HEXA
NETWORKS

Routing and Internet **Experts.**

Semana da **Capacitação**
MPLS além do Convencional:
L3VPN e 6VPE na prática!

André **Dias**

- **Fundador** e **CEO** da **Hexa Networks e NetConfig**;
- **Mais de 14** anos atuando na área de **redes**;
- **Mais de 22 anos** atuando na área de **tecnologia** (iniciando aos 8 anos de idade);
- **Aos 16 anos** iniciou em um provedor de VoIP (Telefonia sobre IP);
- **Aos 17 ingressou** em um **ISP** (Provedor de Acesso a Internet);
- Certificado **MTCNA, MTCRE, MTCINE, MTCIPv6E, JNCIA-JUNOS, JNCIA-DC, JNCIS-SP, NRS-I** ;
- Autor de artigos para o **"Brasil Peering Fórum"**;
- Fanático assíduo por **IPv6**;
- Já fiz parte da maior orquestra do planeta;
- Sou chato e teimoso;
- Colaborador para a maior obra coletiva da humanidade, a Internet.

Wallace **Andrade**

- **CTO na Hexa Networks;**
- Experiência com roteamento, MPLS, engenharia de tráfego, BGP e administração de sistemas autônomos;
- Cursou Engenharia Química e Análise de Sistemas, a partir da qual teve contato com tecnologia e redes;
- Certificações Huawei HCIP, Juniper JNCIA e Nokia NRS II;
- Palestrante do GTER
- Desenvolvedor Full Stack freelancer nas horas vagas;
- Contra baixista, baterista, tecladista, e por aí vai;
- Apreciador de um bom churrasco;

O Que A Hexa Networks faz?

HEXA NETWORKS

Fundada em 2017, a Hexa Networks nasceu com o propósito de apoiar provedores de acesso e operadoras de trânsito na busca por estabilidade, excelência, escalabilidade e estrutura em suas redes. Com foco no crescimento saudável das operações, nos especializamos em tecnologias como protocolos de roteamento, MPLS, engenharia de tráfego e serviços avançados de rede.

Hoje, contamos com uma **equipe altamente capacitada**, formada por especialistas com vasta experiência em ambientes críticos. Atuamos em **redes de todos os portes**, desde infraestruturas simples até as mais complexas, em **clientes espalhados por 4 continentes**. Nosso time está preparado para lidar com os desafios técnicos mais exigentes, sempre com foco em performance e continuidade de serviço.

Entendemos que provedor de internet não pode parar. Oferecemos **suporte técnico 24/7**, garantindo disponibilidade e resposta ágil para ambientes que não podem parar. Independentemente do tamanho da rede ou da localização do cliente, entregamos soluções personalizadas com qualidade e comprometimento.

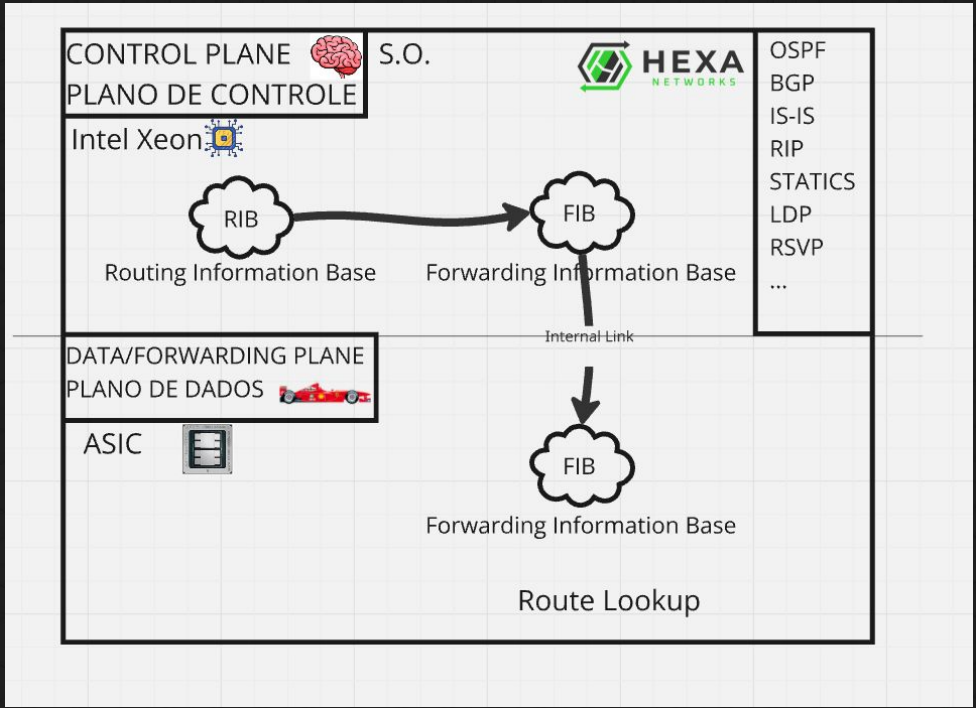
O Que é O NetConfig?

O **NetConfig** é uma plataforma de automação e gestão de redes que **reduz drasticamente o custo operacional dos provedores**, eliminando tarefas manuais, acelerando processos e minimizando erros humanos. Ideal para ISPs e consultorias, é uma solução robusta e escalável para ambientes de missão crítica.

Entre os principais recursos, estão o **acesso remoto via SSH Web com auditoria**, **automação de RPKI e IRR**, **gestão de backups** e **configuração centralizada de equipamentos** por meio de uma interface gráfica prática e poderosa.

Ao unir automação, controle e rastreabilidade em uma única plataforma, o NetConfig entrega eficiência operacional, segurança e agilidade, permitindo que as equipes técnicas foquem no que realmente importa: a evolução e a estabilidade da rede.

