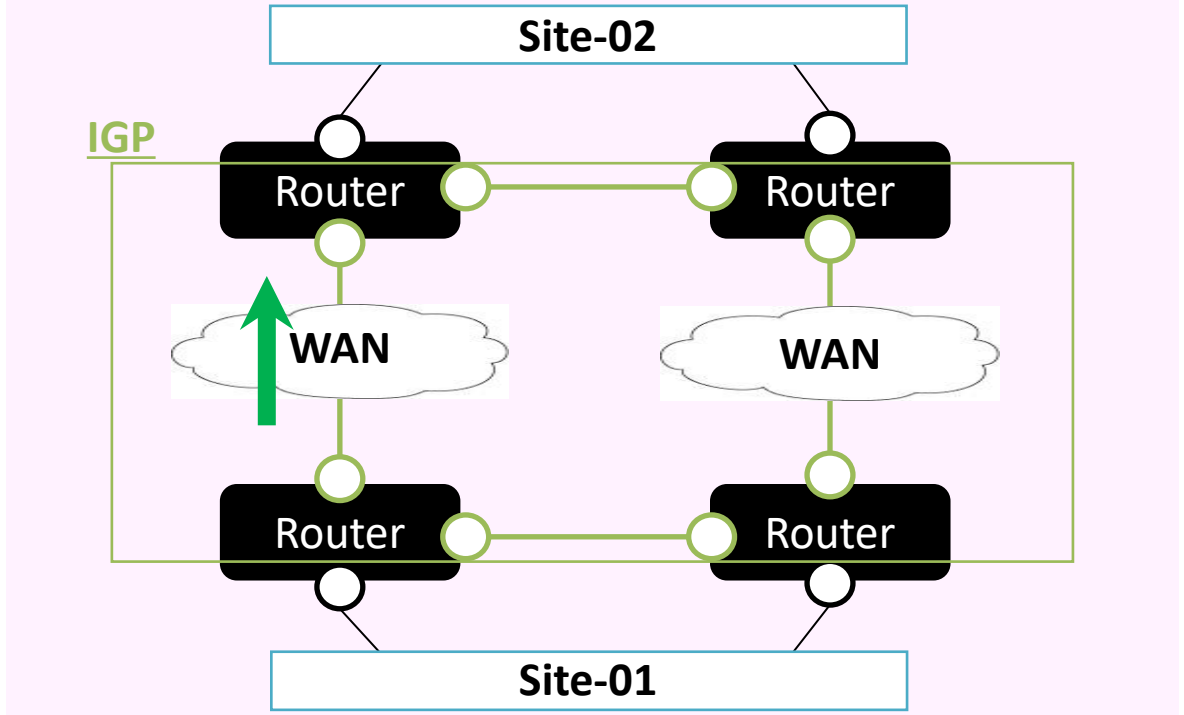
Segment Routing over IPv4ってどうなの？

ネットワンシステムズ株式会社
平河内 竜樹

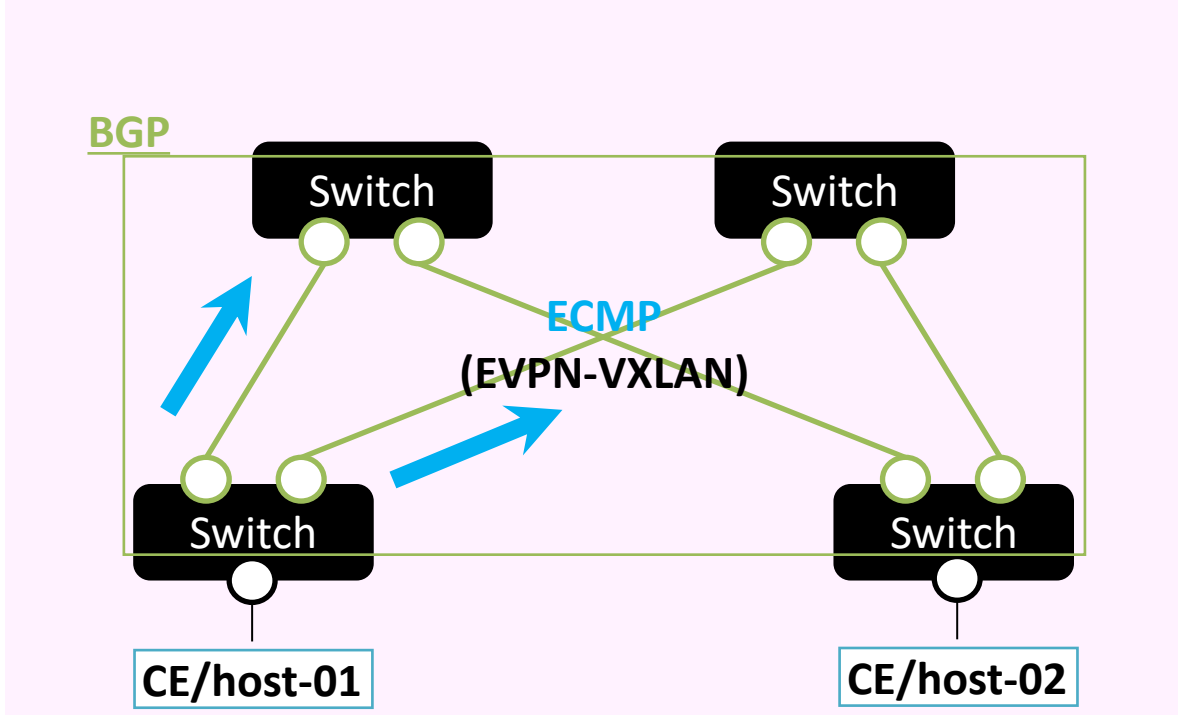
思うこと、、

例）Enterprise WAN



IPv4環境でバックアップパスが持てない問題

例）Data Center Fabric



IPv4環境で伝送路が特定できない問題

MPLSを使えば対応できるけれど…

IPv4でトポロジ非依存のFRRが使えたら…

IPv4でネットワークベースのTEが使えたら…

Segment RoutingとIPv4データプレーン

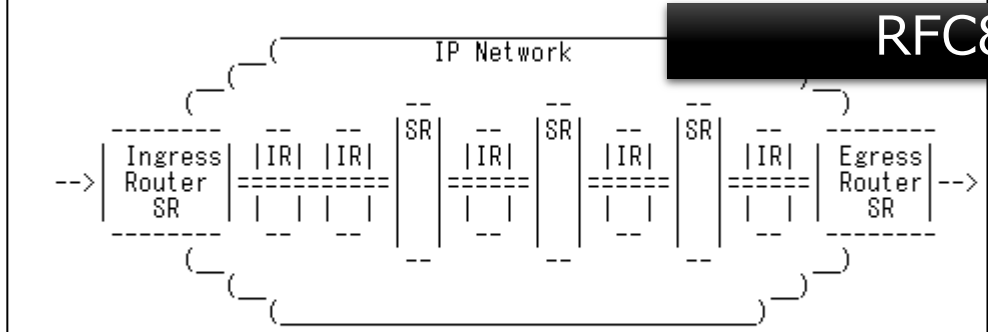
そういえば、昨今ホットなSegment Routingって、IPv4 transportに対応しているの？

RFC8402

The SR architecture can be instantiated on various data planes. This document introduces two data-plane instantiations of SR: SR over MPLS (SR-MPLS) and SR over IPv6 (SRv6).

RFC8663

* Tunneling MPLS over IP provides a technology that enables Segment Routing (SR) in an IPv4 and/or IPv6 network where the routers do not support SRv6 capabilities [IPv6-SRH] and where MPLS forwarding is not an option. This is shown in Figure 2.



Key:
IR : IP-only Router
SR : SR-MPLS-capable Router
== : SR-MPLS-over-UDP Tunnel

Figure 2: SR-MPLS Enabled within an IP Network

RFC-Editor: RFC8663

PROPOSED STANDARD

X. Xu
Alibaba, Inc
S. Bryant
Futurewei Technologies
A. Farrel
Old Dog Consulting
S. Hassan
Cisco
W. Henderickx
Nokia
Z. Li
Huawei

December 2019

MPLS Segment Routing over IP

✓ IPv4がNGであるわけではない✨

✓ いくつかの仕様が存在しRFCになったものもある✨

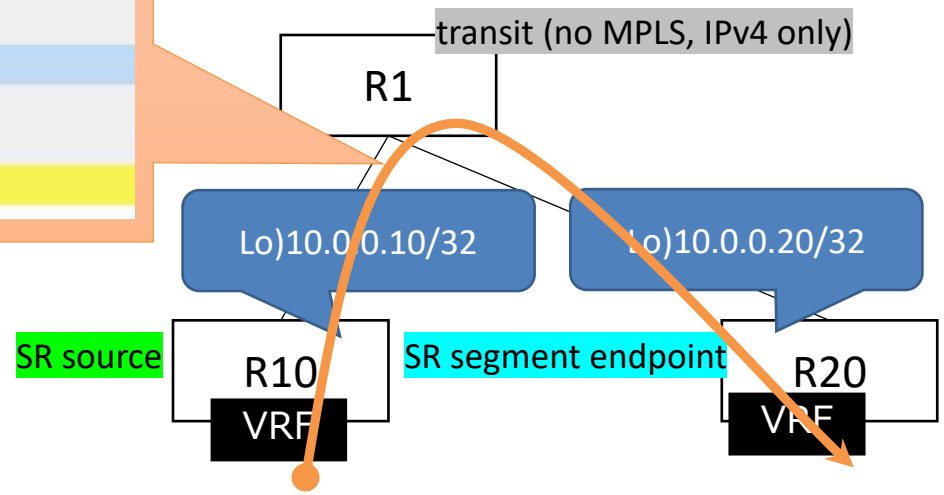
やってみた＃１

■SR-MPLS over IP（RFC 8663）

□サポートされている製品で動かしてみました

No.	Source	Destination	Protocol	Length	Info
1	10.68.101.1	10.68.102.1	ICMP	138	Echo (ping) request id=0x717b, seq=1,

```
> Frame 1: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits)
> Ethernet II, Src: 00:00:00_00:00:10 (00:00:00:00:00:10), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 110
> Internet Protocol Version 4, Src: 10.0.0.10, Dst: 10.0.0.20
> User Datagram Protocol, Src Port: 56622, Dst Port: 6635
> MultiProtocol Label Switching Header, Label: 16, Exp: 0, S: 1, TTL: 64
> Internet Protocol Version 4, Src: 10.68.101.1, Dst: 10.68.102.1
> Internet Control Message Protocol
```



