネットワーク機器	エンドノード
~~1~~	~~VRRP(HSRP)~~	~~bonding~~
2	Virtual Chassis	bonding
3	MC-LAG	bonding (LACP)
4	BGP	BGP

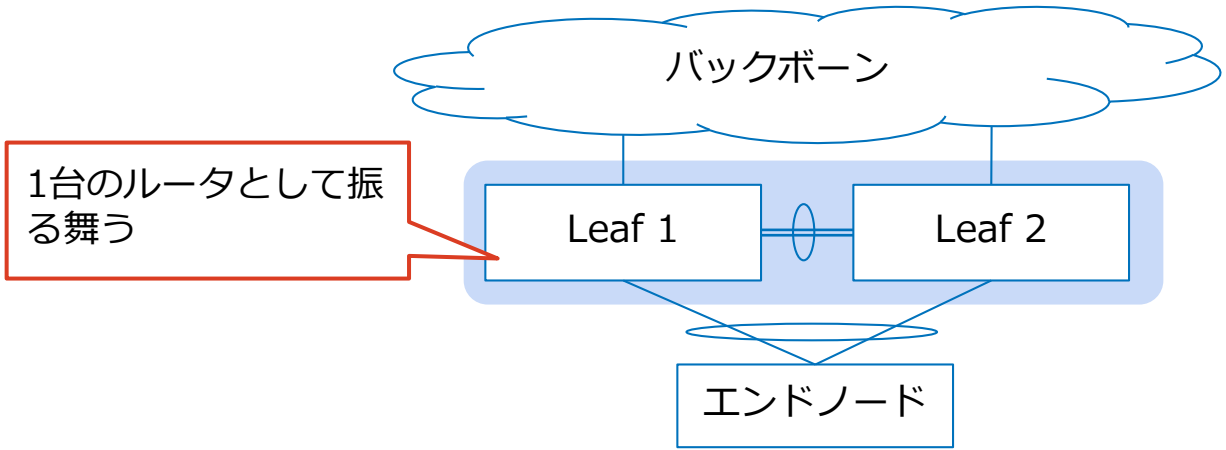
Virtual Chassis

Virtual Chassis

複数台(だいたい2台)の機器を論理的に 1 つの機器に見せる方式

各ベンダーの独自の実装で実現されている

- Juniper QFX/EX: VC (Virtual Chassis)
- Cisco Catalyst: VSS (Virtual Switching System)

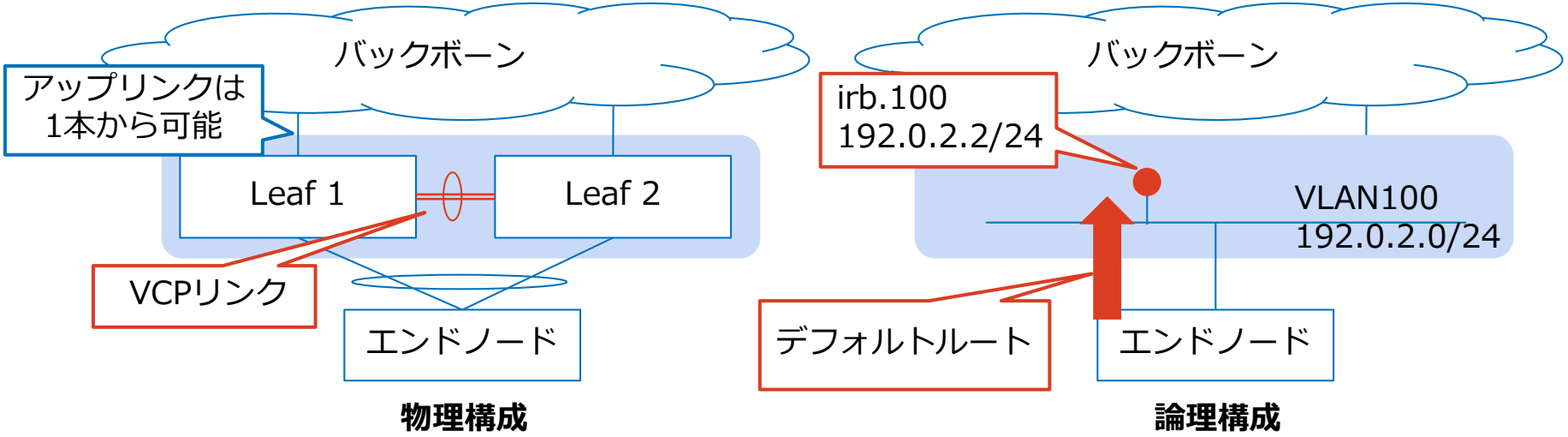


Virtual Chassisの構成

筐体間をLAGでI/F冗長したピアリンク (VCPリンクやVSL)で接続する

ダウンリンクは単なるLAGインターフェイス(ae)に見える

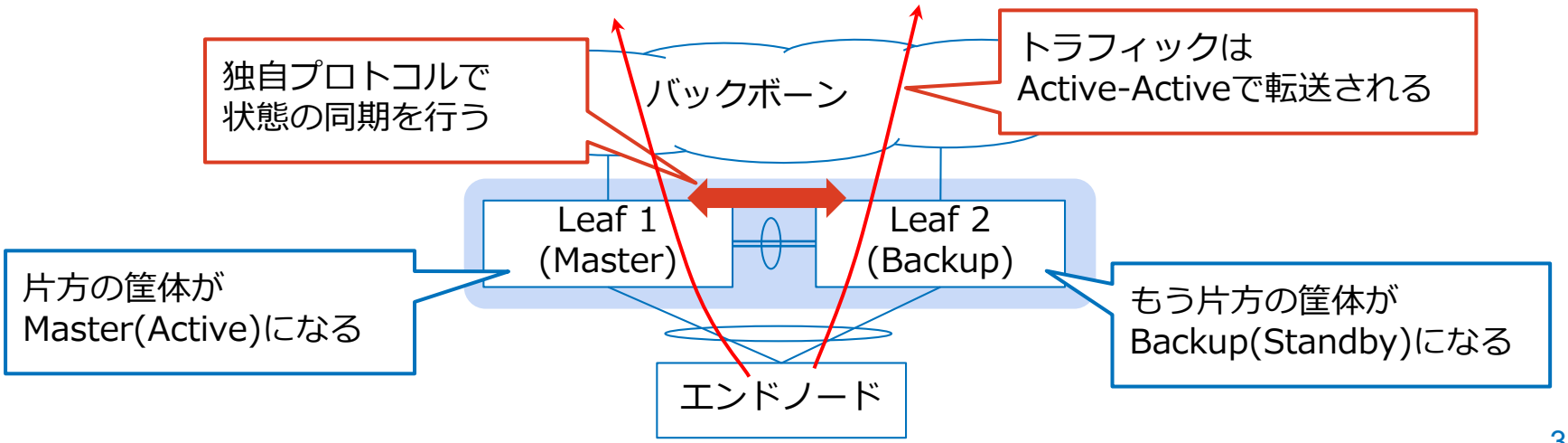
複数のエンドノードを同じサブネットに所属させる場合はダウンリンクを
スイッチポートにしてIRBインターフェイスにゲートウェイアドレスを設定する



Virtual Chassisの特徴

超密結合：論理的に１つの筐体として扱えるので、設定も運用も楽

- データプレーンはActive-Activeで動作する
- コントロールプレーンはActive-Standbyで動作する



Virtual Chassisの問題点

論理的に１つに見せる仕組み

- 論理的な単一障害点になり得る

超密結合で複雑性をNOS内部に閉じ込めた仕組み

- 複雑なのでバグが多かった
  - 片方が死ぬともう片方が引きづられて死ぬパターンが多い
  - StandbyがActiveに昇格した時にうまく動かなくて死ぬパターンもある
- NSSU (Nonstop software upgrade)で1つも問題を引かないことは稀
  - 「バージョンアップに失敗しないようにするためにはバグが修正されたバージョンにバージョンアップしてください」
- In serviceでメジャーバージョンアップ出来ないケースもある
  - メジャーバージョン間で後方互換のないスキーマの変更があった時に、NSSUで片方のバージョンアップが終わったタイミングで状態の同期ができなくなることがあり、両方の筐体を同時に停止してバージョンアップする必要がある

従来の代表的なFirst-Hop Redanduncyの実現手法

Virtual Chassisもやめて疎結合でバージョンアップできるようにしたい

#	ネットワーク機器	エンドノード
~~1~~	~~VRRP(HSRP)~~	~~bonding~~
~~2~~	~~Virtual Chassis~~	~~bonding~~
3	MC-LAG	bonding (LACP)
4	BGP	BGP

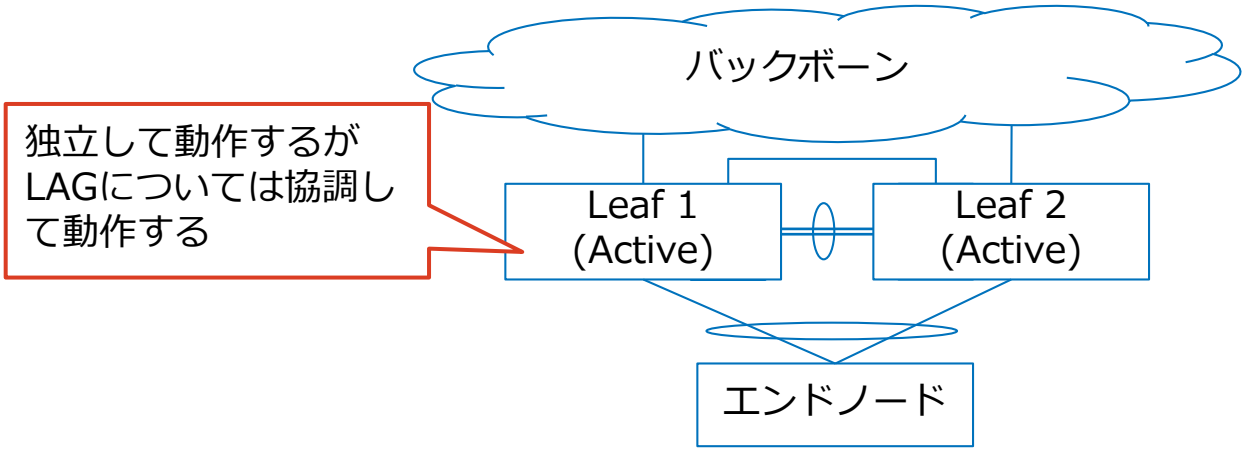
MC-LAG

MC-LAG（マルチシャーシLAG）

論理的に独立した２つの筐体が協調してLAG動作させる方式

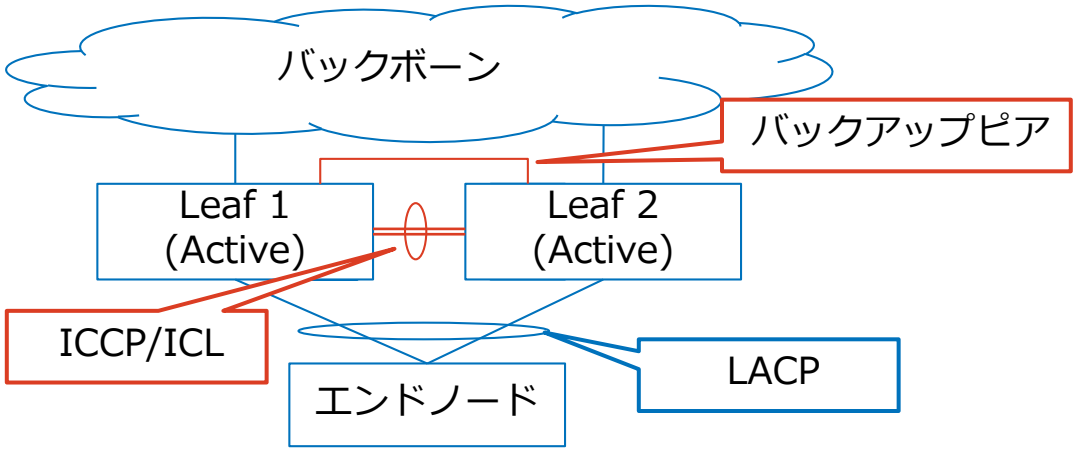
各ベンダーの独自の実装で実現されている (一応IEEE 802.1AX-2008がある)

- Juniper: MC-LAG
- Cisco Nexus: vPC
- Arista: MLAG



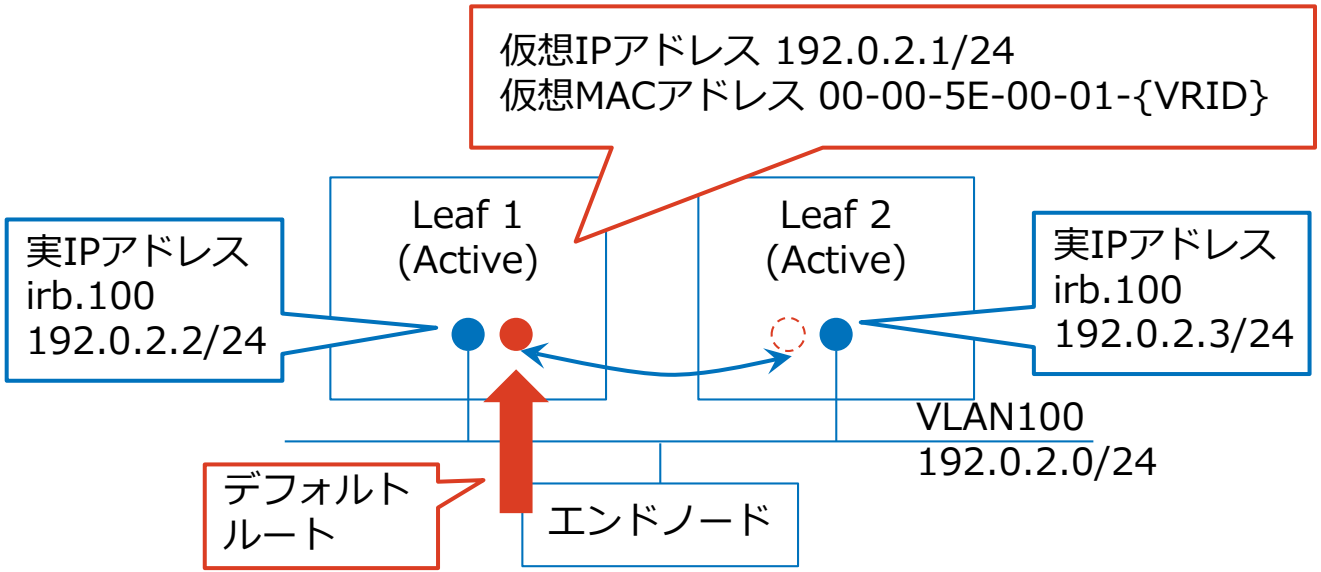
MC-LAG（マルチシャーシLAG）の構成（物理・L2）

- 筐体間をピアリンクで接続する
  - ○ LAGでインターフェイス冗長を確保する
  - ○ ICCP (Inter-Chassis Control Protocol)で制御情報をやりとりする
  - ○ ICL (InterChassis Link)でトラフィックを転送する
- Split Brain状態になったときのために管理網経由でバックアップピアを張る
- LAGの識別のためにLACPが必要になる