Planos de Controle e Dados



Um pouquinho sobre Hardware - Plano de Controle

Para o plano de controle é comum vermos processadores "convencionais" x86 sendo utilizados, como Intel Xeon.

Em alguns casos podemos ver alguns modelos de arquiteturas legadas como PowerPC. Um exemplo desse caso são os Juniper MX5/80 e MX104.

O plano de controle precisa de processamento, e não de especialização. Sendo assim, a tecnologia efetivamente acaba ficando no plano de dados

JUNIPER NETWORKS

Um pouquinho sobre Hardware - Plano de Dados

O plano de dados acaba sendo o responsável pelo encaminhamento de pacotes. Sendo assim, é extremamente importante ser um chip especializado (ASIC - Application Specific Integrated Circuit) para essa finalidade.

Em 1998 a Juniper foi pioneira na separação do plano de controle do plano de dados a nível de hardware com o Juniper M40 com o chipset I-CHIP, equipamento que comportou 40 milhões de pacotes por segundo. Um número impressionante para a época!

Mais informações no livro MX Series , da própria Juniper.

ARTIFACT DETAILS

Title
Juniper M40 Router

Catalog Number
102742462

Type
Physical object

Esse Juniper M40 foi um presente da Juniper para o Computer History Museum, o que demonstra o grau de importância desse equipamento para a tecnologia que temos atualmente

Description
CHM collects actively in the networking area and this router is networking company Juniper Networks' first product, released in 1998. Using a proprietary application specific integrated circuit, the M40 was at least 100 times faster than any other commercial router at the time. Juniper built on the success of the M40 and had net 2013 revenue of $4.7 billion.

Date
1998

https://www.computerhistory.org/collections/catalog/102742462

Manufacturer
Juniper Networks

Identifying Numbers

Um pouquinho sobre Hardware - Plano de Dados

O tempo foi passando, e novos chips foram sendo lançados. Posteriormente a própria Juniper veio com a linha MX usando o chipset Trio, que permanece sendo o carro forte da fabricante até os dias de hoje, cujo qual atualmente está na sua versão 6 com o Juniper MX304

O abaixo é um MX304 e possui capacidade de até 4.8T de encaminhamento.

Um pouquinho sobre Hardware - Plano de Dados

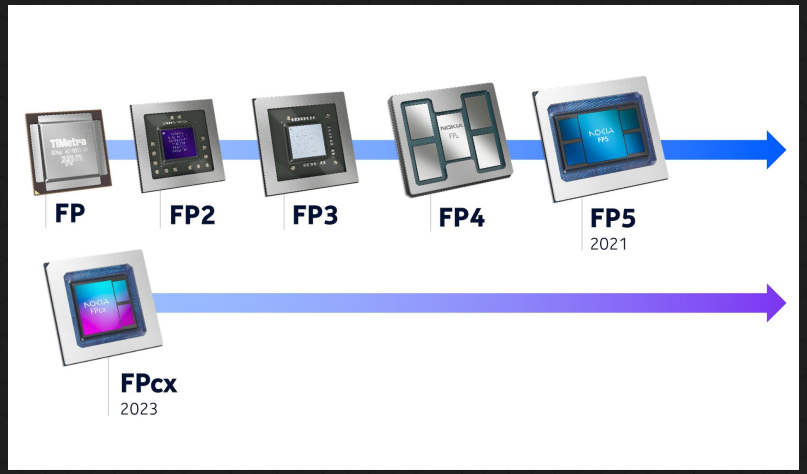
Para a linha PTX vemos outro chipset da Juniper sendo utilizado, sendo o Express. Esse apesar de possuir menos funcionalidades, possui uma capacidade muito maior de encaminhamento.

O equipamento abaixo é o PTX10001, e possui 9.6T de capacidade de encaminhamento.

NOKIA

Um pouquinho sobre Hardware - Plano de Dados

A Nokia também possui fabricação propria de chipset para alguns modelos de equipamento. Podemos citar aqui a linha 7750 SR1-X que utiliza o chipset FP5.



Um pouquinho sobre Hardware - Plano de Dados

A utilização de chipset da Huawei é uma historia um tanto quanto interessante por existir conflitos geo-politicos.

Até o s6720 utilizavam chips Broadcom, onde a partir do s6730 passaram a usar Chips proprios, sendo o SOLAR.

A documentação sobre ele é deveras escassa, sendo citado poucas vezes em documentação dos s12700.

Um pouquinho sobre Hardware - Plano de Dados

Ok professor, mas e outras fabricantes como Datacom? O que utilizam de chip? E a resposta é BROADCOM!

A Broadcom é uma fabricante de chips, onde diversos vendors utilizam chips deles. Podemos citar alguns como Juniper (QFX, EX, ACX), Nokia (7250), Cisco, Huawei, Arista, Datacom, Parks, PADTEC e por ai vai!

A Broadcom tem diversos chips conhecidíssimos como:

-Tomahawk;
-Jericho;
-Trident;
-Qunran;
-Strata;
-Metrolite.

Resumindo...

O que aprenderam aqui?

A separação do plano de dados e plano de controle foi um salto tecnológico importante.

Cada chip tem suas especificações e determina onde cada equipamento entrará no backbone. Um chip pode ter capacidade altíssima de encaminhamento, porém carecer de suporte à features.

Entender isso é de extrema valia!

É por esse motivo que algumas funções só apareceram nos s6720-HI, s5720-HI e nos s6730!

O antecessor do BGP, o EGP

Antes do BGP ser o protocolo padrão para comunicação entre sistemas autônomos, existiu um protocolo chamado de EGP (Exterior Gateway Protocol) - RFC827. O EGP desempenhou um papel crucial no desenvolvimento inicial da Internet, facilitando a troca de informações de roteamento entre sistemas autônomos (AS) distintos.