✓ **IPv4 transport上でMPLS VPNが疎通した✨**

プロトコルへの要求

■SR-MPLS over IP（RFC 8663）の場合…

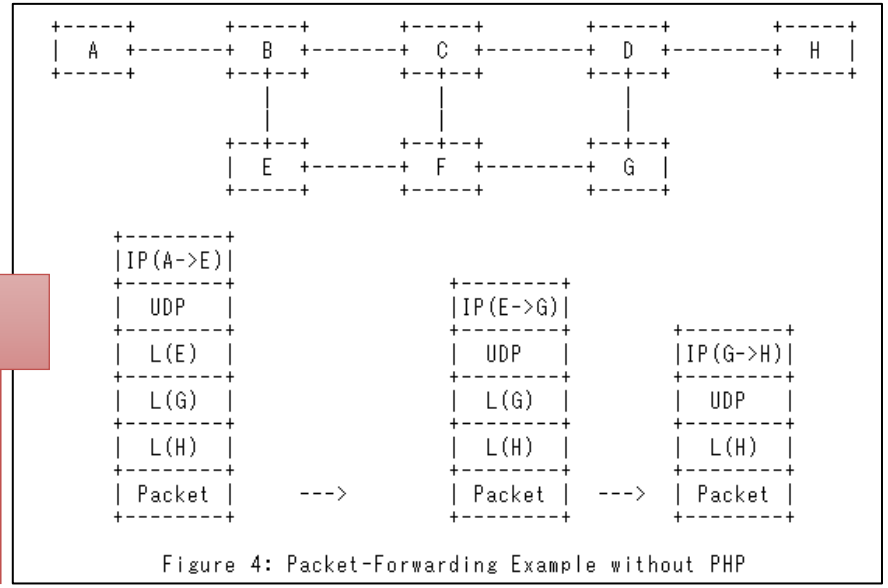


Figure 4: Packet-Forwarding Example without PHP

TEするときは？

- [PUSH] パケットにSR-MPLS SIDが積まれる
- [NEXT] 各SRノードがSIDを基にIPv4ヘッダを書き換える

➢ **マッピング情報とそれを交換するコントロールプレーンが必要**
- ✓ SR-MPLSに対応していれば、SID⇔IPv4アドレスの情報は交換可能であり、差分は少量
- ✓ IP転送のCONTINUE処理を行うノードは、同じプロトコルで中継できれば良い