MC-LAG（マルチシャーシLAG）の構成（L3）

- それぞれにIRB(SVI)を作成してVRRPで冗長を取る
  - ○ VRRP AdvertisementはICLを通る
  - ○ Juniper MC-LAGの場合はMACアドレス同期を行う方法もあるが、こちらが推奨されている



MC-LAG（マルチシャーシLAG）の問題点

L3GWにしようと思うと結局VRRP over IRB

- ダウンリンク側にL2NWがなくても動くので素のVRRPよりマシ
- バックアップピアでスプリットブレイン時の対処もできるはずなので素のVRRPよりマシ
- VRRP Advertisementの通るパスがトラフィックが通るパスと異なる
  - ○ Virtual Chassis同様、Out-of-band監視になる
  - ○ これが原因で障害になった事例ってあります？

従来の代表的なFirst-Hop Redanduncyの実現手法

MC-LAGは実質VRRPだった

#	ネットワーク機器	エンドノード
~~1~~	~~VRRP(HSRP)~~	~~bonding~~
~~2~~	~~Virtual Chassis~~	~~bonding~~
~~3~~	~~MC-LAG~~	~~bonding (LACP)~~
4	BGP	BGP

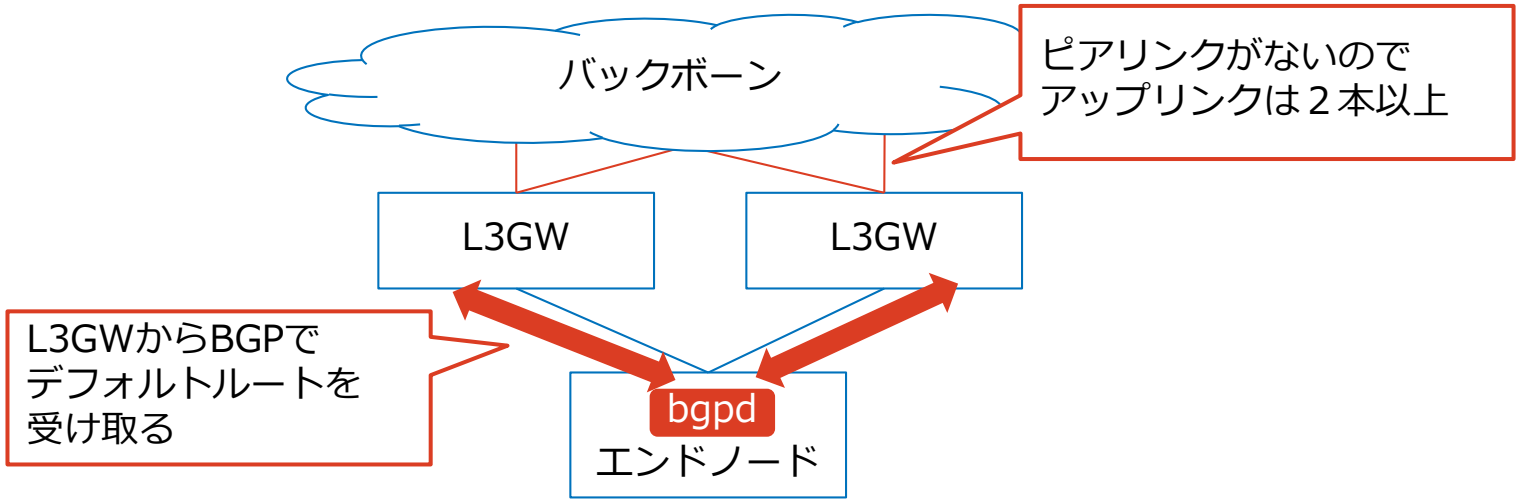
BGP

BGP

エンドノードでBGPスピーカを動作させて、ネットワーク機器からデフォルトルートを動的に受け取る方式

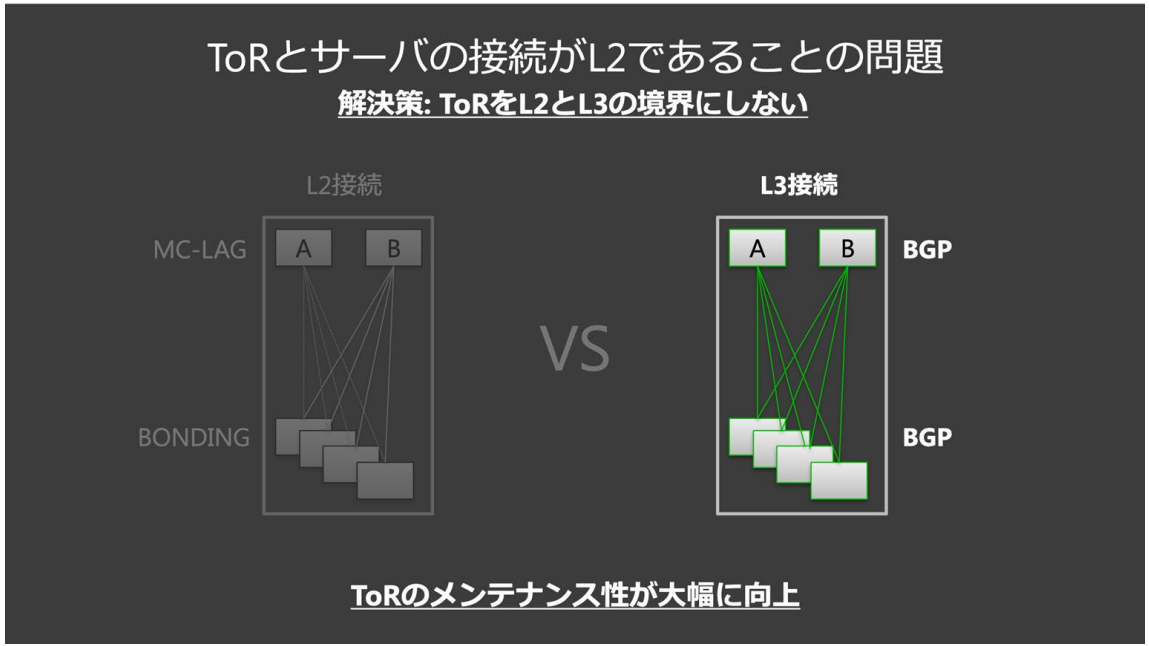
静的にデフォゲを設定しないというパラダイムシフト

コンテンツプロバイダのデータセンターでいま流行りの構成



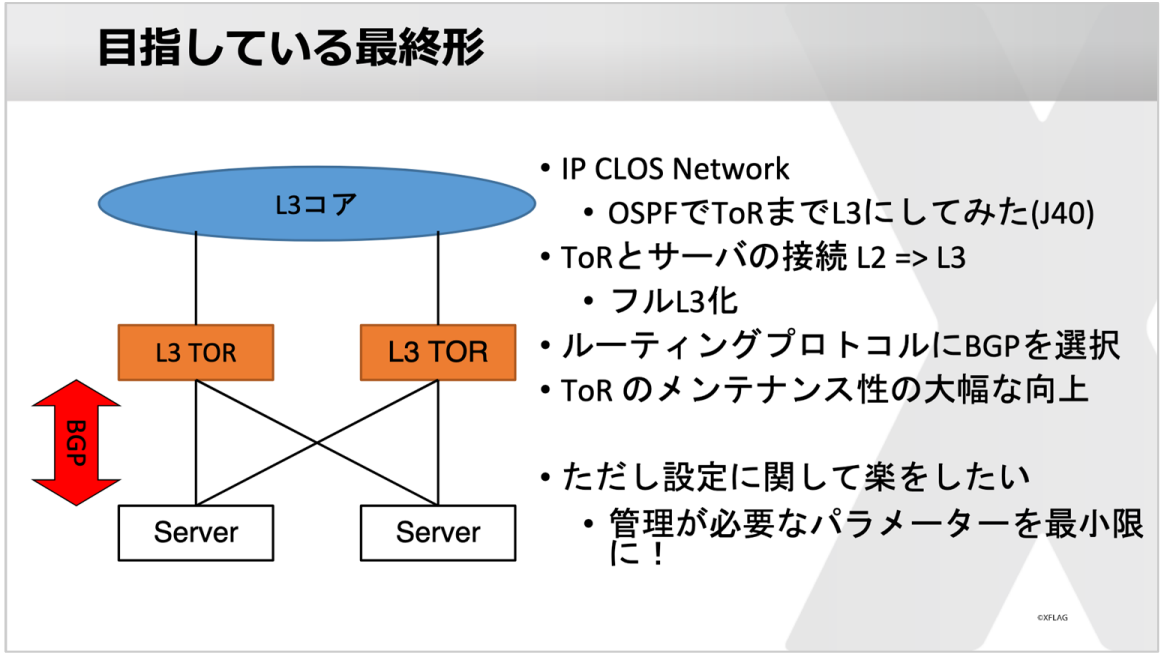
LINEさんの事例

JANOG43 『LINEのネットワークをゼロから再設計した話』



ミクシィさんの事例

JANOG44 『目指せ！Goodbye IPv4on L3ToR』



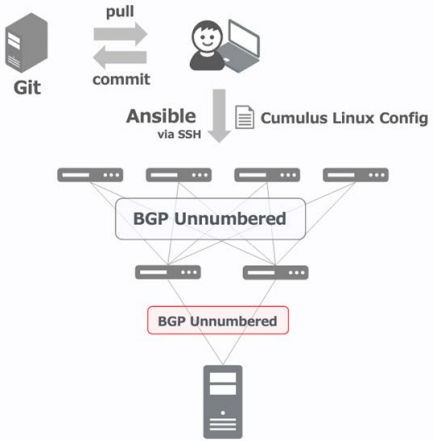
Yahoo! JAPANさんの事例

JANOG46 『ヤフーの IP Closネットワークの歴史と運用』

Gen4.1: Ansible, ALL-L3 IP Clos, 2019~

P 32

- Private Cloud VM用
- Ansible
- NW機器
  - 全て Box 型 Whitebox (Cumulus)
    - Chassis スイッチ → Boxスイッチ
    - 大幅なコスト減
- EVPN/VXLAN 廃止
- サーバまで ALL-L3 接続



SiteOPS
Yahoo! JAPAN Site Operations

BGPの良い点

コントロールプレーンがL4以上でデータプレーンが同じパスを通る

- 片方Leafやそのリンクに異常が発生すればBGPが落ちて、
  トラフィックはもう片方のLeafに自動的に迂回される

ネットワーク側に異常があるとエンドノード側で動的に迂回がかけられる

- 片方のLeafのアップリンクからデフォルト経路が来なくなったら、
  エンドノードにもデフォルト経路が配信されなくなるので、
  トラフィックはもう片方のLeafに自動的に迂回される

BGPが使えるならBGPがベスト（個人の意見です）

BGPの導入を阻むもの

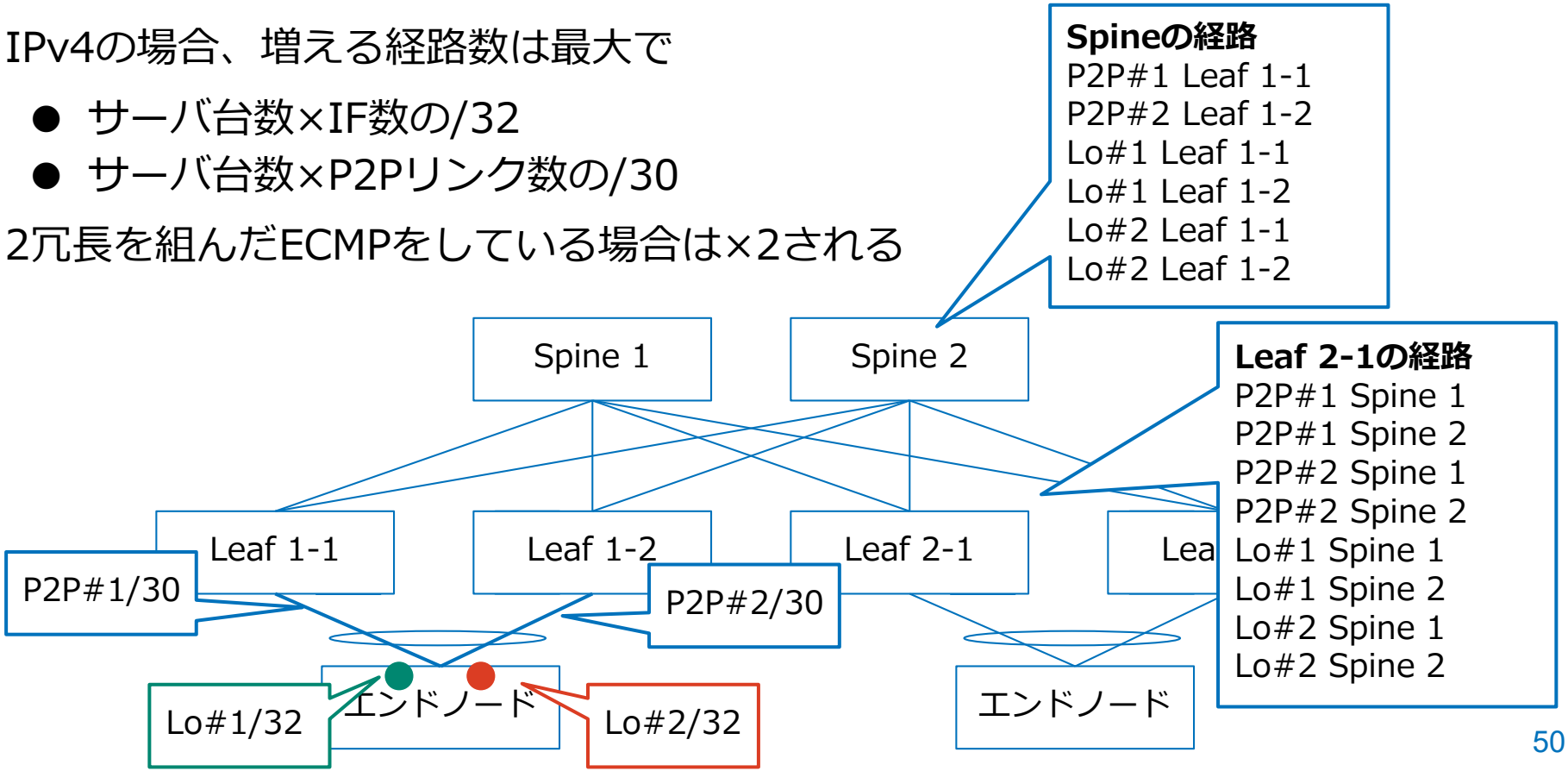
- ストレージ系のアプライアンスで対応していないことがある
  - ○ **みなさんストレージどうしてるんですかね？**
- サーバ側のアプリケーションが…
  - ○ そもそもBGPを喋っていてTCP179が埋まっている
  - ○ Linuxカーネルの持つ経路の動的な変更に追従しない
- P2Pリンクがたくさん必要になる
  - ○ IPv4だとアドレスを浪費するのでIPv6やunnumberedなどの対応が必要になる
- L2セグメントが必要な機器を収容できない
  - ○ HAやクラスタリングをマルチキャストを使って実現するアプリケーション
  - ○ source NATしているLB