O EGP foi eventualmente substituído pelo BGP (RFC1772) devido a suas limitações, como a incapacidade de suportar múltiplos caminhos, atributos de caminho e controle baseado em políticas para o tráfego de rede. A transição para o BGP marcou uma evolução significativa na infraestrutura de roteamento da Internet, atendendo às necessidades de uma rede global cada vez mais complexa e interconectada

O que é BGP?

BGP é o protocolo utilizado como alicerce da internet, costurando as redes para comunicar os provedores.

O BGP utiliza a TCP/179 para comunicação, e faz trocas de NLRI (Network Layer Reachability Information) assim como o OSPF faz trocas de LSA.

Atualmente o BGP está na sua versão 4, e possui suporte a extensões tornando-o MP-BGP (Multi-Protocol Border Gateway Protocol) em conjunto com as famílias de endereço (AFI), e famílias de endereços subsequentes (SAFI).

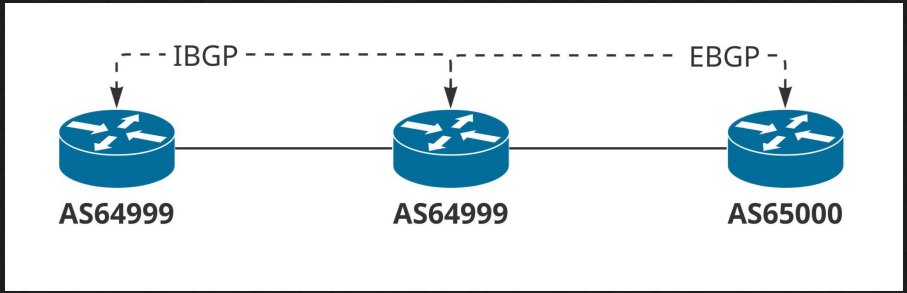
Um exemplo prático é o suporte a IPv6, onde utilizamos o AFI 2 SAFI 1 (IPv6 Unicast).

iBGP e eBGP

O BGP se baseia na utilização de ASN para que suas sessões sejam configuradas, a partir do qual se determina se a relação é iBGP ou eBGP.

O eBGP (External Border Gateway Protocol) é utilizado para comunicação entre dois sistemas autônomos diferentes, como por exemplo, na contratação de um link.

O iBGP (Internal Border Gateway Protocol) é utilizado para comunicação dentro do mesmo sistema autônomo, onde o ASN remoto e o ASN local são os mesmos.



iBGP e eBGP

Funcionamento básico do eBGP:

- Realiza troca de gateway por padrão;
- O que aprende via eBGP, consegue ensinar tanto via eBGP, quanto via iBGP.

O iBGP sofre algumas alterações conceituais, e merece atenção:

- Por padrão não realiza troca de gateway, onde pode passar a fazê-lo utilizando a opção “Next-Hop Self” (pode variar);
- O que aprende via iBGP ensina apenas via eBGP => split horizon => caso necessário comunicar diversos equipamentos via iBGP é necessário Full Mesh ou Route Reflector...

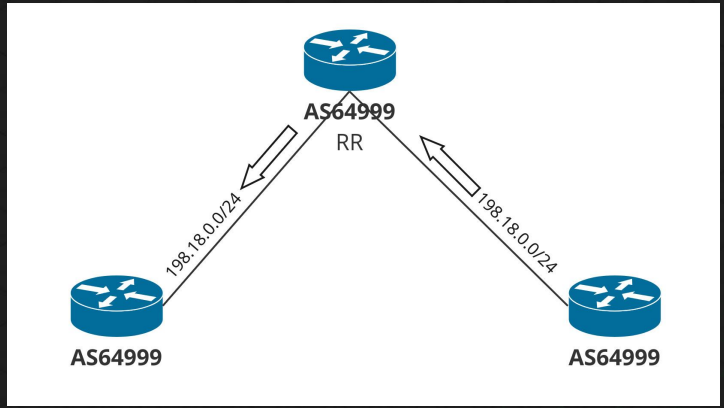
Detalhes mais sórdidos sobre o protocolo estão além do escopo deste treinamento.