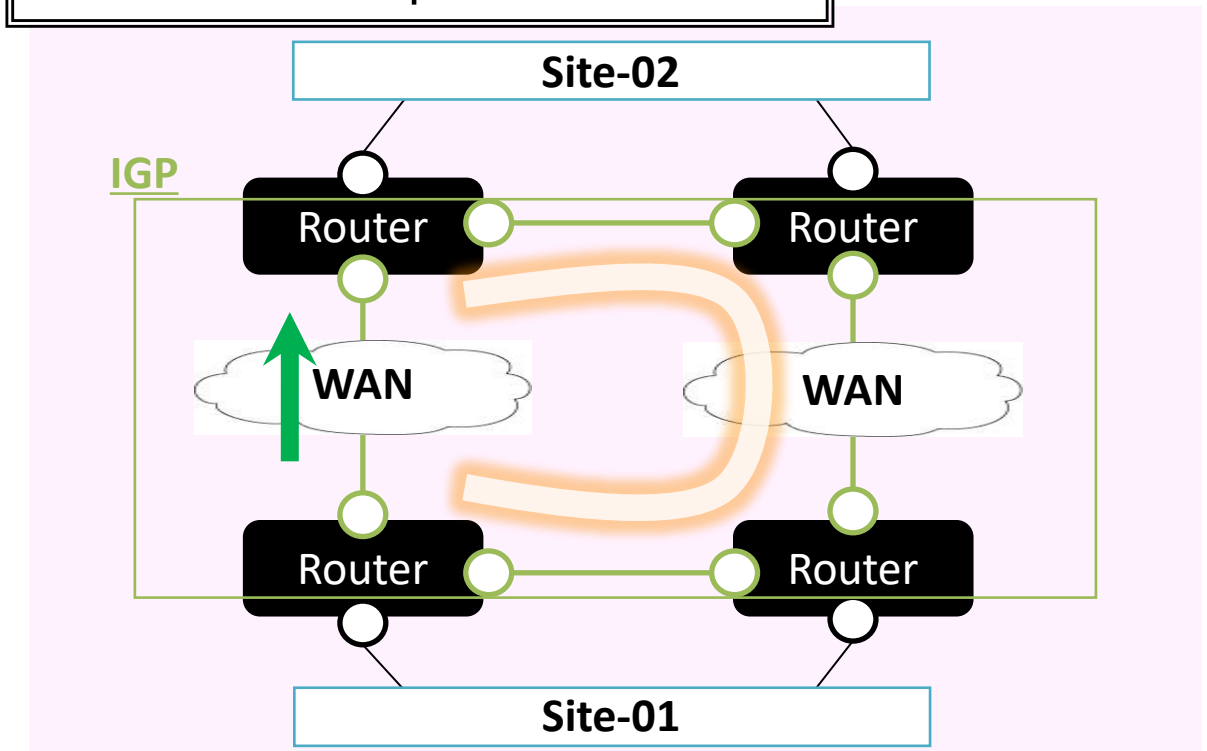
✚

➢ **VPN構成の場合、VPNラベルが発生する**
- ✓ MPLS VPN資産を継承して利用可能な点はアドバンテージ

✓ MPLSの広域展開に適している✨
✓ 環境により、 MPLSの利用やコントロールプレーンへの要求が、導入ハードルに作用し得る😥

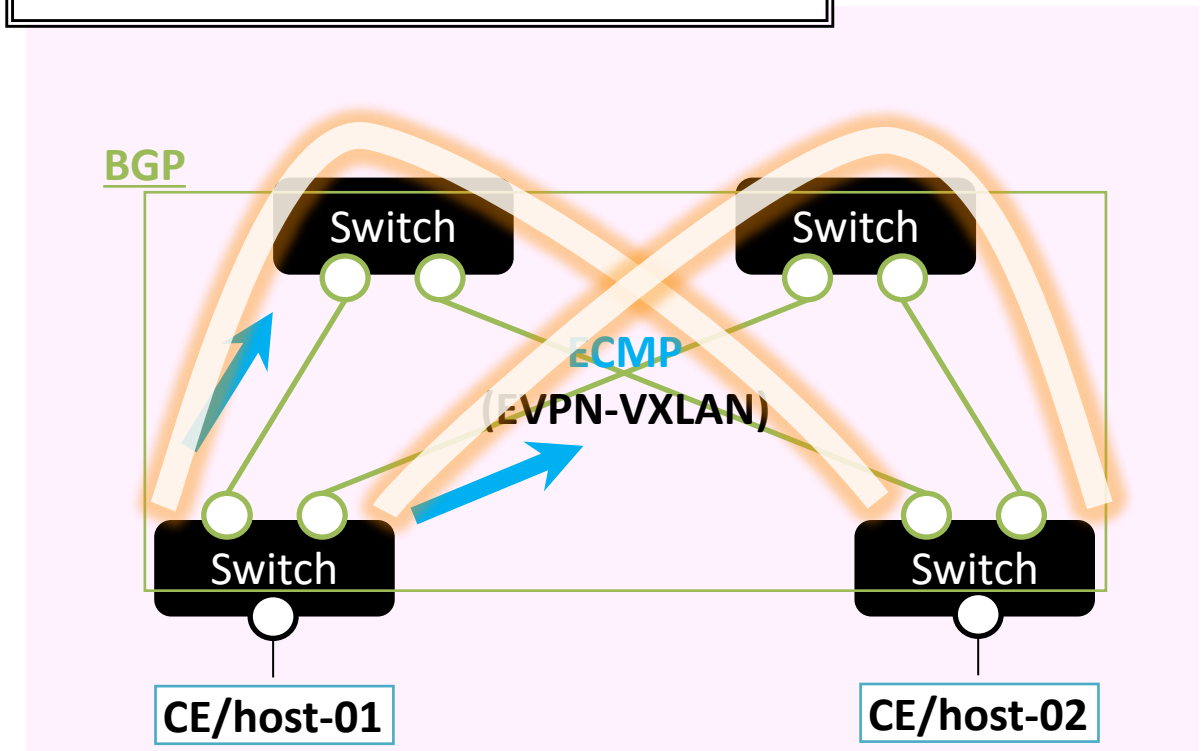
こんな技術があったらニーズに合う？

例）Enterprise WAN



SIDが積まれたIPv4 Tunnelを形成
↓
バックアップパスを実現

例）Data Center Fabric



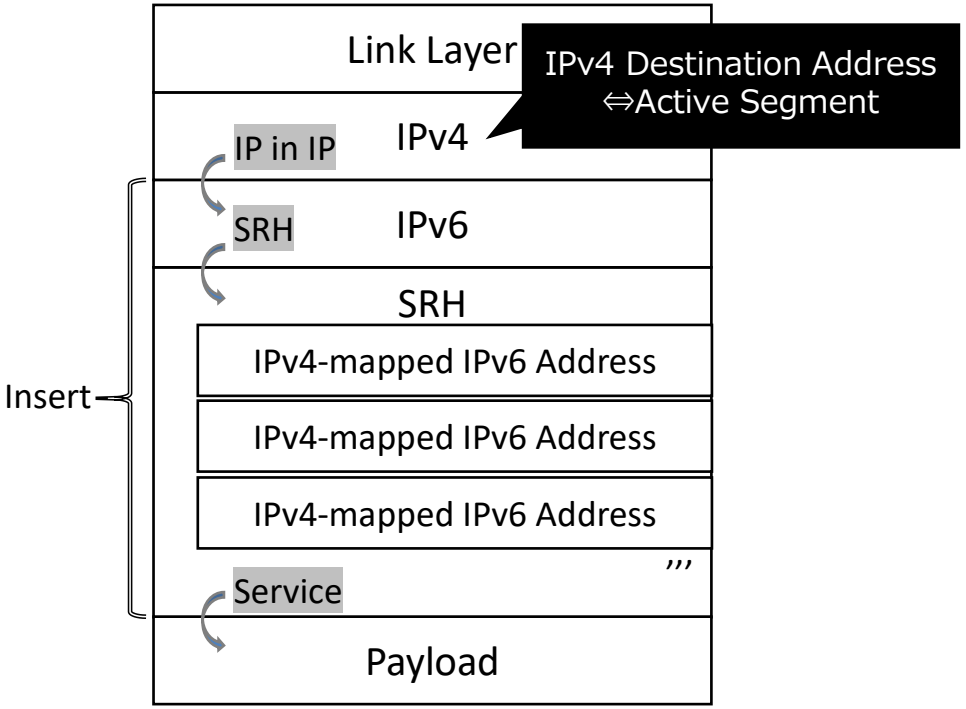