（参考）バックボーンへの影響

IPv4の場合、増える経路数は最大で

- サーバ台数×IF数の/32
- サーバ台数×P2Pリンク数の/30

2冗長を組んだECMPをしている場合は×2される



（参考）バックボーンへの影響を軽減する方法

P2Pリンクのサブネットを広告しない or Unnumberedにする

➡ いくつかの運用上の課題を解決する必要がある

LeafでエンドノードのもつアドレスをAggregateする

➡ アドレス設計を上手にしないといけない

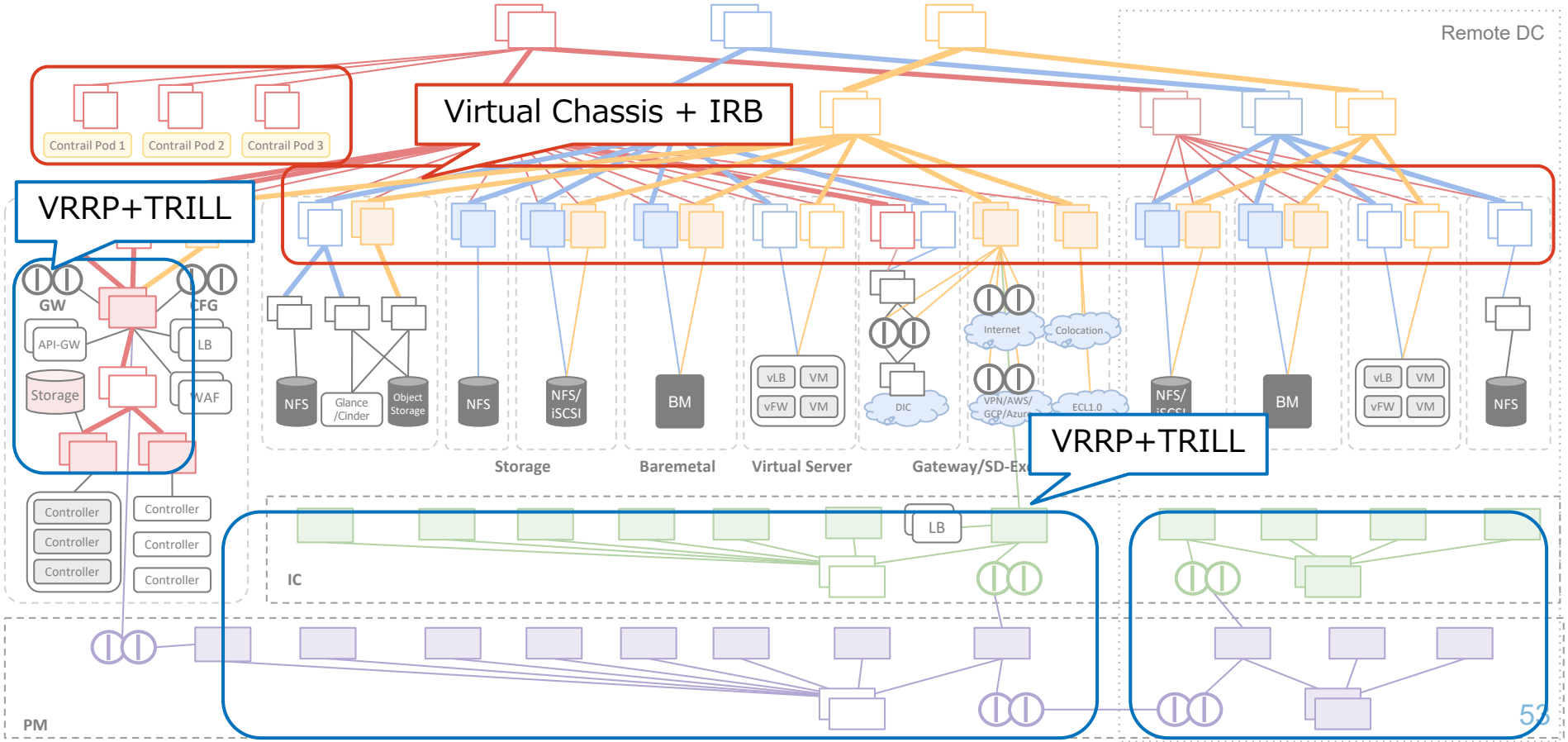
くわしくはRFC7938で！（翻訳しました）

RFC7938 - Use of BGP for Routing in Large-Scale Data Centers

RFC7938 - 大規模データセンター内でのルーティングのためのBGPの利用方法

First-Hop Redundancy再考

ECL2.0 Network Overview



Enterprise Cloud 2.0 (IaaS事業者)の接続パターン

		自社機器							お客様機器
		自社利用			他社リソース収容			他社専有	自社仕様
		サーバ	ストレージ	NW機器	サーバ	ストレージ	NW機器	サーバ	NW機器
P2P L3	LAG	SDNコントローラ							
P2MP L3	LAG	コントローラ	内部用ストレージ	NAT GW LB/FW	VM用HV、コントローラ	VM用ストレージ			
L2オーバーレイ	LAG					テナント用ストレージ	GW		コロケーション接続
	bond mode 1							ベアメタルサーバ	
P2MP L3 管理網	LAG					テナント用ストレージ			
	単一I/F	コントローラ	内部用ストレージ	各NW機器	VM用HV、コントローラ	テナント用ストレージ	各NW機器	ベアメタルサーバ	

伝送あり

VC IRB　VC VxLAN　VRRP + TRILL

Requirement

ネットワーク機器側の機器種別×I/Fパターン数をふやしたくない

- 検証パターンが増える
  - ○ I/Fパターンがn個だと通信パターンだけで n^2 通り。
- 運用パターンが増える
- 踏むバグのパターンが増える

➡ エンドノードに応じて使い分けをしたくない

BGPは使えるか？

ストレージでBGPしゃべれない子がいる…

- しゃべれる子もいたけど自信なさそう

サーバでもBGPしゃべれない子がいる…

- Linuxの動的なルーティングテーブルに従わないSDN仮想ルータ
  - コード書いてコントリビューションすればなんとかなる
- オーバーレイのBGPを喋っていてTCP179を既に使ってるSDNコントローラ
  - ポート番号変えればなんとかなｒ……変えられない！？

従来の代表的なFirst-Hop Redanduncyの実現手法

方式を統一したいのにエンドノードにBGPをしゃべれない子がいた

#	ネットワーク機器	エンドノード
1	VRRP(HSRP)	bonding
2	Virtual Chassis	bonding
3	MC-LAG	bonding (LACP)
4	BGP	BGP

万策尽きたか…？

シリコンバレーへ

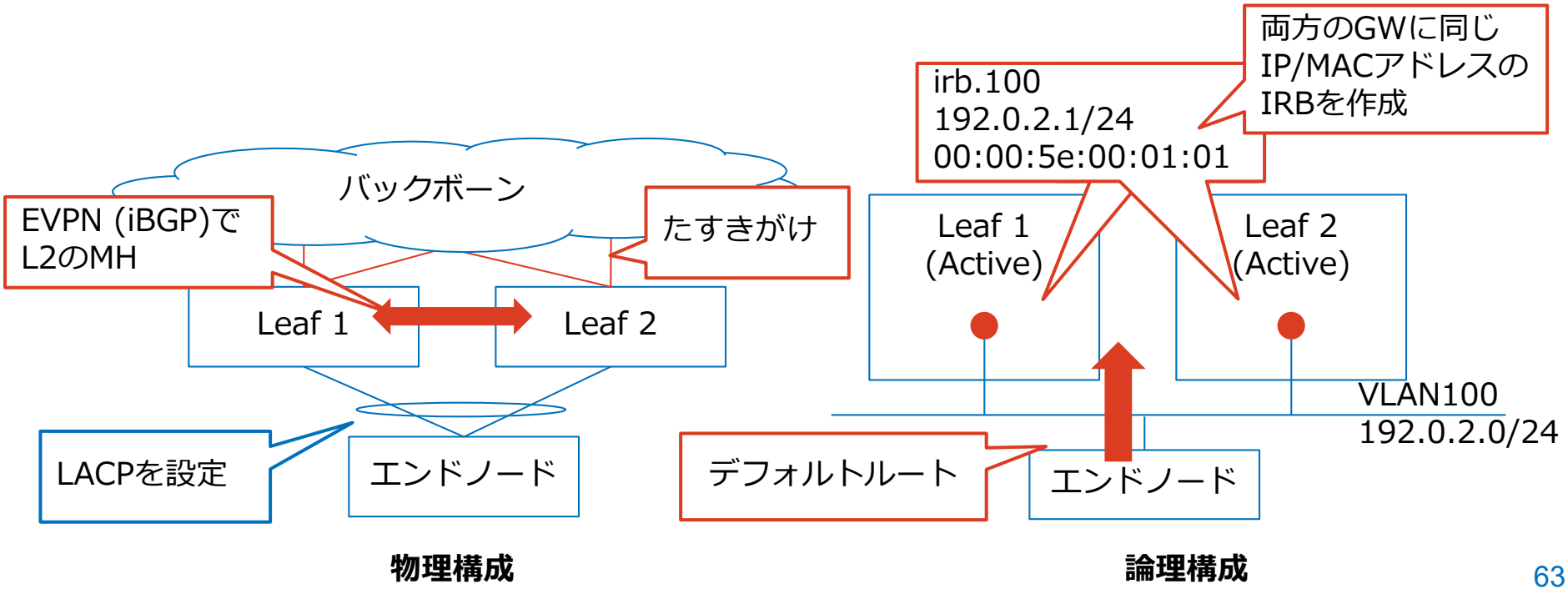
EVPN Anycast Gatewayってのがあるから使ってみない？

EVPN Anycast Gateway ！ ？

EVPN Anycast Gateway

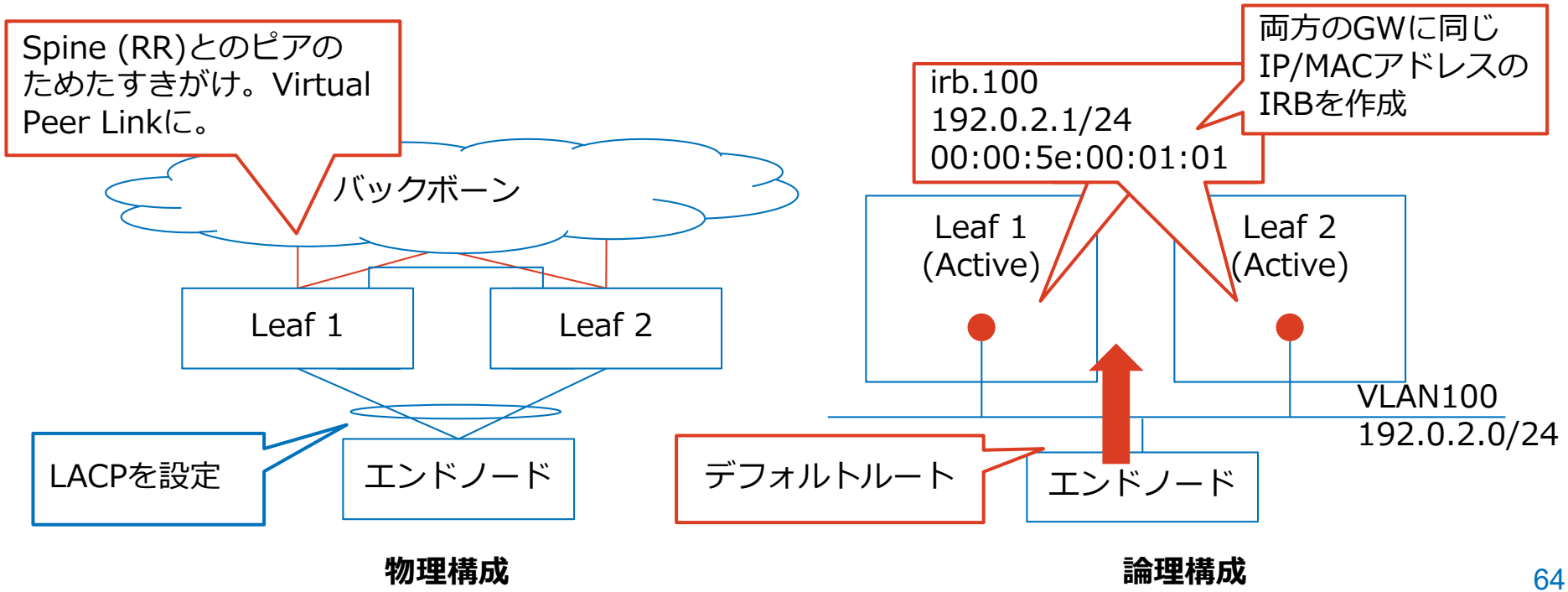
EVPN Multihoming & EVPN Anycast Gateway

EVPN Multihoming(MH)の機能を使ってLAGでL2冗長を行い、
EVPN Anycast Gateway(AG)の機能を使ってAct-ActのL3GWを実現する方式



EVPN + MC-LAG & EVPN Anycast Gateway

MC-LAGの機能を使ってL2冗長を行い、
EVPN Anycast Gatewayの機能を使ってAct-ActのL3GWを実現する方式



EVPN Multihoming or EVPN + MC-LAG ?