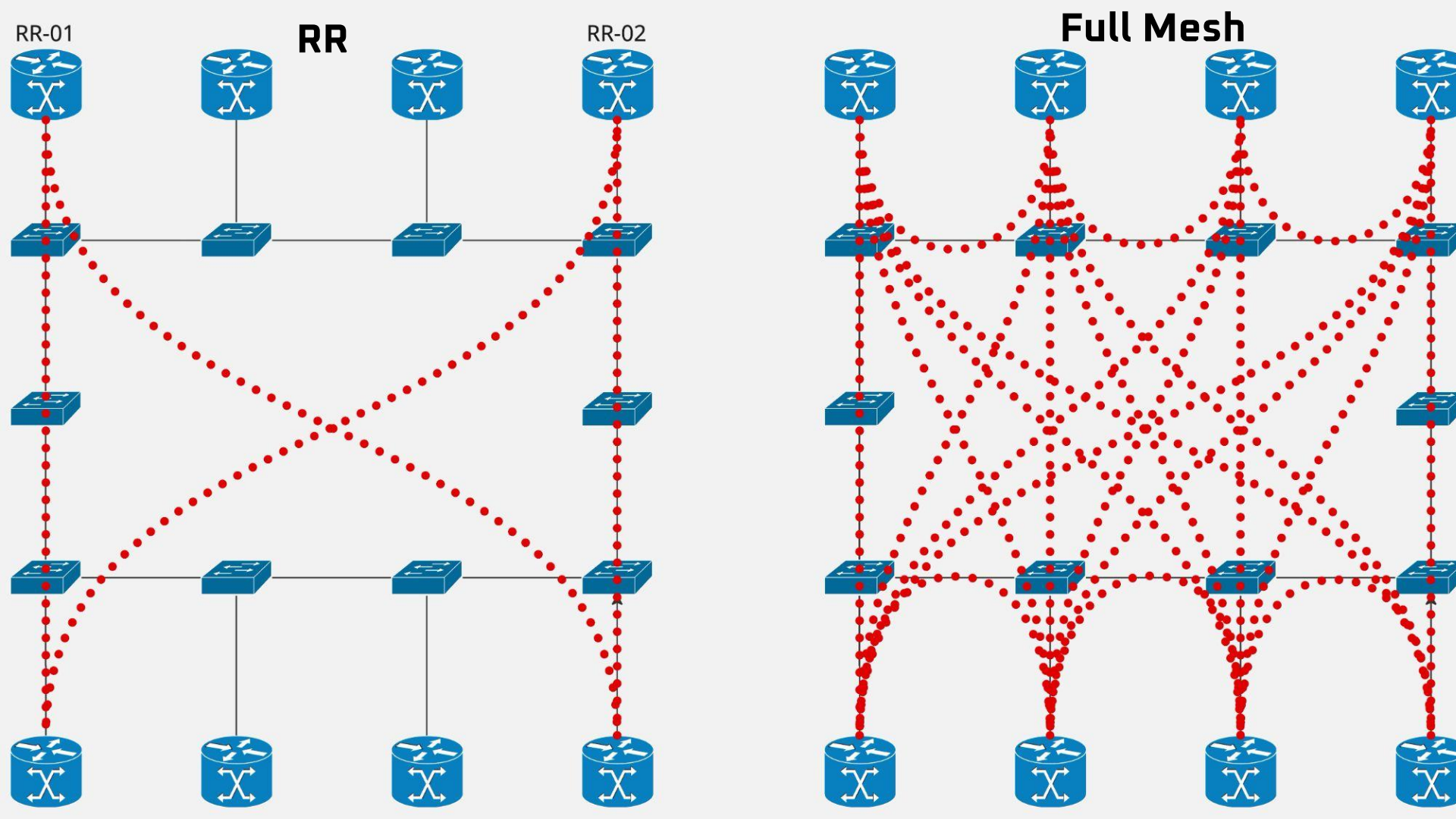
Pera aí, Route Reflector???

Mencionamos o route reflector anteriormente, mas o que seria isso?

De modo geral, podemos dizer que é uma feature que permite sobrescrever o split horizon (proteção de loop) do iBGP para permitir a reflexão de rotas entre sessões BGP com mesmo AS, sendo muito mais escalável do que o full-mesh!

Há mais detalhes, claro, mas isso já é suficiente para continuarmos.




RR-01
RR
RR-02
Full Mesh

MP-BGP AFI/SAFI

O MP-BGP (Multi-Protocol BGP) permite uma série de funcionalidades adicionais ao BGP. Um exemplo disso é a possibilidade de termos sessão BGP IPv6.

AFI = Address Family Identifier
SAFI = Subsequent Address Family Identifier

Existe uma longa lista de AFI e SAFI no abaixo:

https://www.iana.org/assignments/address-family-numbers/address-family-numbers.xhtml

https://www.iana.org/assignments/safi-namespace/safi-namespace.xhtml

MP-BGP AFI/SAFI

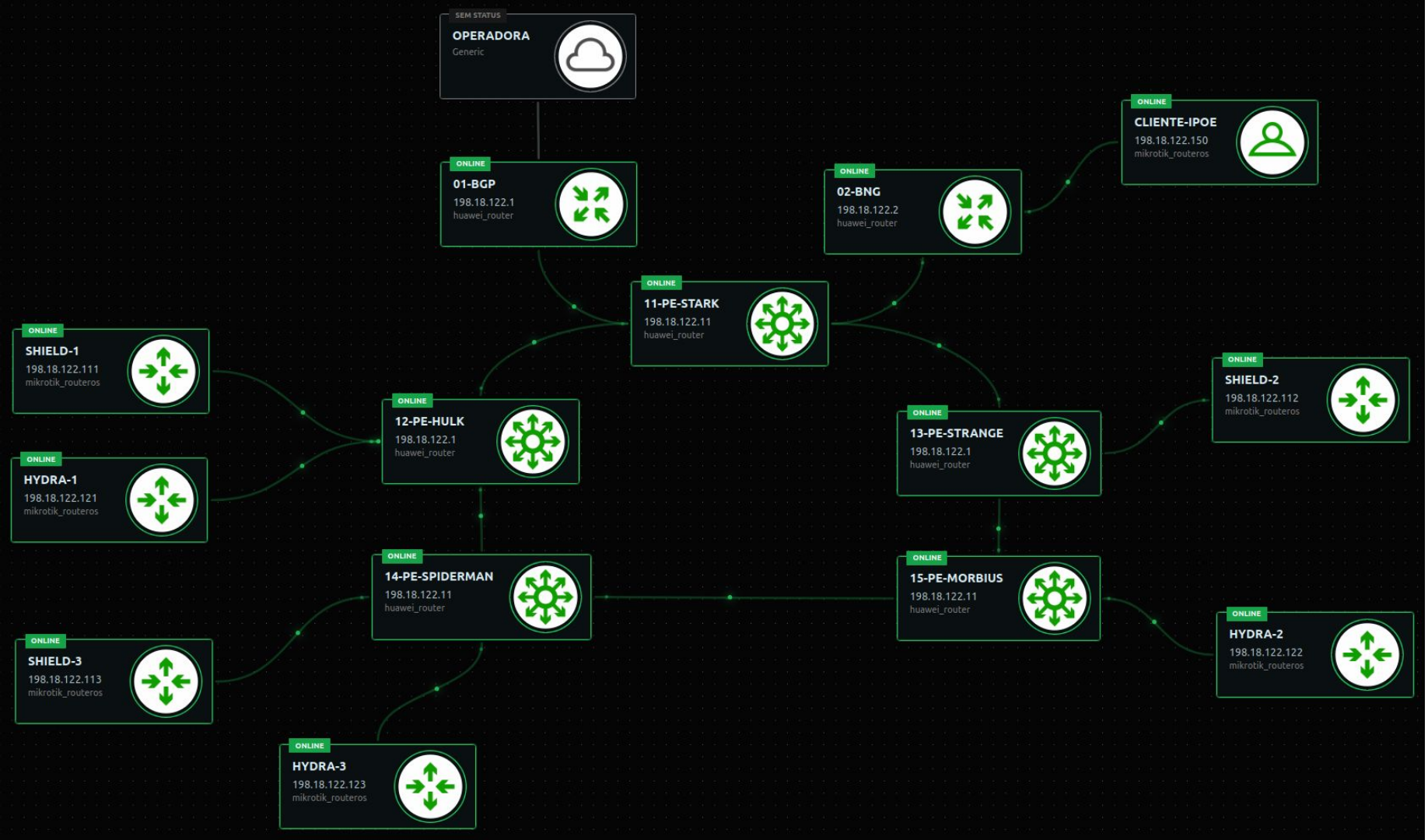
Vamos citar os mais comuns e utilizados:

AFI 1 SAFI 1 = IPv4 Unicast
AFI 1 SAFI 4 = IPv4 Labeled Unicast (BGP-LU)
AFI 1 SAFI 128 = IPv4 MPLS-labeled VPN address (L3VPN)

AFI 2 SAFI 1 = IPv6 Unicast

AFI 25 SAFI 70 = EVPN-L2VPN

AFI 1 SAFI 133 = BGP Flowspec


SEM STATUS
OPERADORA
Generic
ONLINE
01-BGP
198.18.122.1
huawei_router
ONLINE
02-BNG
198.18.122.2
huawei_router
ONLINE
CLIENTE-IPOE
198.18.122.150
mikrotik_routeros
ONLINE
11-PE-STARK
198.18.122.11
huawei_router
ONLINE
SHIELD-1
198.18.122.111
mikrotik_routeros
ONLINE
HYDRA-1
198.18.122.121
mikrotik_routeros
ONLINE
12-PE-HULK
198.18.122.1
huawei_router
ONLINE
13-PE-STRANGE
198.18.122.1
huawei_router
ONLINE
SHIELD-2
198.18.122.112
mikrotik_routeros
ONLINE
14-PE-SPIDERMAN
198.18.122.11
huawei_router
ONLINE
15-PE-MORBIUS
198.18.122.11
huawei_router
ONLINE
SHIELD-3
198.18.122.113
mikrotik_routeros
ONLINE
HYDRA-2
198.18.122.122
mikrotik_routeros
ONLINE
HYDRA-3
198.18.122.123
mikrotik_routeros

Laboratório

Para adiantar, o OSPF já está configurado no backbone dessa maneira:

```
ospf 1 router-id 172.16.0.1
 bandwidth-reference 800000
 area 0.0.0.0

interface LoopBack0
 ip address 172.16.0.1 255.255.255.255
          ospf              enable              1              area              0.0.0.0

interface Ethernet1/0/0
 description CORE
 ip address 10.1.11.1 255.255.255.0
 ospf network-type p2p
 ospf enable 1 area 0.0.0.0
```

Não faz diferença se vai utilizar OSPF ou IS-IS!

A importância do MPLS

Começamos bem, mas falta um carinha muito importante! Sabe quem???

Ele mesmo, o MPLS!!!

Que tal darmos uma pausa no BGP e falarmos um pouco sobre ele?

MPLS

Com poucas exceções, todo encaminhamento de rede é sempre baseado no destino, seja no MAC de destino em redes Layer 2, ou no IP de destino quando nos referimos a uma rede Layer 3.

Não é atoa que as rotas aprendidas por um roteador interferem única e exclusivamente no seu upload, nunca no seu download. Essa premissa também é verdadeira no que tange a MPLS, porém dessa vez baseado no label de destino.

MPLS

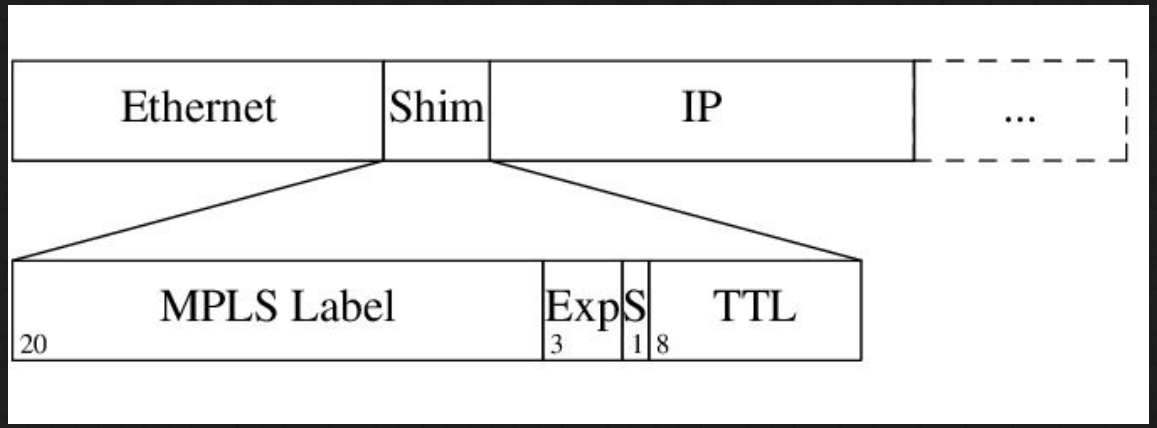
Trocando em miúdos, MPLS vem de Multi-Protocol Label Switching, o qual define um novo ether-type para que a comutação seja baseada em labels (etiquetas).