生成されるIPv4 TunnelにSIDを挿入
↓
サービスマッピングで任意のTEパスを選択

MPLS及びコントロールプレーン拡張無しで・IPv4-only環境でも使えるSegment Routing🤔

こんな技術があったらニーズに合う？

【対応例】

- 割り当てられたIPアドレスを**Transport SID**として扱う
  - ✓ 他のVPN機構と組み合わせて各種サービスに対応
- SRv6の**SRHヘッダ**に**32-bit SID**情報を埋め込む
  - ✓ IPv4射影空間を定義し、値で処理を分岐



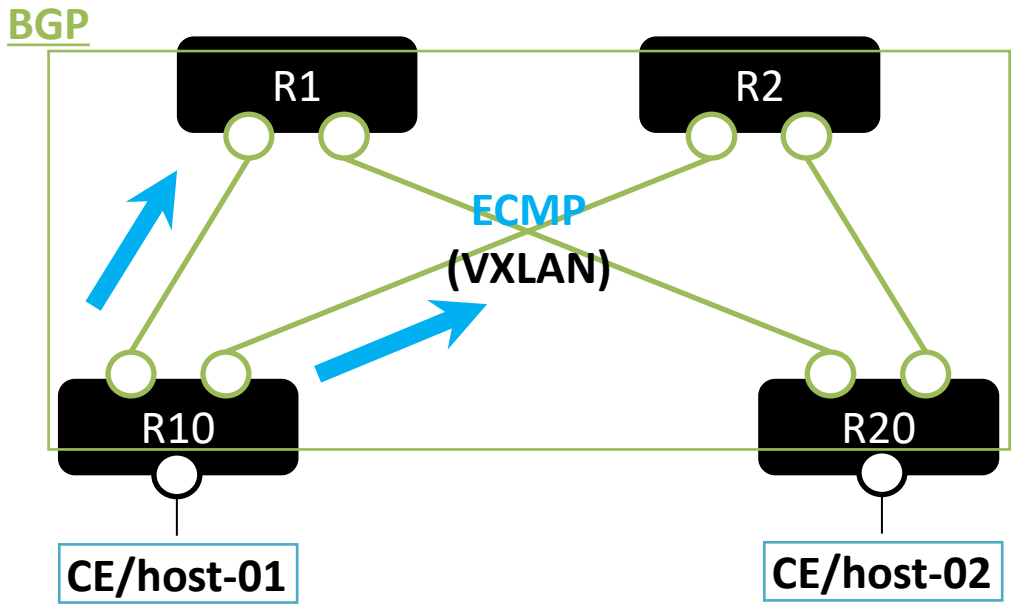
Segment Routing over IPv4を実現する一手段（SRv6 over IP）