どちらもEVPNのL2オーバーレイのLeafで独立した筐体間でLAGを組む技術

ベンダーによって方向性が違う

- JuniperはEVPN Multihoming推し
- Cisco NexusはMC-LAG (vPC)推し
- Cumulus LinuxはMC-LAG (CLAG)のみの実装だったが、2020-07リリースの4.2でEVPN Multihomingに対応した

➡ **採用する機器で推している・対応している方を使えばよい**

※ 個人的にはEVPN Multihoming推し

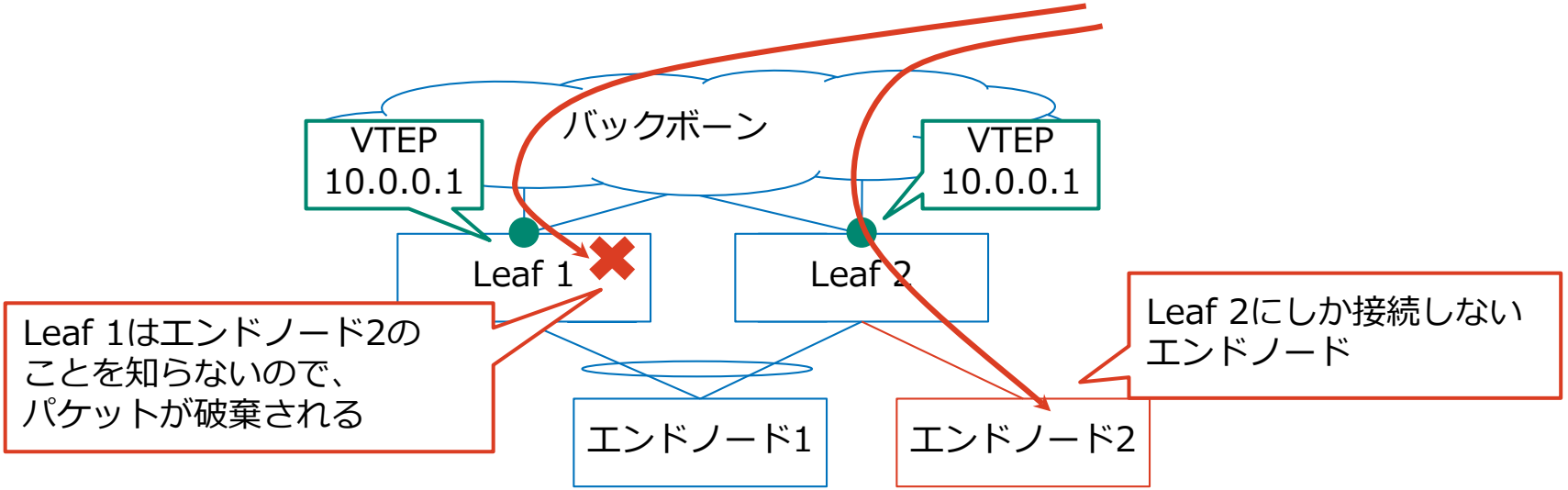
- EVPN + vPC 構成ではOrphan Port問題が厄介

（参考）EVPN VXLAN + Orphan Port問題

Orphan Port: vPCの片方向にしか接続していないポート

vPCでは両Leafで同じVTEPアドレスを使うため、エンドノード2宛のVXLANパケットがLeaf 1にも届いてしまうが、Leaf 1はエンドノード2に転送できない

➡ 物理ピアリンクと少しトリッキーな設定が必要になる



（参考）EVPN Anycast Gatewayの要件

ハードウェアがVXLAN Routingに対応している必要がある

Broadcom社のチップではTrident 2+以降（Trident 2は非対応）

EVPN MH&AGの良いところ

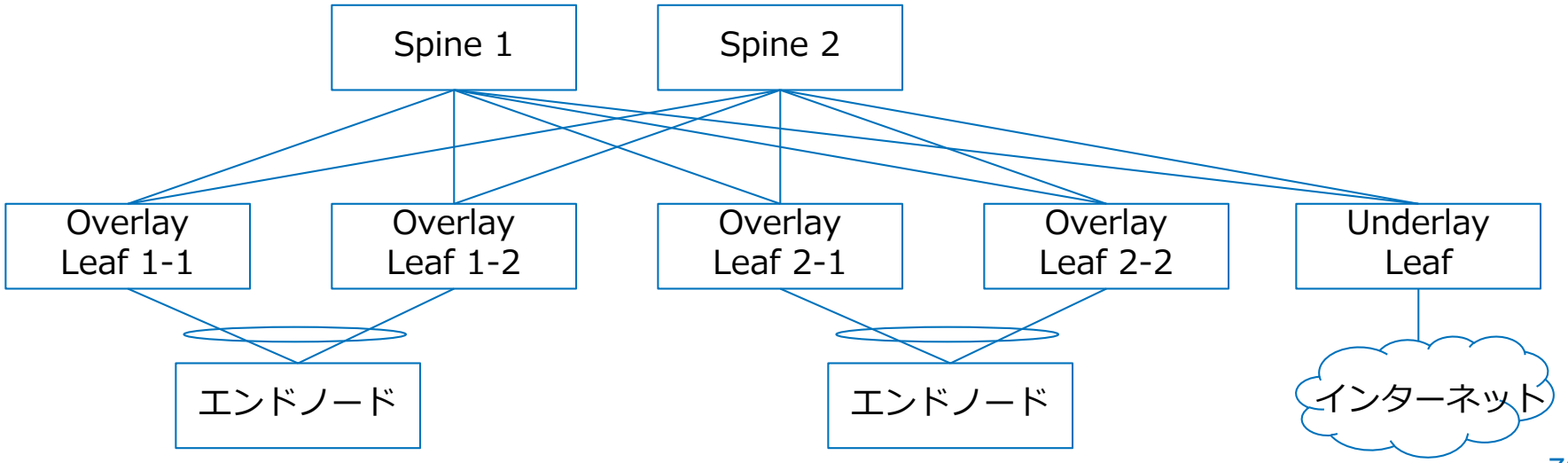
- コントロールプレーンは標準技術で構成されている
- EVPNなのでL2/L3オーバーレイに対応している
  - 我々の場合、テナント用オーバーレイがEVPN/VXLANなので
    同じ技術でアンダーレイのFirst-Hop Redundancyが実現できる
- コントロールプレーンもデータプレーンも独立していてActive-Active構成
  - バージョンアップしやすい
  - VRRPのように仮想IP/MACアドレスの追加・削除（FIBの更新）が起きない

EVPN Multihoming & Anycast Gatewayの動作イメージ

※あくまでざっと感覚をつかむためのイメージで技術的に的確ではないです

EVPN MH&AGの動作イメージ (1/13)

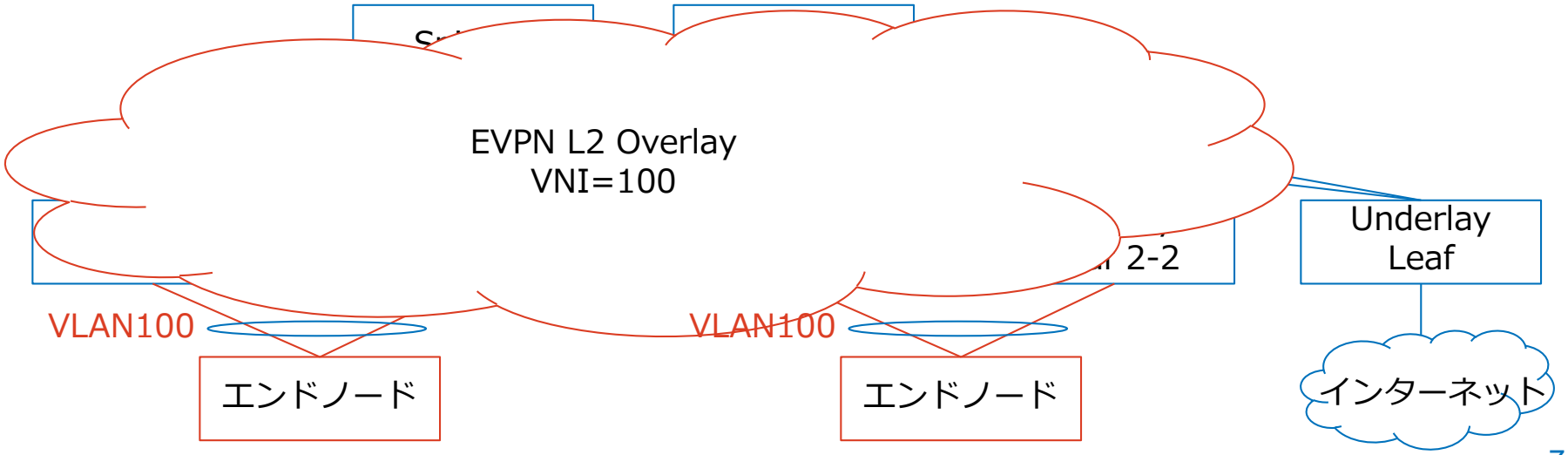
よくあるIP CLOSネットワーク



EVPN MH&AGの動作イメージ (2/13)

普通のEVPNのL2オーバーレイネットワークをつくる

エンドノードはEVPN Multihomingを使ってLAGで接続する

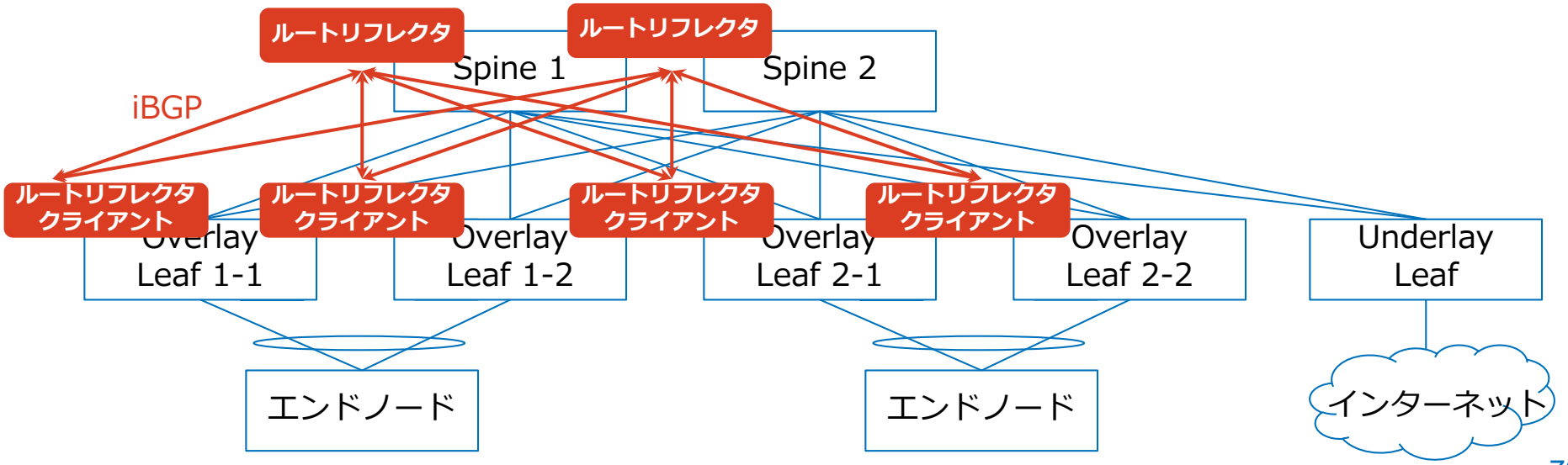


EVPN MH&AGの動作イメージ (3/13)

Spineをルートリフレクタにする構成が一般的

iBGPはループバックアドレスを使って張る

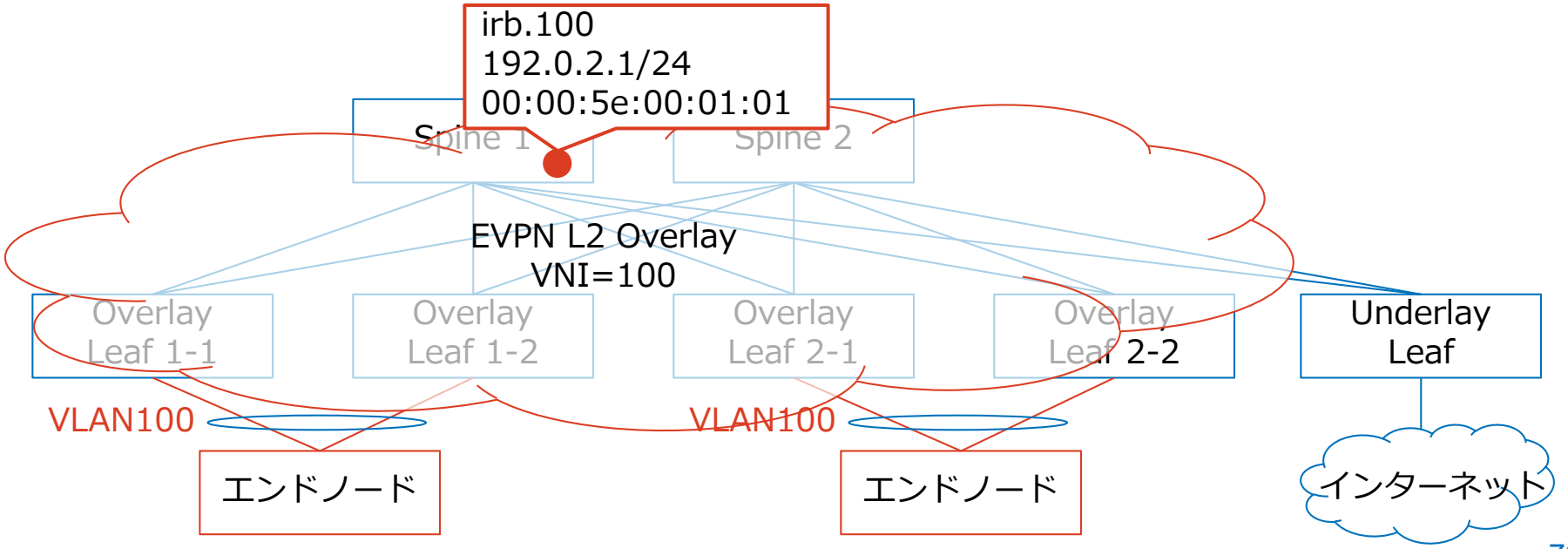
- I/FアドレスはIP CLOSのeBGPで使われているため使えない



EVPN MH&AGの動作イメージ (4/13)