- 0x0800 – IPv4;
- 0x86DD – IPv6;
- 0x8847 – MPLS;
- 0x8848 – MPLS with upstream-assigned label.

Entender que o ether-type é alterado e como a forma que os pacotes são interpretados é de grande valia para o entendimento da tecnologia!

Shim Header

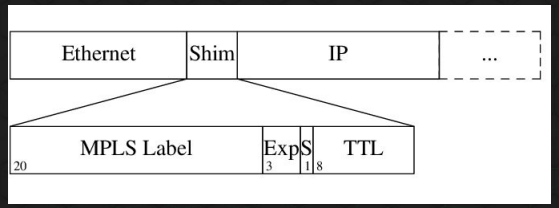
Para que o MPLS seja possibilitado, é adicionado um campo entre o cabeçalho ethernet e o cabeçalho IP chamado “Shim Header”. Por esse motivo, o MPLS é conhecido como “Protocolo de camada 2,5”



Shim Header

O Shim Header é um campo de 32 bits (4 bytes), tendo contido nele o Label, Exp bit, Stack e TTL.

- O campo MPLS Label contém 20 bits, e é responsável pelo número do label que será utilizado, contendo 1048576 possibilidades;
- O campo Exp (Experimental bit) é um campo de 3 bits para utilização de QoS;
- O campo Stack (Ou Bottom of Stack) serve para informar ao equipamento se existem labels empilhados (explicado mais à frente);
- O campo TTL segue a mesma lógica do TTL do IPv4, onde o TTL do pacote IPv4 é copiado para o TTL do Shim Header



Mas o que é um Label?

- Identificador numérico;
- 2^20;
- 1048576 labels;
- Existem espaços reservados;
- Significado apenas entre vizinhos;
- Pode se repetir entre equipamentos (situação comum, e não problemática);
- Gerenciado através de operações (PUSH/SWAP/POP, explicado mais à frente).

Labels reservados

- 0 IPv4 Explicit NULL Label - [RFC3032]
- 1 Router Alert Label - [RFC3032]
- 2 IPv6 Explicit NULL Label - [RFC3032]
- 3 Implicit NULL Label - [RFC3032]
- 4-6 Unassigned
- 7 Entropy Label Indicator (ELI) - [RFC6790]
- 8-12 Unassigned
- 13 Generic Associated Channel Label [RFC5586]
- 14 OAM Alert Label - [RFC3429]
- 15 Extension Label (XL) - [RFC7274][RFC9017]

Protocolos de Distribuição de Label

De nada adianta o equipamento gerar labels, mas não distribuí-las pela rede. Com a distribuição de labels passamos a criar os LSPs (Label Switched Paths). Para essa finalidade existem alguns protocolos:

- LDP;
- RSVP;
- BGP;
- IGP Link-State (Segment Routing).

Também é possível criar LSPs estáticos, porém isso não é nada escalável!

LDP - Label Distribution Protocol

Nesse momento, vamos entender o funcionamento do LDP.

O LDP (Label Distribution Protocol), definido na RFC5036, é um protocolo que atua na porta 646/UDP durante a fase de discovery, sendo que a sessão depois disso é mantida na porta 646/TCP. O LDP depende de um protocolo de roteamento dinâmico do tipo link-state (OSPF ou IS-IS) para funcionar, entretanto também é possível gerar labels através de rotas estáticas.

Sua configuração é bastante simples, tendo apenas um adicional quando necessário para se comunicar com equipamentos que não estejam diretamente conectados (LDP Targeted). Essa configuração adicional será citada no módulo de L2VPN.

Laboratório

Para adiantar, o OSPF já está configurado no backbone dessa maneira:

```
ospf 1 router-id 172.16.0.1
 bandwidth-reference 800000
 area 0.0.0.0

interface LoopBack0
 ip address 172.16.0.1 255.255.255.255
          ospf              enable              1              area              0.0.0.0

interface Ethernet1/0/0
 description CORE
 ip address 10.1.11.1 255.255.255.0
 ospf network-type p2p
 ospf enable 1 area 0.0.0.0
```

Não faz diferença se vai utilizar OSPF ou IS-IS!

Y = ID DO EQUIPAMENTO

Laboratório

Vamos configurar o MPLS e o LDP em todo backbone:

```
mpls lsr-id 172.16.0.Y
mpls
quit

mpls ldp
quit
commit
```

*Nas interfaces de backbone:

```
mpls
mpls ldp
commit
```

Podemos verificar com os comandos abaixo se deu certo:

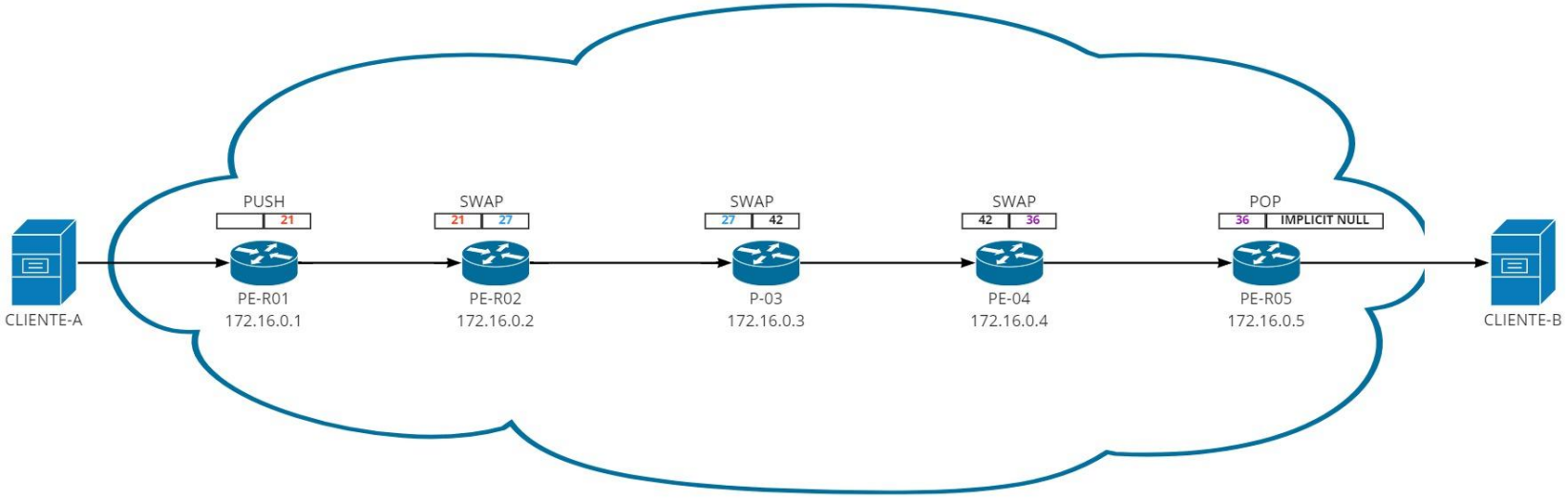
```
display mpls ldp peer
display mpls lsp
display tcp status local-port 646
display tcp status remote-port 646
```

Operações do MPLS

Para que o MPLS funcione, algumas operações são adicionadas no encaminhamento, sendo elas:

- PUSH – Põe um label;
- SWAP – Troca o label;
- POP – Tira o label.

Operações do MPLS



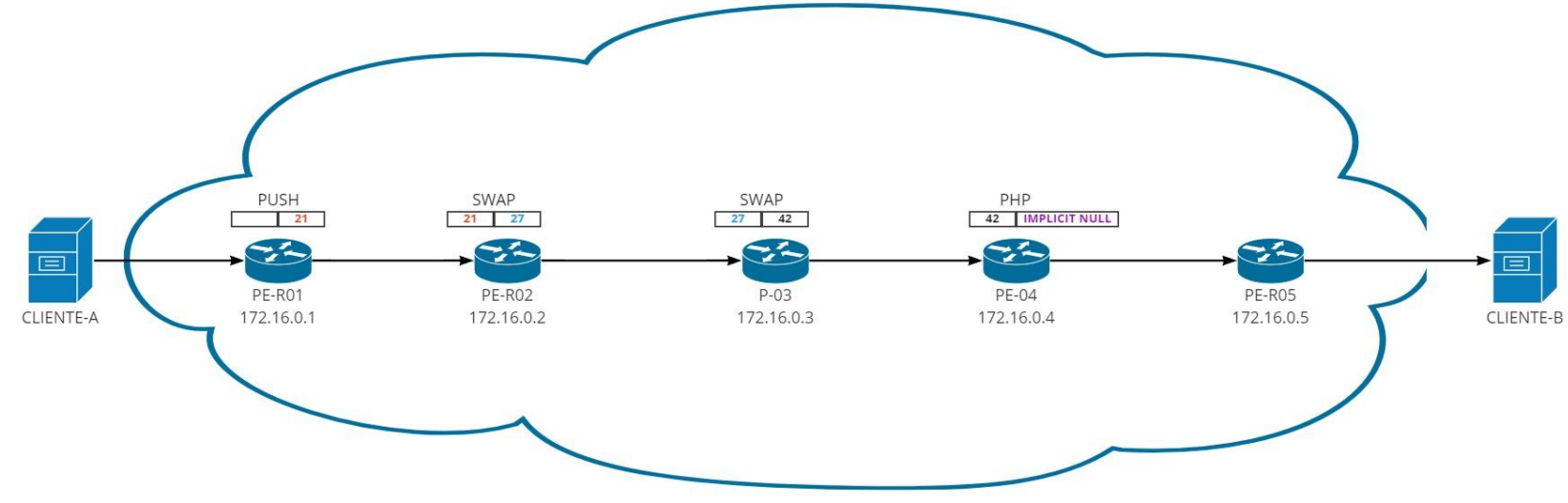
Penultimate Hop Popping (PHP)

Apesar de os processos de PUSH, SWAP e POP já terem grande relevância no backbone, ainda foi criado um critério a mais para a ação de POP, chamado de “PHP”.

Essa operação consiste em realizar a operação de POP (remoção do label) no penúltimo salto, e não no último.