- **[MPLSおよび専用テーブル不要]** Active Segmentの中継・更新後処理はIPv4 FIBの参照で完結
- **[コントロールプレーン拡張不要]** 必要なコントロールプレーンは通常のIPv4 Routing
- **[ヘッダ変更不要]** 標準のSRv6 SIDとの共存

以降の検討対象

適用前

■対象とするVXLAN Fabricの例



トンネル終端アドレスは10.0.0.{Node番号}

R1

Prefix	NH (backup)	Protocol
10.0.0.10/32	R10	BGP
10.0.0.20/32	R20	BGP

R2

Prefix	NH (backup)	Protocol
10.0.0.10/32	R10	BGP
10.0.0.20/32	R20	BGP

R10

Prefix	NH (backup)	Protocol
10.0.0.20/32	R1 R2	BGP

R20

Prefix	NH (backup)	Protocol
10.0.0.10/32	R1 R2	BGP

✓ フローベースで負荷分散されるスケールアウトな構成
✓ パケットの情報から伝送路を特定する要件との相性に課題

【課題に関しても言及】
JANOG39 :: IDCフロンティア"データ集積地構想/Data Centric Cloud"を支えるネットワークのアーキテクチャ
https://www.janog.gr.jp/meeting/janog39/program/dcc.html