IRBインターフェイスを生やしてVNI 100のL3 Interfaceにする

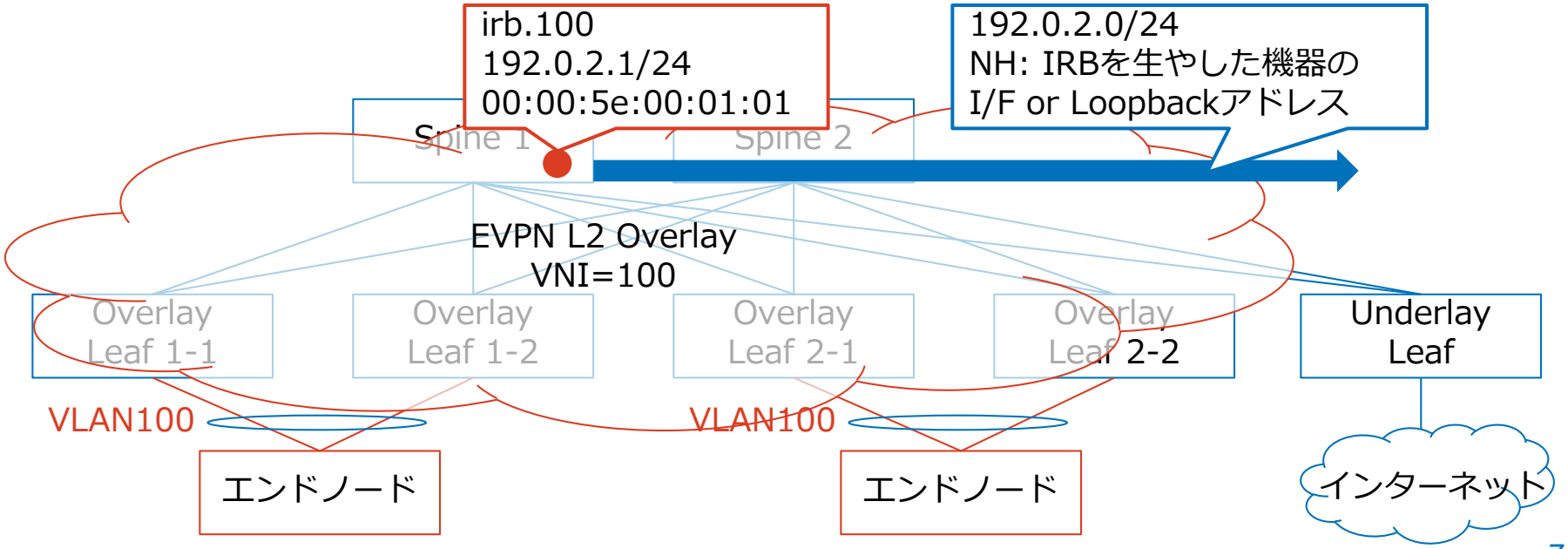
生やす場所はiBGPでEVPN経路を交換しているどこか：たとえばSpine 1



EVPN MH&AGの動作イメージ (5/13)

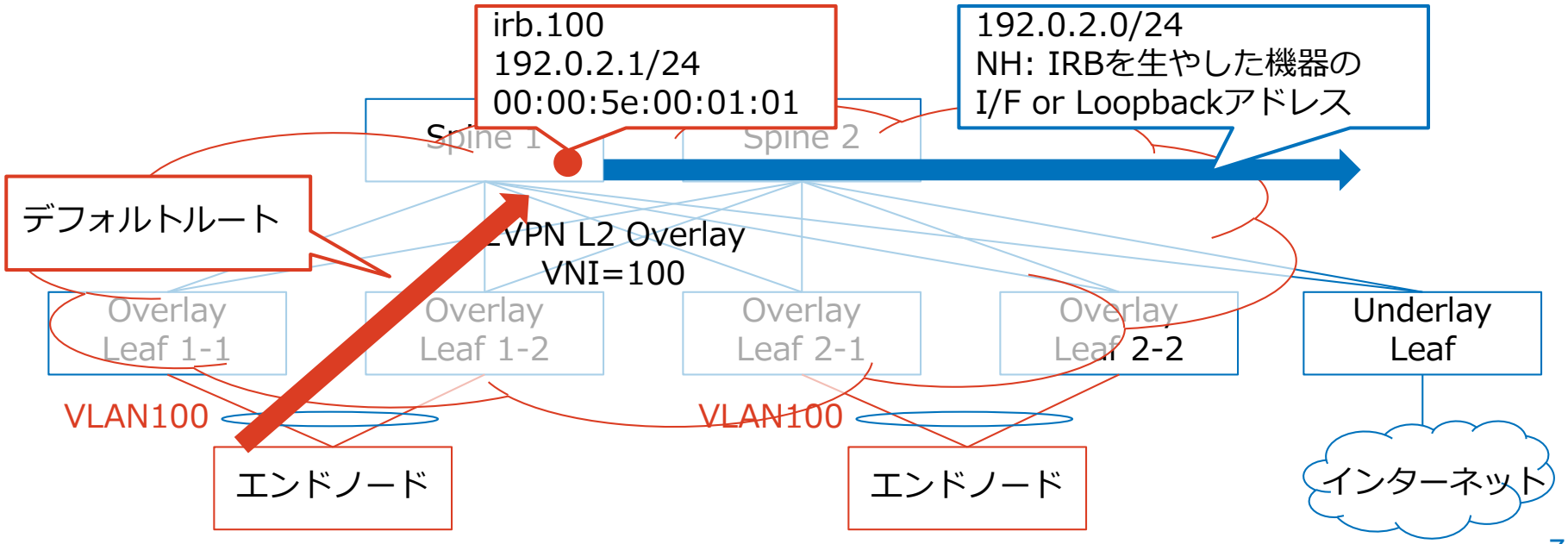
irb.100のサブネットをアンダーレイのIP CLOSにeBGPで広告する

ネクストホップはIRBを生やした機器のI/FまたはLoopbackアドレス



EVPN MH&AGの動作イメージ (6/13)

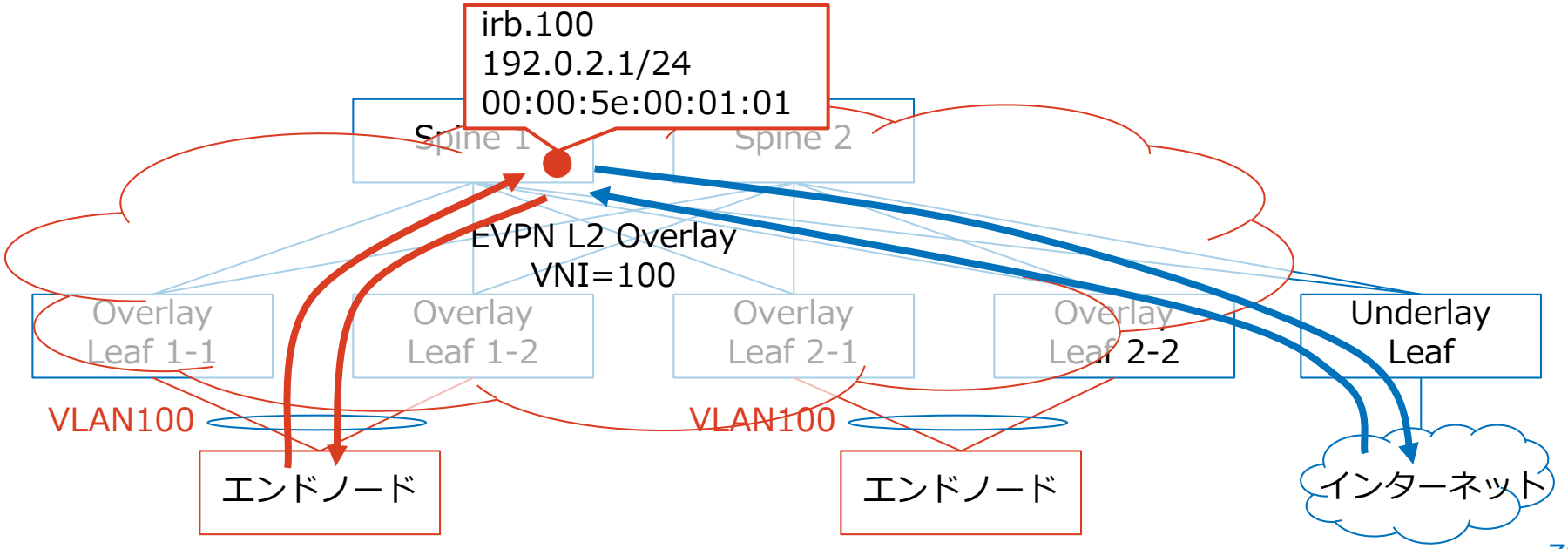
エンドノードのデフォルトルートをIRBに向ける



EVPN MH&AGの動作イメージ (7/13)

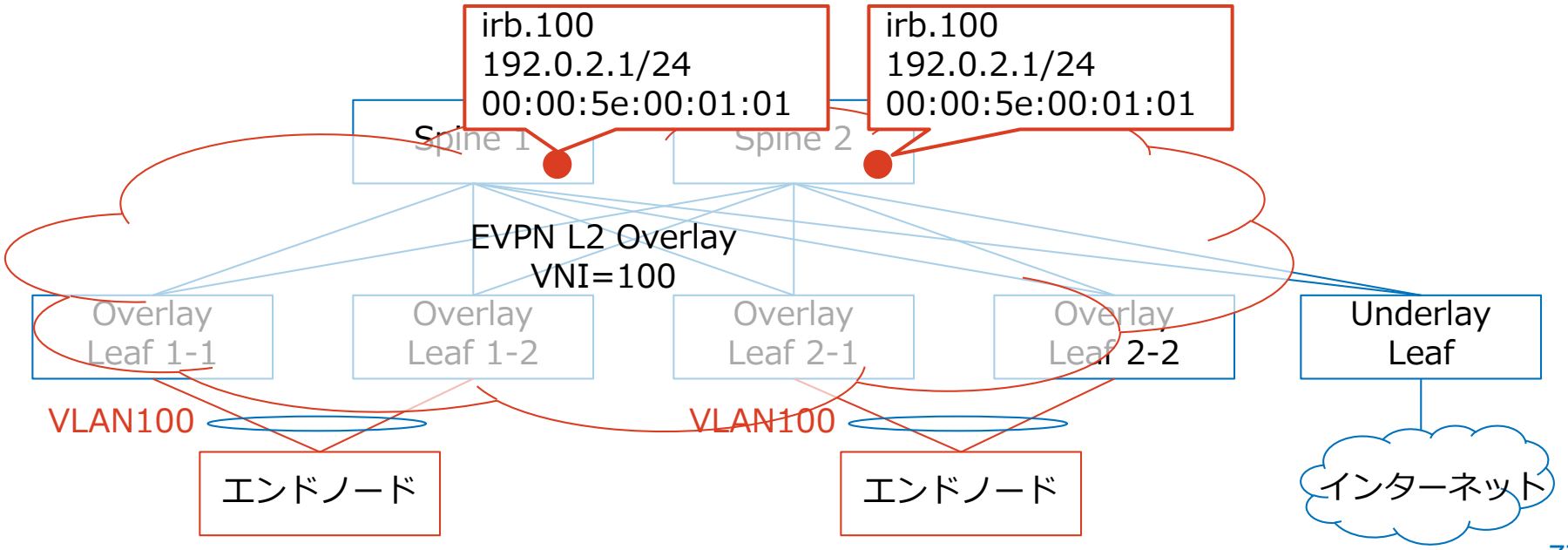
IRBインターフェイス経由でインターネットと通信ができるようになる

Spine 1のirb.100でアンダーレイに乗り換える



EVPN MH&AGの動作イメージ (8/13)

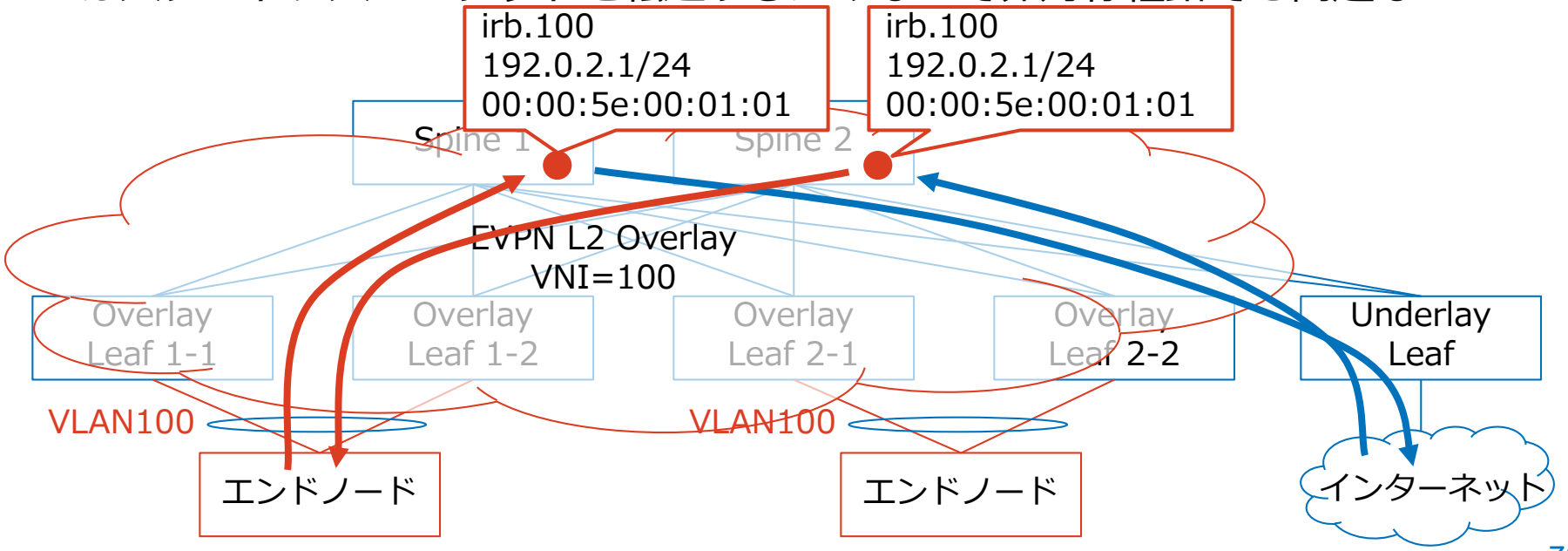
Spine 2にもIRBインタフェイスを生やして同じIP/MACアドレスを与える



EVPN MH&AGの動作イメージ (9/13)