Stacked Tunnel的なことをやってみる

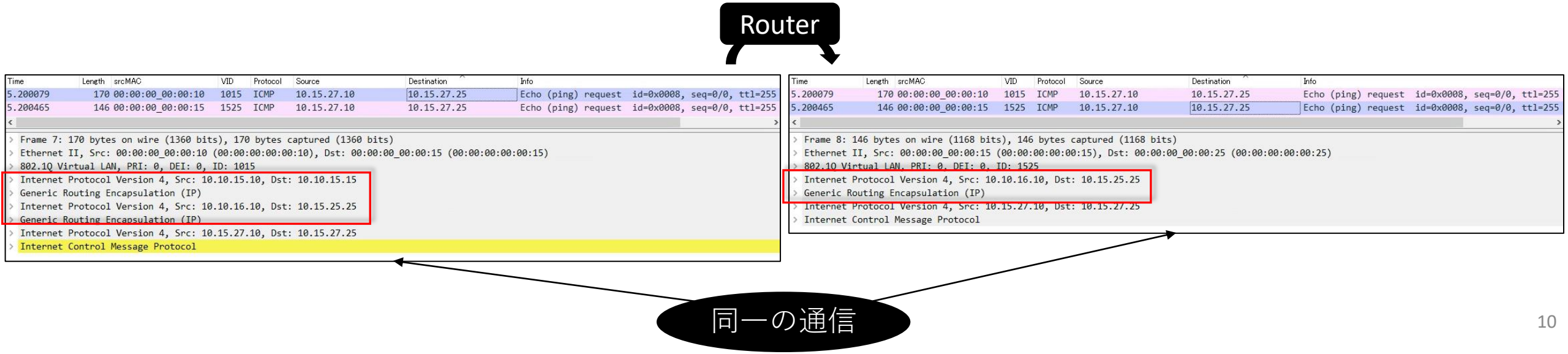
■GREトンネルで32-bit SIDのスタックを模擬

□完全な再現は期待せずに、Stacked Tunnelの効果を観察

➢今回はVXLANに追加してさらに一段積む

□多段の際は、自分宛のGREを解くと自分宛でないGREが出てくる構成を検討

➢下図は検討過程の記録

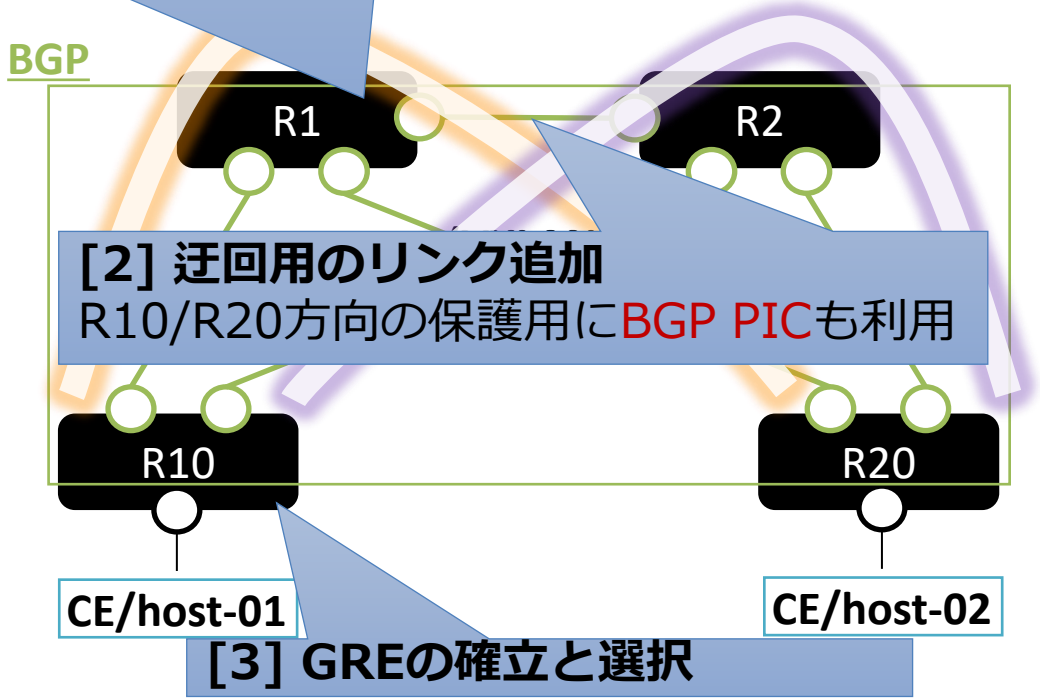


やってみた＃２

適用後

■“中継の選択”と”Anycast的なSIDでの保護”を想定した、GREの併用

[1] GREエンドポイント用のアドレス2種類
保護用にR1/R2両方にPrefix長を変えて割り当て
-Primary path用　　(今回：10.0.**1**.12)
-Secondary path用　(今回：10.0.**2**.12)



R1

Prefix	NH (backup)	Protocol
10.0.0.10/32	R10 (R2)	BGP
10.0.0.20/32	R20 (R2)	BGP

```
R1#show ip route repair-paths
~(snip)~
B        10.0.0.10/32 [20/0] via 10.1.10.10, 08:51:21
                      [RPR][20/0] via 10.1.2.2, 08:51:21
B        10.0.0.20/32 [20/0] via 10.1.20.20, 08:51:21
                      [RPR][20/0] via 10.1.2.2, 08:51:21
~(snip)~
```

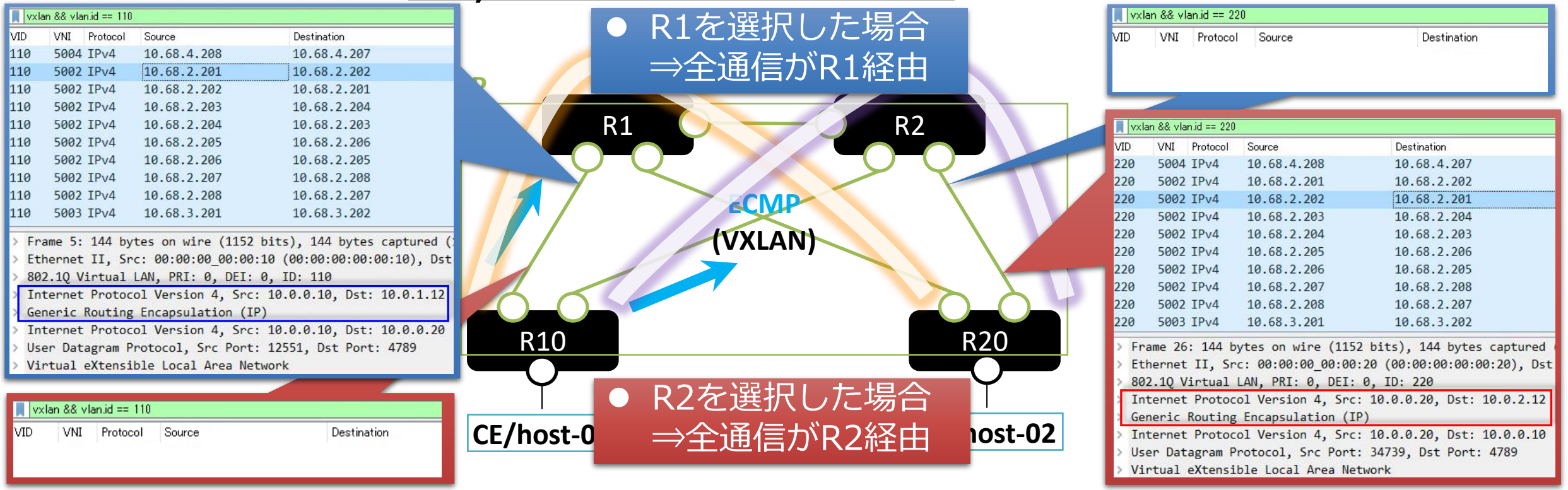
R10

Prefix	NH (backup)	Protocol
10.0.0.20/32	R1 R2	BGP
10.0.1.12/29	R1	BGP
10.0.1.12/28	(R2)	BGP
10.0.2.12/29	R2	BGP
10.0.2.12/28	(R1)	BGP

```
R10#show ip route
~(snip)~
B        10.0.0.20/32 [20/10] via 10.2.10.2, 08:45:42
                      [20/10] via 10.1.10.1, 08:45:42
B        10.0.1.0/28 [20/10] via 10.1.10.1, 00:00:58
B        10.0.1.8/29 [20/10] via 10.1.10.1, 00:01:28
B        10.0.2.0/28 [20/10] via 10.2.10.2, 00:01:28
B        10.0.2.8/29 [20/10] via 10.2.10.2, 00:00:58
~(snip)~
```