EVPNのL2オーバーレイもアンダーレイも、BGP Anycast + ECMPで
ロードバランスされていずれかのIRBインターフェイスに到達できる

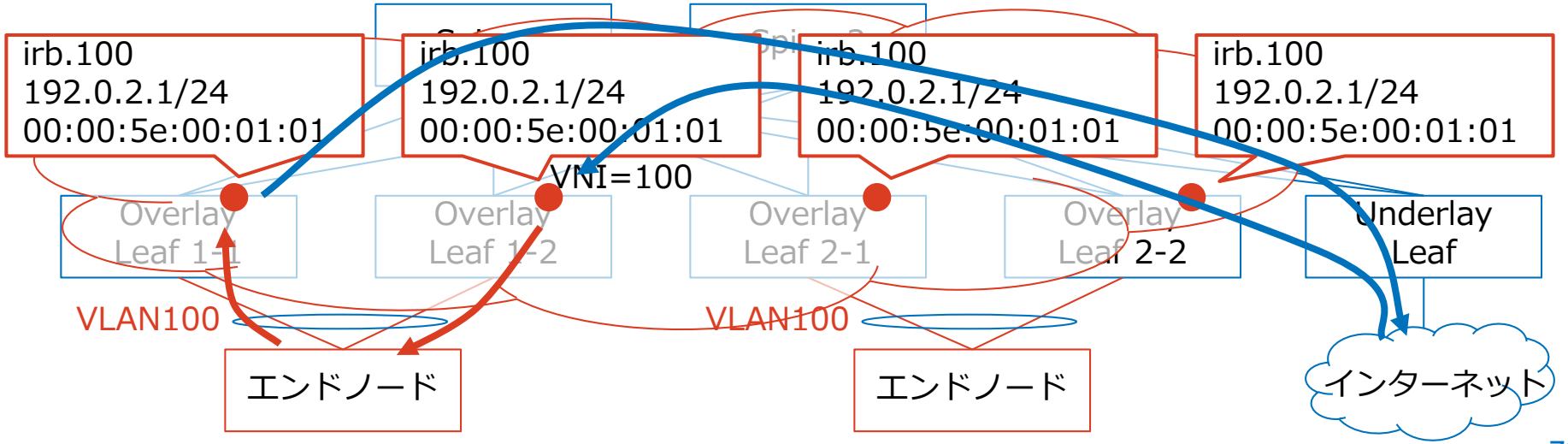
IRBはステートレスにパケットを転送するだけなので非対称経路でも問題ない



EVPN MH&AGの動作イメージ (10/13)

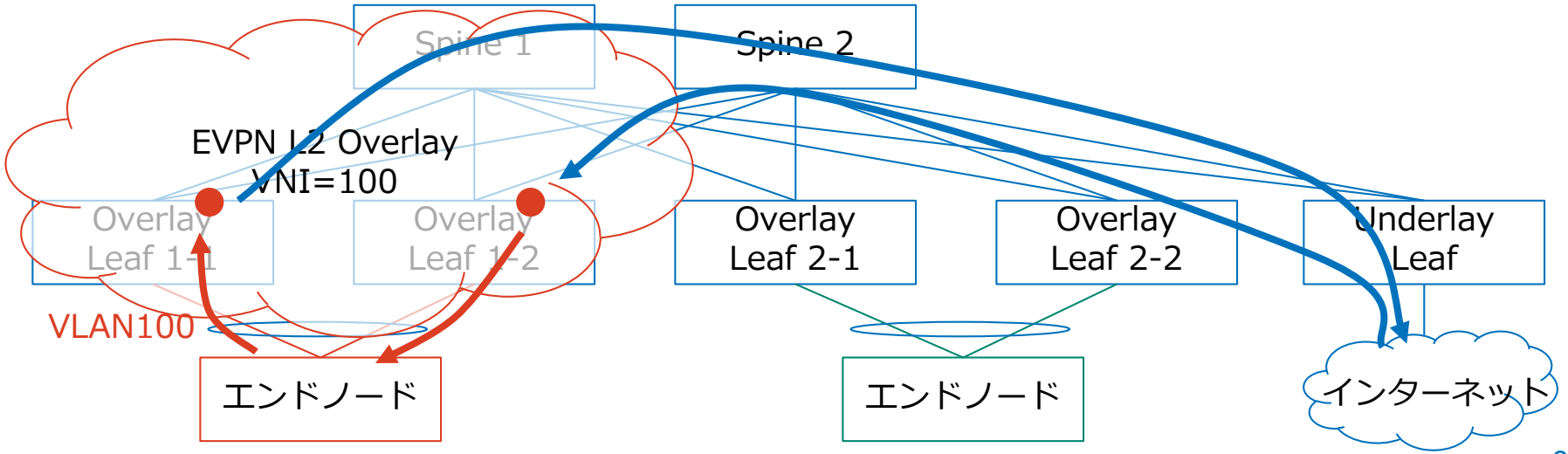
IRBインターフェイスはLeafに作ることもできる

- エンドノード→IRBはLAGのハッシュで分散する
- IRB→エンドノードはVXLANのLocal Biasにより自身のダウンリンクに送信する



EVPN MH&AGの動作イメージ (11/13)

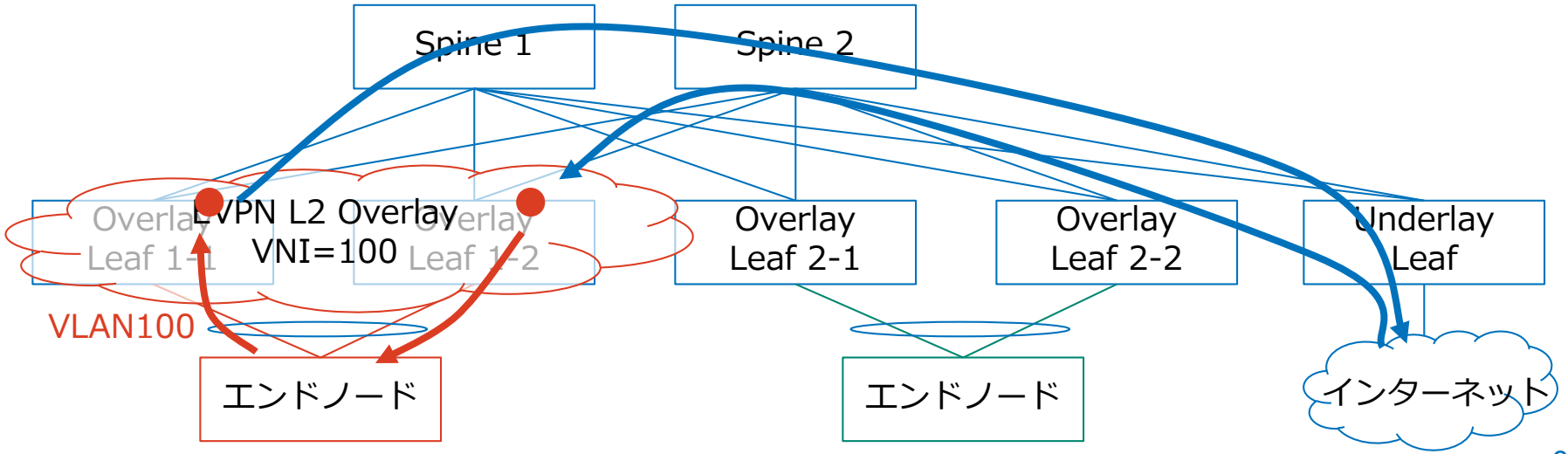
L2セグメントがLeafのペアをまたがない場合はオーバーレイの範囲を
縮小して考えることができる



EVPN MH&AGの動作イメージ (12/13)

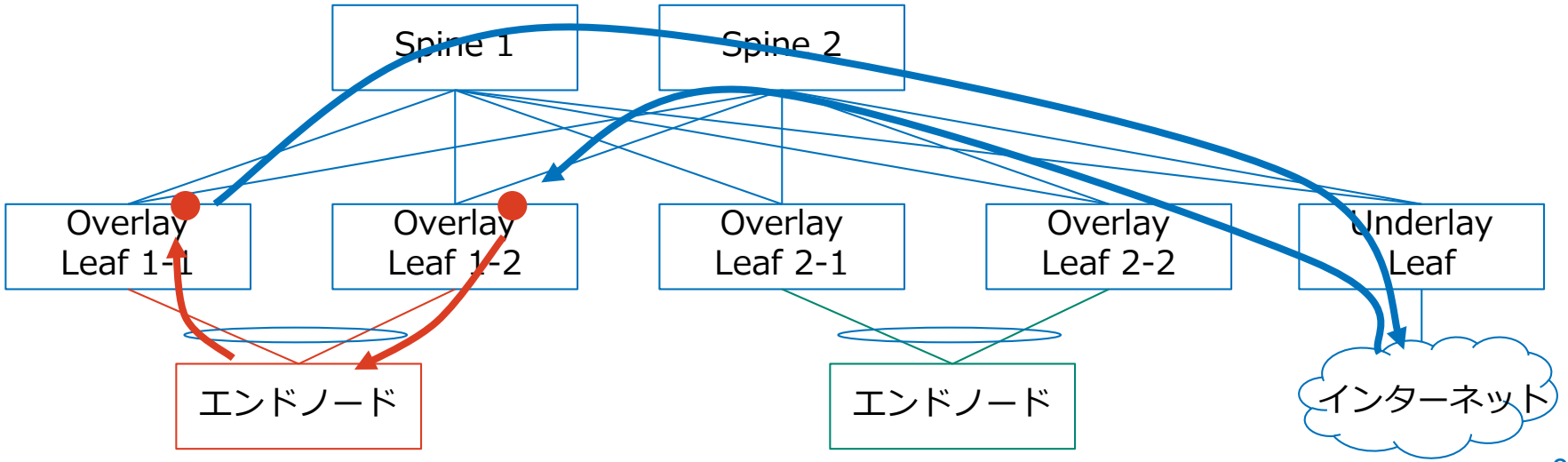
すべてのダウンリンクをMHを組んでいる場合オーバーレイトラフィックは
Spineに行かないので、実質的にオーバーレイはLeaf内部で閉じることになる

➡ 一切VXLANでEncapされない



EVPN MH&AGの動作イメージ (13/13)

これによってLeafで完全に閉じたAct-ActなFirst-Hop Redundancyが実現！！！

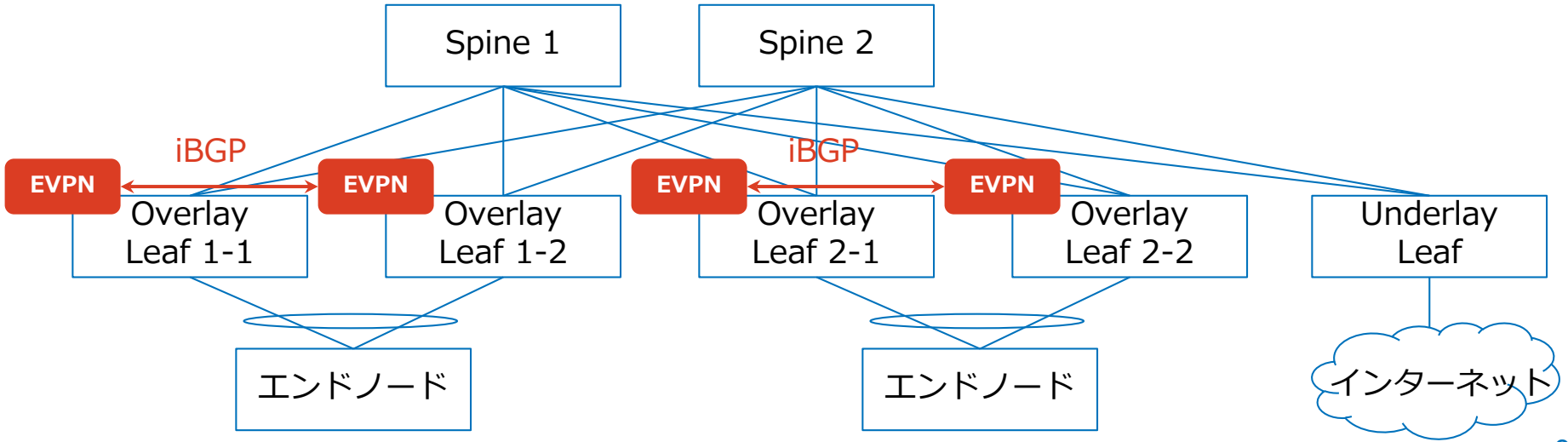


Leaf-Only EVPN Multihoming & Anycast Gatewayの設計上のポイント

さらに改善できる？

他のLeafペアにL2がまたがずSpineにIRBも作らないのであれば

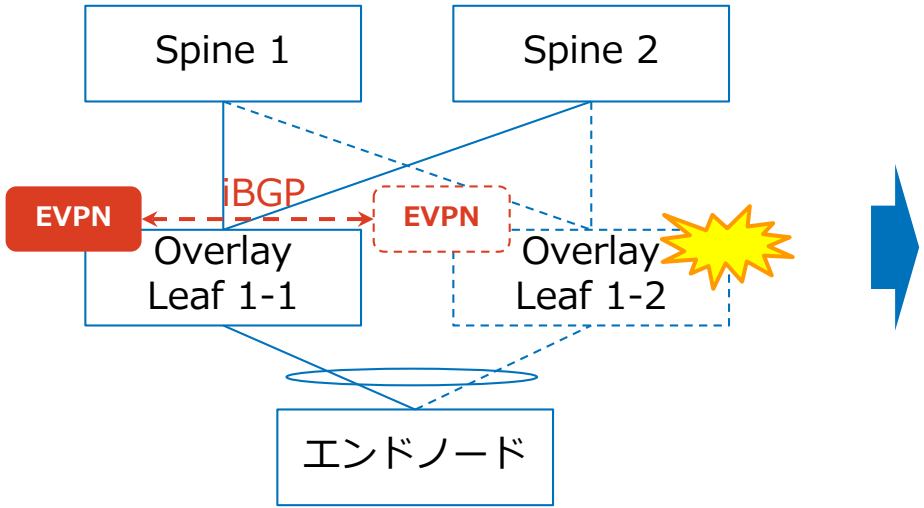
SpineとEVPNの経路を交換する必要もないのでLeafペア間でiBGPを張れば十分？



片方のリーフがダウンすると何が起こるか？

問題

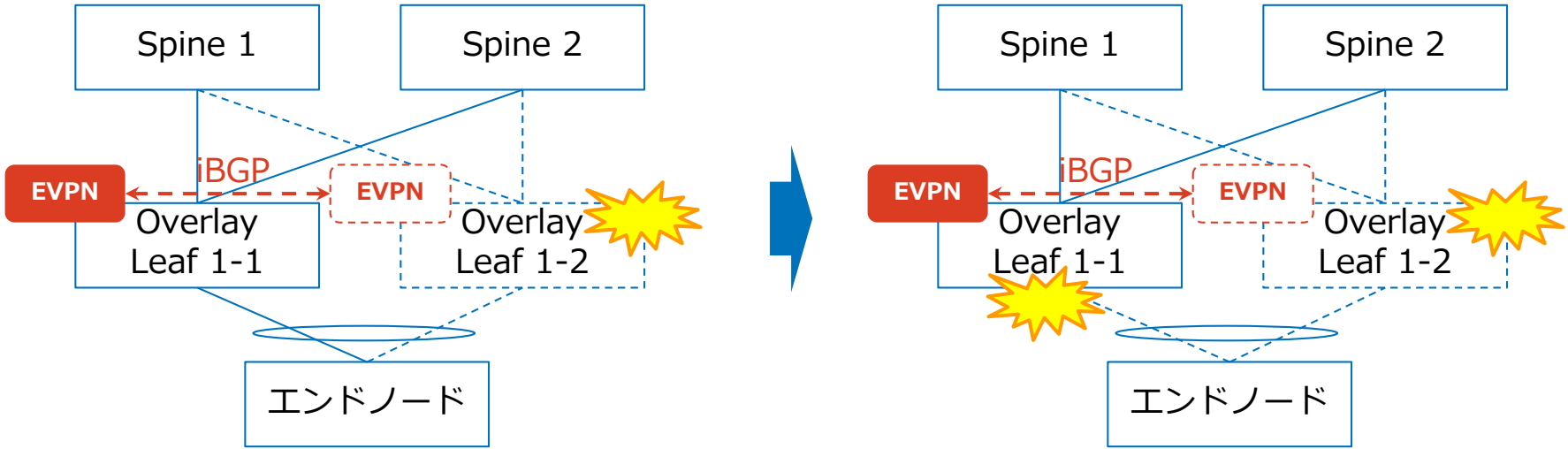
Leaf 1-2がダウンすると何が起きるか？



片方のリーフがダウンすると何が起こるか？

答え

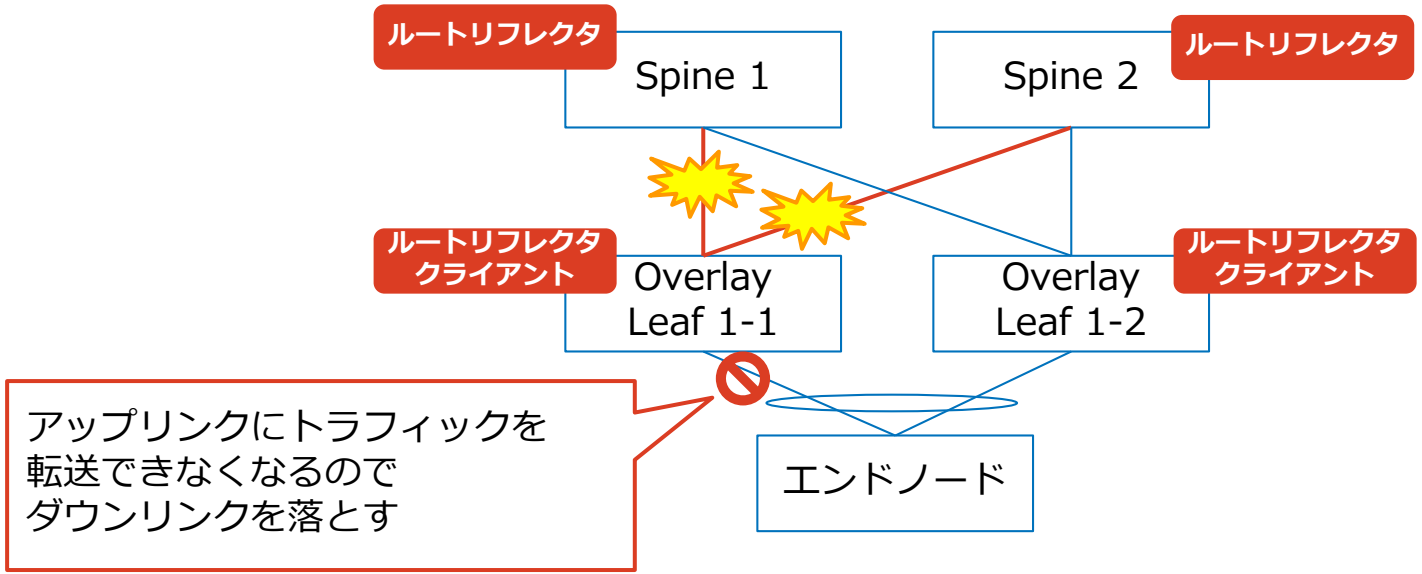
Leaf 1-1がダウンリンクをすべてシャットダウンしてエンドノードが全断する



core-isolation

EVPNですべてのiBGPピアが切断された場合にオーバーレイのポートを
リンクダウンさせてトラフィックを受け取らないようにする機能

※無効化することもできる



２台冗長の難しさ

系に機器が２台しか存在しない場合には原理的に異常検知が意味をなさない

- 自分が正常である担保が出来ない
- スプリットブレインした場合は確認もできない

２台冗長のVRRP/VC/MC-LAGは基本的に全部このパターン

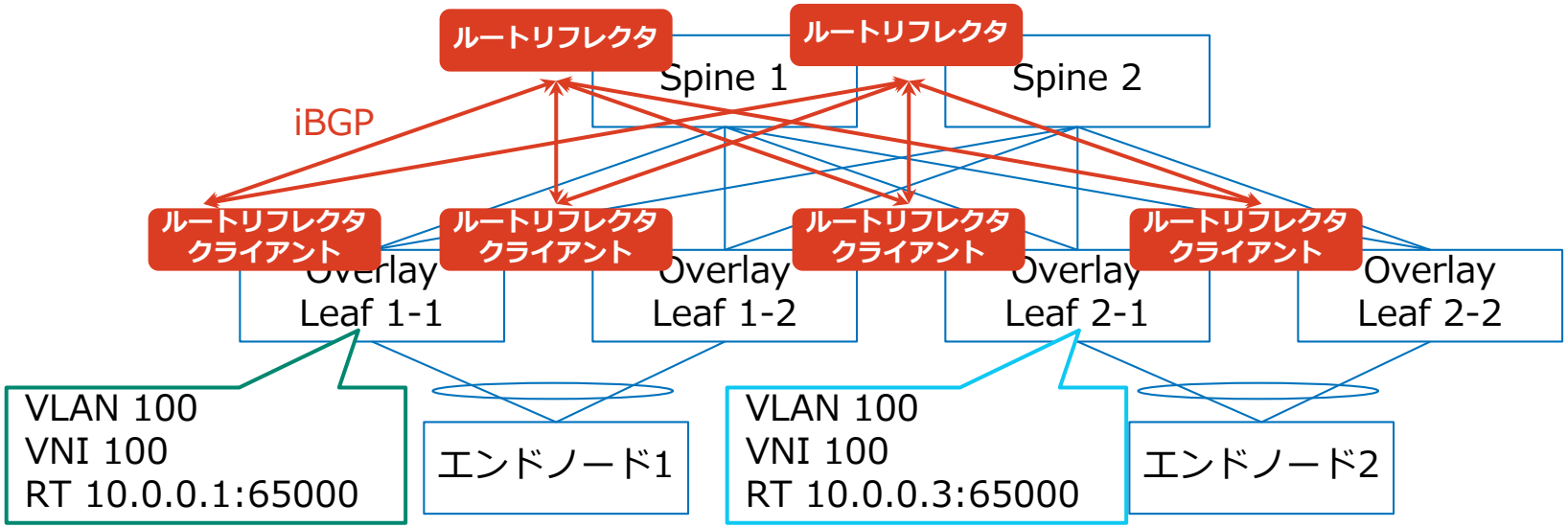
➡ なんらかの外部からの観測や外部とのやりとりが必要

Leaf-Only EVPN MH&AGの最適な構成

直上の**すべての**Spineをルートリフレクタにして、core-isolationは有効にする

Leafペアごとに固有のRTを設定して、他のペアの経路を受信しないようにする

➡ Leaf 1-1/1-2とLeaf 2-1/2-2でそれぞれ同じVNIを使用することができる

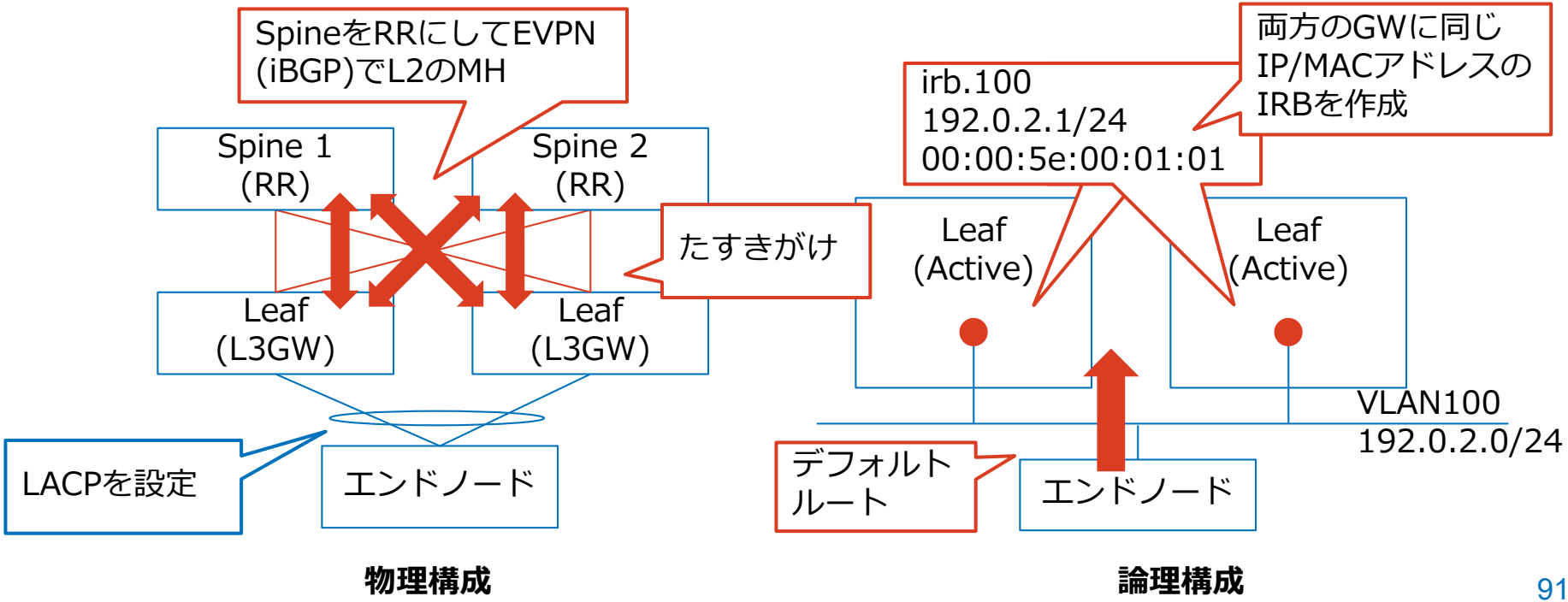


この構成のポイント

- **LeafからほかのLeafへのトラフィックが絶対に通るポイント「すべて」をルートリフレクタにする**
  - ○ Leafから見てiBGPがすべて切れるとき＝アップリンクが全断したとき
  - ○ core-isolationの機能を正しく活かすことができる
  - ○ コントロールプレーンとデータプレーンが同じリンクを通るBGPの良さが活かせている
  - ○ **スプリットブレインが起きない！**
- **Anycast Gatewayのポイントは依然としてLeafにする**
  - ○ Spineも経路を持っているのでAnycast Gatewayのポイントになり得るが、しない
  - ○ 別のLeafペアで同じVNIを使う必要があった・使いたかった
    - ■ サーバ側のVLAN番号はいつも同じ値（アドレスはもちろん違う）
    - ■ VNIをVLAN番号からstaticでmappingしている（実際は同じ値）
    - ■ VNIをIPv4アドレスからmappingすればSpineでAnycast Gatewayすることも可能
    - ■ でも別にSpineでやる必要もない
  - ○ 複雑さはLeafに集中させてSpineは頭空っぽで動作してほしい
    - ■ **Spineがトラブルを起こしたときの罹障範囲は広い**
    - ■ 本当はルートリフレクタもさせたくなかった

EVPN Multihoming & EVPN Anycast Gateway 完成版

EVPN Multihomingの機能を使ってLAGでL2冗長を行い、
EVPN Anycast Gatewayの機能を使ってAct-ActのL3GWを実現する方式



EVPN MH&AGの良いところ 完成版

- コントロールプレーンは標準技術で構成されている
- EVPNなのでL2/L3オーバーレイに対応している
  - 我々の場合、テナント用オーバーレイがEVPN/VXLANなので
    同じ技術でアンダーレイのFirst-Hop Redundancyが実現できる
- コントロールプレーンもデータプレーンも独立していてActive-Active構成
  - バージョンアップしやすい
  - VRRPのように仮想IP/MACアドレスの追加・削除（FIBの更新）が起きない
- ２台冗長構成の原理的欠点を克服している
  - ルートリフレクタを使うことで冗長系外部からの監視ができている
  - core-isolation機能を使うことでスプリットブレインにならない
- BGPを使うのでコントロールプレーンとデータプレーンが同じところを通る
  - Spine-Leaf間はすべてのアップリンクSpineをRRにすることで実現
  - エンドノード-Leaf間はLACPのままなのでBGP手法に比べれば劣る

超推しポイント

EVPN MH&AGの問題点

ライセンス費用

- EVPN/VXLANのためのライセンス費用が余分にかかる場合があり、コスト増になる可能性がある
  - ○ 我々の場合はオーバーレイSDNのためにもともとその分を見積もっているのでとくに障壁なく採用できた

新しい技術だしIRB周りの動作が難しい

- バグが多そう…

どうせトラブルを引くなら
新しい技術にチャレンジして引きたい！

First-Hop Redundancyの実現手法まとめ

EVPN Anycast Gateway が なかま に なった！！！

#	ネットワーク機器	エンドノード
~~1~~	~~VRRP(HSRP) + L2NW~~	~~bonding~~
~~2~~	~~Virtual Chassis~~	~~bonding~~
~~3~~	~~MC-LAG~~	~~bonding (LACP)~~
~~4~~	~~BGP~~	~~BGP~~
5-1	EVPN Multihoming & EVPN Anycast Gateway	bonding (LACP)
5-2	EVPN + MC-LAG & EVPN Anycast Gateway	bonding (LACP)