やってみた＃２

■“中継の選択”と”Anycast的なSIDでの保護”を想定した、GREの併用



✓ スタックされたIPv4アドレスで、TE/FRRできた✨ … アドレス選択の効用

✓ リンク障害時は迂回路利用の継続が発生した🤔 … 本アプローチの制約事項

✓ static routeでのサービスマッピングとなった😱 … 活用には適切な手段が必要

Flex-Algo的なことをやってみる

■前項では、複数のノードに同一のアドレスを割り当てた

- 上乗せするIP(32-bit SID)にAnycastを利用したStacked Tunnel的なこと

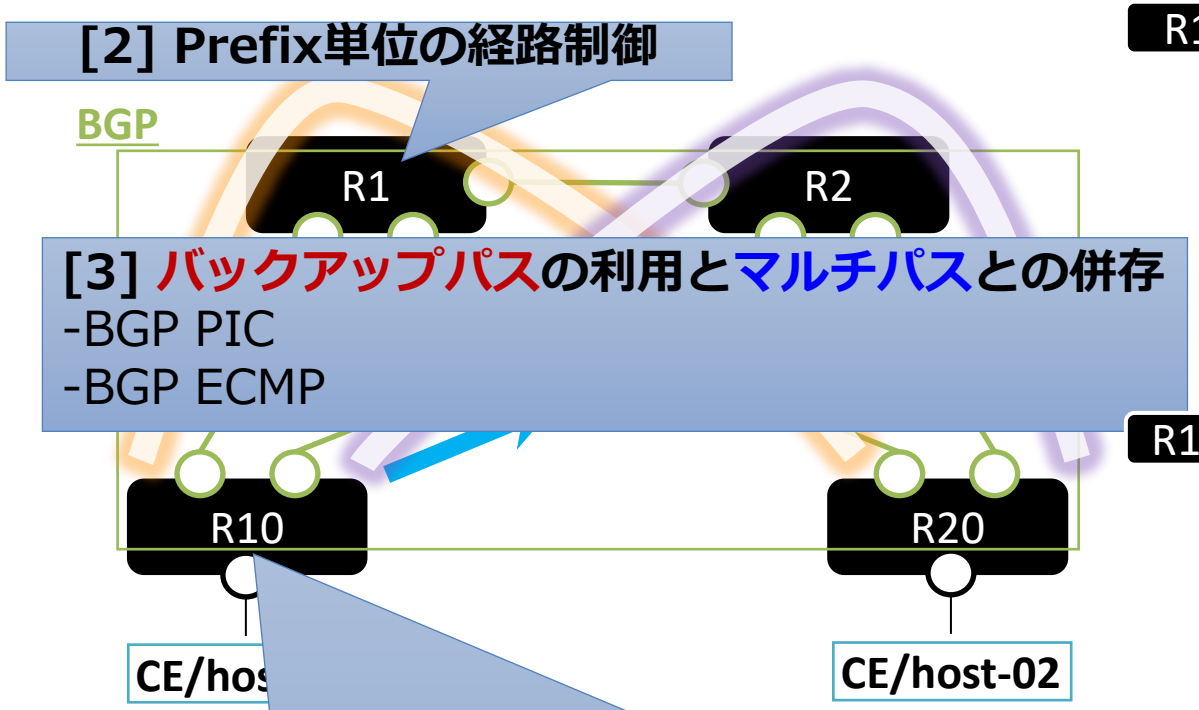
■次は、単一のEdgeノードに対する複数のアドレス割り当てにトライ

- 同一の対象に複数のIP(32-bit SID)が割り当てられる側面に着目
  - ここでの意趣として、IGPは関係ありません…

やってみた＃３

適用後

■“Flex-Algo的なSID割り当て”を想定した、VXLANのアドレス選択



R1

Prefix	NH (backup)	Proto
10.0.0.10/32	R10 (R2)	BGP
10.0.0.20/32	R20 (R2)	BGP
10.0.1.10/32	R10 (R2)	BGP
10.0.1.20/32	R20 (R2)	BGP
10.0.2.10/32	R10 (R2)	BGP
10.0.2.10/32	R20 (R2)	BGP

R10

Prefix	NH (backup)	Proto
10.0.0.20/32	R1 R2	BGP
10.0.1.20/32	R1 (R2)	BGP
10.0.2.20/32	R2 (R1)	BGP

```
R1#show ip route repair-paths
~(snip)~
B        10.0.0.10/32 [20/0] via 10.1.10.10, 00:10:48
                      [RPR][20/0] via 10.1.2.2, 00:10:48
B        10.0.0.20/32 [20/0] via 10.1.20.20, 00:10:48
                      [RPR][20/0] via 10.1.2.2, 00:10:48
B        10.0.1.10/32 [20/0] via 10.1.10.10, 00:10:48
                      [RPR][20/0] via 10.1.2.2, 00:10:48
B        10.0.1.20/32 [20/0] via 10.1.20.20, 00:10:48
                      [RPR][20/0] via 10.1.2.2, 00:10:48
B        10.0.2.10/32 [20/0] via 10.1.10.10, 00:10:48
                      [RPR][20/0] via 10.1.2.2, 00:10:48
B        10.0.2.20/32 [20/0] via 10.1.20.20, 00:10:48
                      [RPR][20/0] via 10.1.2.2, 00:10:48
~(snip)~
```