（参考）EVPN MH&AGの設定例 (1/3)

```
set interfaces xe-0/0/0 description endnode-1_eno1
set interfaces xe-0/0/0 ether-options 802.3ad ae0

set interfaces ae0 description endnode-1_bond0
set interfaces ae0 mtu 9216
set interfaces ae0 esi auto-derive lacp
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options minimum-links 1
set interfaces ae0 aggregated-ether-options local-bias
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp periodic slow
set interfaces ae0 aggregated-ether-options lacp system-priority 100
set interfaces ae0 aggregated-ether-options lacp system-id 02:0a:0:0:01:00
set interfaces ae0 aggregated-ether-options lacp admin-key 1
set interfaces ae0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae0 unit 0 family ethernet-switching vlan members VLAN100

set interfaces irb gratuitous-arp-reply
set interfaces irb unit 100 description endnode-1_bond0.100
set interfaces irb unit 100 family inet address 192.0.2.1/24:
set interfaces irb unit 100 mac 00:00:5e:00:01:01

set interfaces lo0 unit 0 family inet address 10.0.0.1/32
```

ESIで識別されるのでAE番号は何番でもいい

ESIはLACPのsystem-idを使う

LACPのsystem-idはESIに使うので、
{ペア,AE}ごとに固有の値にする。
この例では若番のLeafのLoopbackアドレスと
AE番号を16進数で埋め込んでいる

IRBにゲートウェイになるIPアドレスとMACアドレスを設定する。MACアドレスは他のVLANのIRB重複しても良いのですべてのペア、IRBで同じ値を使っている

Leafのペアごとに固有の値を設定する項目
ペア内の両方のLeafで同じ値を設定する項目
ペア内のそれぞれのLeafで違う値でもいい項目
ペア内のそれぞれのLeafで固有の値を設定する項目

※検証中の設定例なので告知なく改善修正される場合があります

（参考）EVPN MH&AGの設定例 (2/3)

```
set routing-options forwarding-table chained-composite-next-hop ingress evpn
set routing-options router-id 10.0.0.1
set routing-options route-distinguisher-id 10.0.0.1

set protocols bgp group UNDERLAY_AG type internal
set protocols bgp group UNDERLAY_AG local-address 10.102.4.1
set protocols bgp group UNDERLAY_AG family evpn signaling
set protocols bgp group UNDERLAY_AG local-as 65000
set protocols bgp group UNDERLAY_AG neighbor 10.0.0.11
set protocols bgp group UNDERLAY_AG neighbor 10.0.0.12

set protocols evpn encapsulation vxlan
set protocols evpn multicast-mode ingress-replication
set protocols evpn default-gateway do-not-advertise
set protocols evpn extended-vni-list all
set protocols l2-learning global-mac-table-aging-time 900

set switch-options vtep-source-interface lo0.0

set vlans VLAN100 vlan-id 100
set vlans VLAN100 l3-interface irb.100
set vlans VLAN100 vxlan vni 100
```

ルートリフレクタのSpineとEVPNのiBGPを張る設定

EVPN/VXLAN周りの設定あれこれ
LeafをGWにするのでdefault-gatewayはadvertiseしない

VLAN100のIRBを作成してVNI 100に紐付ける

Leafのペアごとに固有の値を設定する項目
ペア内の両方のLeafで同じ値を設定する項目
ペア内のそれぞれのLeafで違う値でもいい項目
ペア内のそれぞれのLeafで固有の値を設定する項目

※検証中の設定例なので告知なく改善修正される場合があります

（参考）EVPN MH&AGの設定例 (3/3)

```
set policy-options policy-statement AG-PAIR-IMPORT term ACCEPT-AG-LEAF-PAIR from community AG-LEAF-PAIR
set policy-options policy-statement AG-PAIR-IMPORT term ACCEPT-AG-LEAF-PAIR then accept

set policy-options policy-statement IMPORT-EVPN term ACCEPT-ESI-IN from community COMMUNITY-ESI-IN
set policy-options policy-statement IMPORT-EVPN term ACCEPT-ESI-IN then accept
set policy-options policy-statement IMPORT-EVPN term REJECT then reject

set policy-options community AG-LEAF-PAIR members target:10.0.0.1:65000
set policy-options community COMMUNITY-ESI-IN members target:10.0.0.1:65000

set protocols evpn vni-options vni 100 vrf-target target:10.0.0.1:65000

set switch-options route-distinguisher 10.0.0.1:65000
set switch-options vrf-import AG-PAIR-IMPORT
set switch-options vrf-import IMPORT-EVPN
set switch-options vrf-target target:10.0.0.1:65000
```

RTはペアごとに設定する

RDは筐体ごとに設定する

LeafでAGする場合はこの辺がテキトーでも動いてしまう

Leafのペアごとに固有の値を設定する項目
ペア内の両方のLeafで同じ値を設定する項目
ペア内のそれぞれのLeafで違う値でもいい項目
ペア内のそれぞれのLeafで固有の値を設定する項目

※検証中の設定例なので告知なく改善修正される場合があります

Before After

今までの接続パターンとFirst-Hop Redundancy

		自社機器							お客様機器
		自社利用			他社リソース収容			他社専有	自社仕様
		サーバ	ストレージ	NW機器	サーバ	ストレージ	NW機器	サーバ	NW機器
P2P L3	LAG	SDNコントローラ							
P2MP L3	LAG	コントローラ	内部用ストレージ	NAT GW LB/FW	VM用HV、コントローラ	VM用ストレージ			
L2オーバーレイ	LAG					テナント用ストレージ	GW		コロケーション接続
	bond mode 1							ベアメタルサーバ	
P2MP L3 管理網	LAG					テナント用ストレージ			
	単一I/F	コントローラ	内部用ストレージ	各NW機器	VM用HV、コントローラ	テナント用ストレージ	各NW機器	ベアメタルサーバ	

VC IRB　VC VxLAN　VRRP + TRILL

これからの接続パターンとFirst-Hop Redundancy

		自社機器							お客様機器
		自社利用			他社リソース収容			他社専有	自社仕様
		サーバ	ストレージ	NW機器	サーバ	ストレージ	NW機器	サーバ	NW機器
P2P L3	LAG	SDNコントローラ							
P2MP L3	LAG	コントローラ	内部用ストレージ	NAT GW LB/FW	VM用HV、コントローラ	VM用ストレージ			
L2オーバーレイ	LAG					テナント用ストレージ	GW		コロケーション接続
	bond mode 1							ベアメタルサーバ	
P2MP L3管理網	LAG					テナント用ストレージ			
	単一I/F	コントローラ	内部用ストレージ	各NW機器	VM用HV、コントローラ	テナント用ストレージ	各NW機器	ベアメタルサーバ	

VRRPが一部残る…

EVPN MH&AG

EVPN VxLAN (+MH)

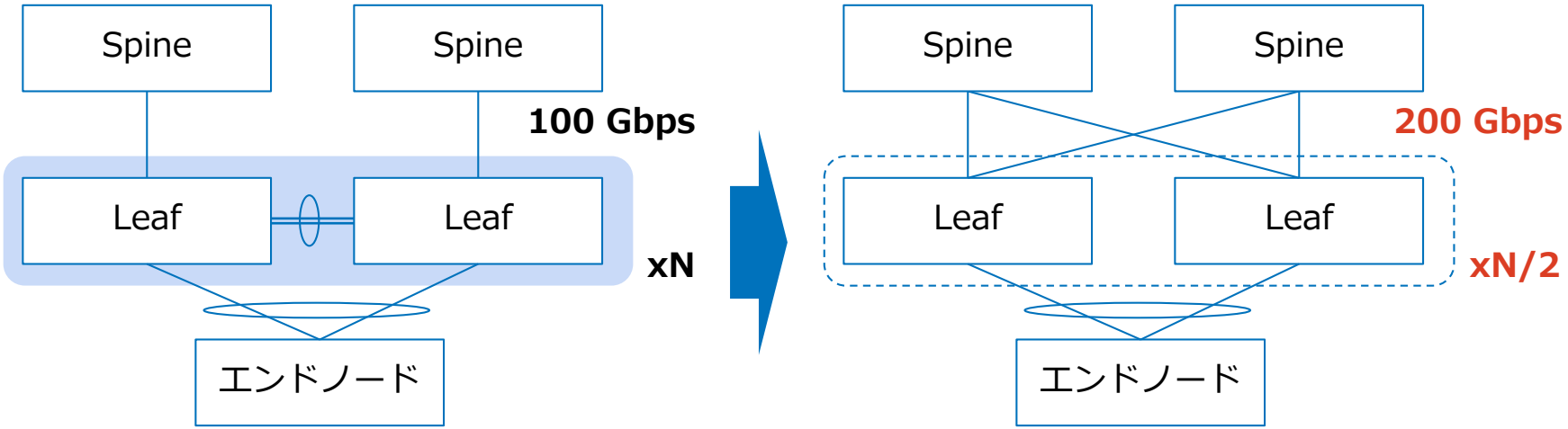
EVPN AG + vPC

VRRP + vPC + EVPN

Virtual Chassisをやめた影響

たすきがけにする必要があるのでSpineあたりのLeaf収容数が半分になる

➡ データセンター全体のスケールが半分になる（帯域は２倍になる）



まとめ

First-Hop Redundancyの実現手法のざっくり比較

項目	VRRP	VC	MC-LAG	BGP	EVPN MH&AG
仕様	標準技術	プロプライエタリ	プロプライエタリ	標準技術	標準技術
NW機器間の結合度合い	疎結合	超密結合	密結合	結合なし	疎結合
データプレーン	Act-Stb	Act-Act	Act-Act	Act-Act	Act-Act
L2収容	必須	可能	可能	不可能	可能
BUMトラフィック	マルチキャスト	なし	なし	なし	なし
IPアドレス	3個	1個	3個	2サブネット	1個
ハートビート	半分inband	out-of-band	out-of-band	inband	inband
エンドノードに必要な技術	bonding	bonding	LACP	BGP	LCAP
バックボーンのスケーラビリティ	影響なし	影響なし	影響なし	加算される	影響なし
最大罹障範囲	収容機器	収容機器	収容機器	リンク	Spine配下
スプリットブレイン	起きる	起きる	起きるけど救える	起きない	起きない

まとめ&これから

- VRRPとVirtual Chassisを捨てて次なる技術を求めてさまよったらEVPN Anycast Gatewayに出会いました
- このEVPN Anycast Gatewayを商用リージョンに導入します
- 次のJANOGあたりで実際どうなの？を検証結果や運用実績を交えて紹介できると思います

議論ネタ

- ストレージの対応どうしてます？
- ストレージもBGPしゃべってくれませんか？
- MC-LAGのVRRP over IRBって元気に動いてます？
- BGPを使う場合にスケールで問題になったケースってあります？
- もうEVPN Anycast Gatewayで痛い目にあった方いらっしゃいます？
- その他痛い目に合った経験談

議論の記録

- L社Sさん
  - 質問：ストレージなどのBGPをしゃべれないアプライアンスについてはTop of Rackに向かってStaticを書いて、BGPをToRで終端すればいいのでは？
  - 回答：そのStaticがデフォルトゲートウェイであり、そこをどう冗長するかが問題。ストレージの機種やサーバかどうかで技術を分けたくなかった。
- I社Mさん
  - コメント：MC-LAGでVARPではなくVARPを使ってAnycast Gatewayを使っている。VRRPを使いたくなくてこの方式にしているが元気に動いている。エンドノードのゲートウェイの冗長はうまくいくけど、下りトラフィックだとARPテーブルをゲートウェイ同士でうまく同期できなくて、障害があって寄るとARP学習が大量に行われて通信断が発生する。
  - 質問：EVPN Anycast Gatewayだとどうか？
  - 回答：今の所救えているが、大量のエンドノードを収容した状態ではまだ検証していないので、わからない。我々のユースケースでは40台で10VLAN以下なので、切り替え自体はかなり高速に出来ていてロスがないことが確認できている。
  - Syncが発生しない仕組みなので、片方に情報が有れば動くのが良い点

議論の記録

- A社Tさん
  - コメント：EVPN Anycast Gatewayはかなり良いと思う。どこまでのスケールを考えるかにもよるが、マルチベンダでも使えるし、あいだのリンクがいらないので帯域も有効活用できるし、JANONの人もやりたがる三重化や四重化もできる。EVPN AGがあるのでこれから構築する人はこれを選ぶと思う。１，２年前まで流行っていたのはARPやNDPのエントリーをサブスクライバールートのようにredistributeするのがクラウドプロバイダで流行っていた
  - 回答：おっしゃるとおりデータプレーンは各自の実装だがコントロールプレーンは標準化されているので違うベンダーで組むことも理屈上は可能だ（実際はだいたい動かない）。今回のユースケースでは２冗長を取り扱っているが、３冗長・４冗長も可能ですね。

議論の記録

- N社Oさん
  - Q. Anycastのケースに抵触しないのですが、L2オーバーレイで提供されていて、エンドノード側で冗長をVRRPなどで実施している場合や、VMのvMotionなどが行われる場合に制約を設けているか？質問の意図としてはRFC7432の規定でMAC MobilityでL2ループを防止する仕組みがあるのでエンドノード側でイーサネットセグメントをまたぐ事があると問題になる。
  - A. 今回はAnycast Gatewayの部分はアンダーレイ部分でしか使ってしかおらず、MAC Mobilityがカウントアップするような収容設計にしていない
  - 追記：L2オーバーレイ部分ではVRRPを使っている箇所がありここではVRRPのMasterフラップが起きた時にMAC Mobilityの問題が発生するため、実装と設計に応じた対処が必要になる。

議論の記録

- K社Tさん
  - Q. ストレージの冗長化でSoftware Defined Storageなどを使えばLinuxの上で動くので、そういった技術も使いやすくなるのかなと思うのですが、iSCSIなどを使わないといけないのでしょうか？
  - A. ストレージ側の性能や価格の要件が優先されてしまい、ネットワーク側の要件が二の次三の次になってしまう。Cephの運用も今の所の耐性で難しい問題もある。
- L社Iさん
  - コメント：ストレージについてはオブジェクトやボリュームにはCephを使って運用している。他のコンテンツプロバイダもCephやストレージサービスを独自でソフトウェアで書いている。そういった世界ではサーバがBGPを喋ることがある。ネットワーク帯域を使うところは少し違う対応もしている箇所がある。エンタープライズな要件があるところは古いL2が残っている。ソフトウェアの世界でも冗長クラスタを組む時にL2が必要なものがあって、そういった箇所にはL2が残っている。

議論の記録

- N社Hさん
  - ○ Q. Active-Activeになっているかと思いますが、通り道の把握は問題になっていないか？
  - ○ A. エンドノードとLeafの間で問題になる。Leaf1から打ったpingがLAGのハッシュでLeaf2に返されてしまう。LAGの問題でそこまで今までと変わらないのであまり気にはしていない
  - ○ コメント：ESXiサーバなどでも使える知見だと思うのでありがたい