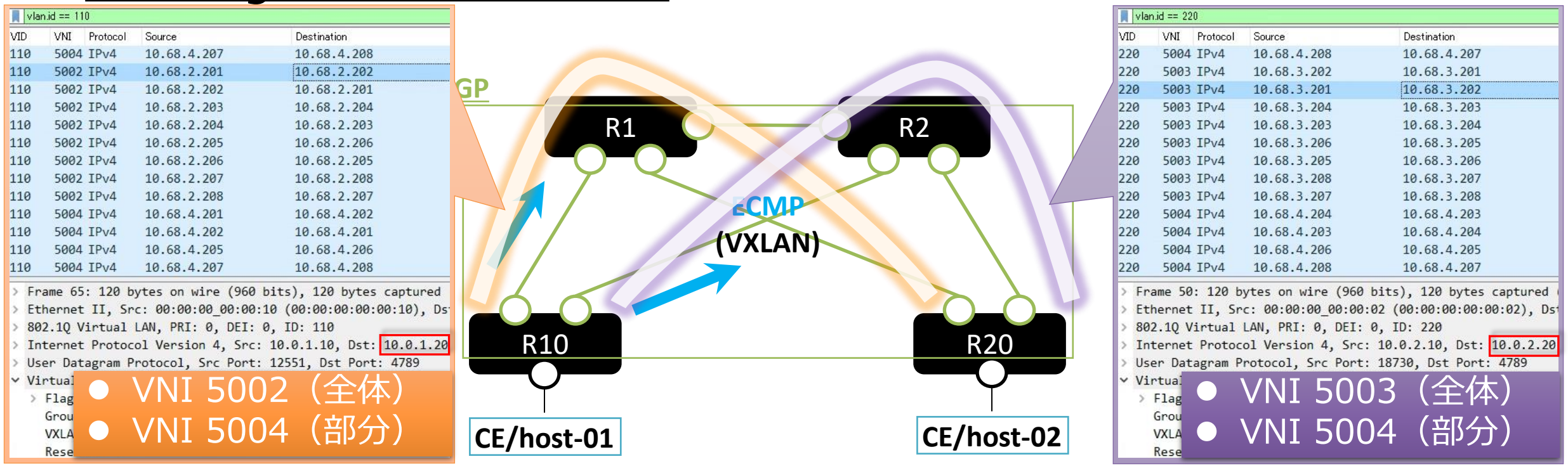
```
R10#show ip route repair-paths
~(snip)~
C        10.0.0.10/32 is directly connected, Loopback0
B        10.0.0.20/32 [20/10] via 10.2.10.2, 00:10:48
                      [20/10] via 10.1.10.1, 00:10:48
C        10.0.1.10/32 is directly connected, Loopback1
B        10.0.1.20/32 [20/10] via 10.1.10.1, 00:10:48
                      [RPR][20/100] via 10.2.10.2, 00:10:48
C        10.0.2.10/32 is directly connected, Loopback2
B        10.0.2.20/32 [20/10] via 10.2.10.2, 00:10:48
                      [RPR][20/100] via 10.1.10.1, 00:10:48
~(snip)~
```

[1] VXLANエンドポイント用のアドレス3種類

-ECMP用　　(今回：10.0.**0**.{NODE-ID}　⇒ VNI 5004で利用)
-Primary path用　(今回：10.0.**1**.{NODE-ID}　⇒ VNI 5002で利用)
-Secondary path用　(今回：10.0.**2**.{NODE-ID}　⇒ VNI 5003で利用)

やってみた＃３

■“Flex-Algo的なSID割り当て”を想定した、VXLANのアドレス選択



✓ スタック無しで、TE/FRRできた✨
✓ 同時に、ECMPも提供できた✨

✓ サービスマッピングは各ingressで個別に指定🤔

✓ トランスポートにIGPを使う際のprefix単位の制御🤔

… アドレス選択の効用

… BGP連携への期待

… IP Flex-Algorithmへの期待

まとめ・ご意見頂けると有難いこと

■まとめ（SR over IPv4）

□仕様も実装もある

1

- ➢例）SR-MPLS over IP
  - ⇔ SR-MPLS SIDとIPのマッピング

□IPv4 Routingで実現するには？

2

- ➢例）SRHから分岐、IPv4転送のみ実行
  - ⇔ SID値で判断 (SRv6 over IP)

3

- ➢もしくは、積まない範囲での対応
  - ✓関連技術を活用

■ご意見頂けると有難いこと

□意義はありそう？

- ➢動態があれば試してみたい？
- ➢使える場面はありそう？

□他方面で影響ありそう？

- ➢例）『IPv6準備を促進するのでは？』
- ➢例）『IPv6利用を阻害するのでは？』

Appendix

【検討】このTEパス、どのように利用する？

	IGPに参加させる	BGPでマッピングする	個別に指示する
概要	➢ IGPの最短パスを基に決定	➢ BGPのパス属性を基に決定	➢ 設定で直接指定
利点	◘ IGP経路の追加に透過的 ◘ IGPドメインが延伸可	◘ BGP経路の追加に透過的 ◘ BGPパス属性を外部で編集可 ◘ BGP生成トンネルと連動可	◘ 実装次第で任意のマッピング
留意点	◘ サービスはL3に限定 ◘ TEパス間の使い分けが困難 ◘ recursive routingに注意	◘ サービスはBGP広報が前提	◘ 都度設定が必要

やってみた＃２・３

【検討】このTEパス、どのように保護する？

	Stacked Tunnel的アプローチの場合	Flex-Algo的アプローチの場合
概要	➢ 中継に固有のIPv4アドレス(SID)を割り当て ✓ sourceは追加挿入するSIDを選択 ⇒ Anycast Addressでバックアップパスを保持	➢ 出口に複数のIPv4アドレス(SID)を割り当て ✓ sourceは宛先となるSIDを選択 ⇒ IPのFRR機能でバックアップパスを保持
利点	◘ コントロールプレーンの拡張無しで保護可能 （Longest Matchを利用しFIBに同時挿入）	◘ 一段のSIDでTE可能（SRH/IPv6省略可能） ◘ コンバージェンスによるパスの最適化
留意点	◘ 縮退運転中の迂回路がトポロジ制約に繋がる	◘ 必要となるTE/FRR制御が制約事項になり得る （BGP、IGP拡張、リンクレイヤを刻む等）

やってみた＃ 2

やってみた＃ 3

【検討】このTEパス、どのように分散する？

■ECMPエントロピー情報の格納先について

- □IPv4転送で考慮が必要な箇所
- □前提として、ECMP-aware TEに対応する？
  - ➢out of scopeとすることも選択肢

やってみた＃2

	ハッシュ値をIPv4ヘッダに格納	ハッシュ値を追加UDPに格納	SRH/IPv6ヘッダを積まない
概要	➢ 送信元IPアドレスに格納	➢ 送信元ポート番号に格納	➢ VXLAN等のヘッダ構造を維持
利点	□ ヘッダ構造が維持	□ 送信元アドレスが維持	□ ヘッダ・アドレスが維持
留意点	□ Local Bias等の妨げになる	□ 別のヘッダ構造が必要	□ TEは一段SIDの範囲内

やってみた＃3