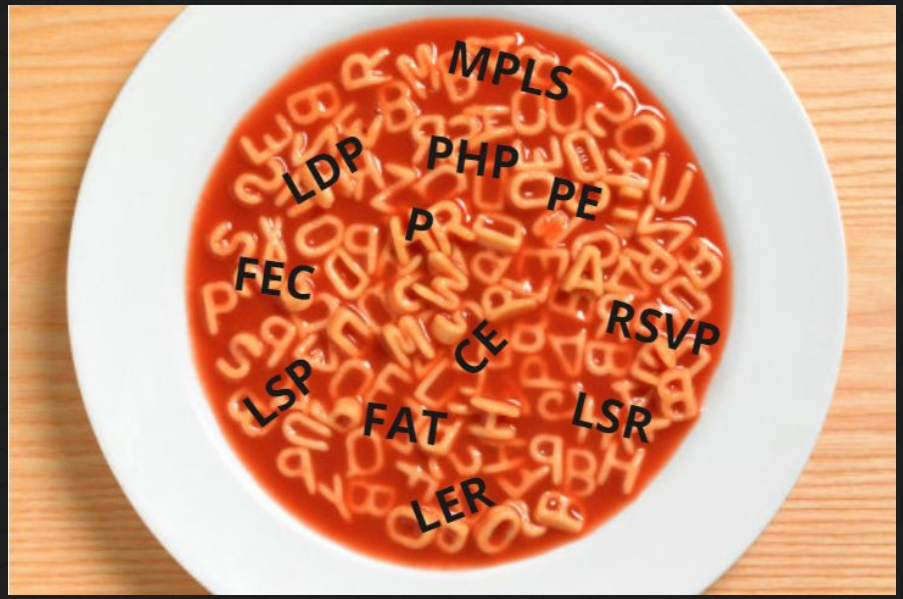
Esse processo pode ser desativado marcando a opção “explicit null”.

Penultimate Hop Popping (PHP)



Acrônimos MPLS

Se preparem para a sopa de letrinhas!

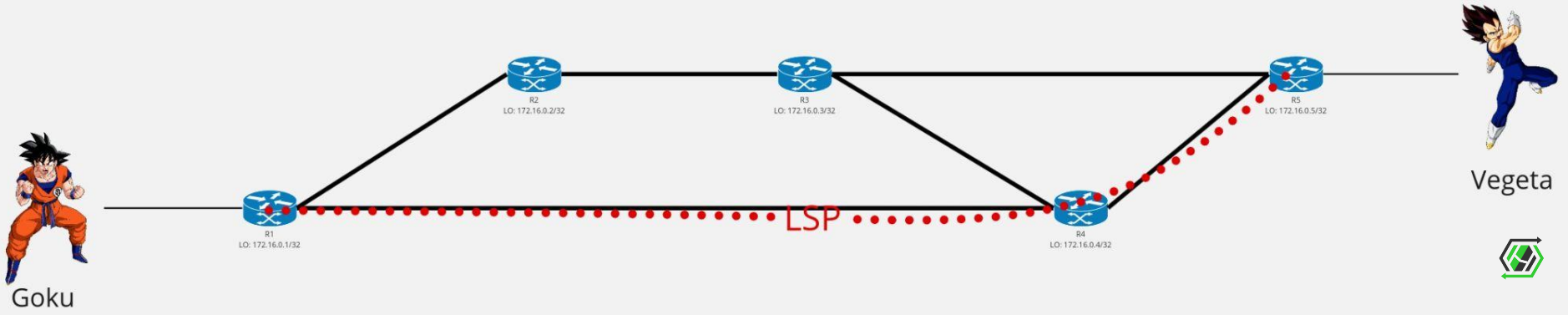


Acrônimos MPLS

MPLS	-	Multi-Protocol Label Switching
LDP	-	Label Distribution Protocol
T-LDP	-	Targeted Label Distribution Protocol
RSVP	-	Resource Reservation Protocol
PHP	-	Penultimate Hop Popping
FEC	-	Forwarding Equivalence Class
FAT	-	Flow-Aware Transport
LSP	-	Label Switched Path
LER	-	Label Edge Router
LSR	-	Label Switch Router
E-LSR	-	Eggres Label Switch Router
I-LSR	-	Ingress Label Switch Router
P	-	Provider
CE	-	Customer Edge
PE	-	Provider Edge
CPE	-	Customer Premise Edge

Label Switched Path

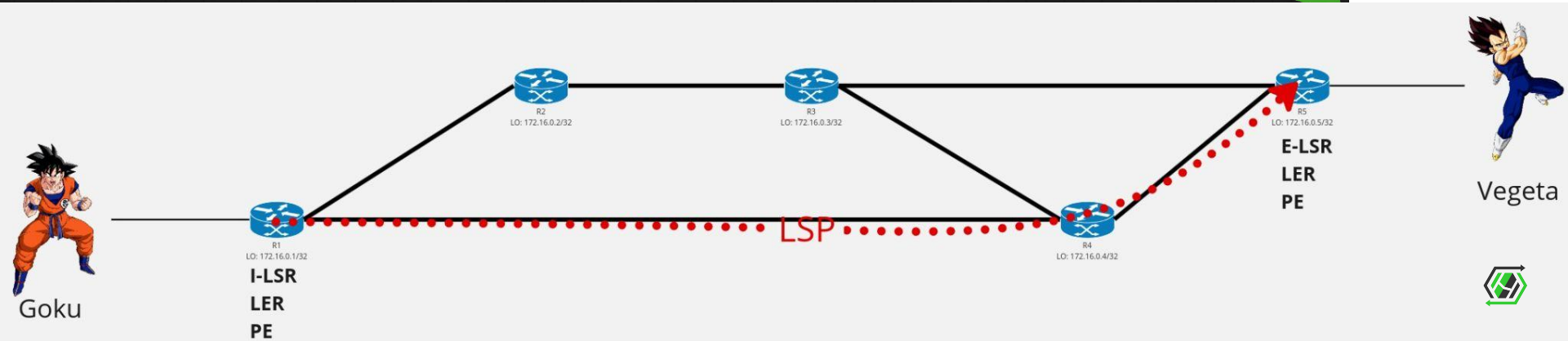
LSP (Label Switched Path) é um caminho pré-definido em uma rede MPLS, criado para direcionar pacotes de dados usando labels.



Label Edge Router / PE

Um Label Edge Router (LER) é um dispositivo situado na borda de uma rede MPLS, responsável por iniciar e terminar os caminhos de labels (LSPs).

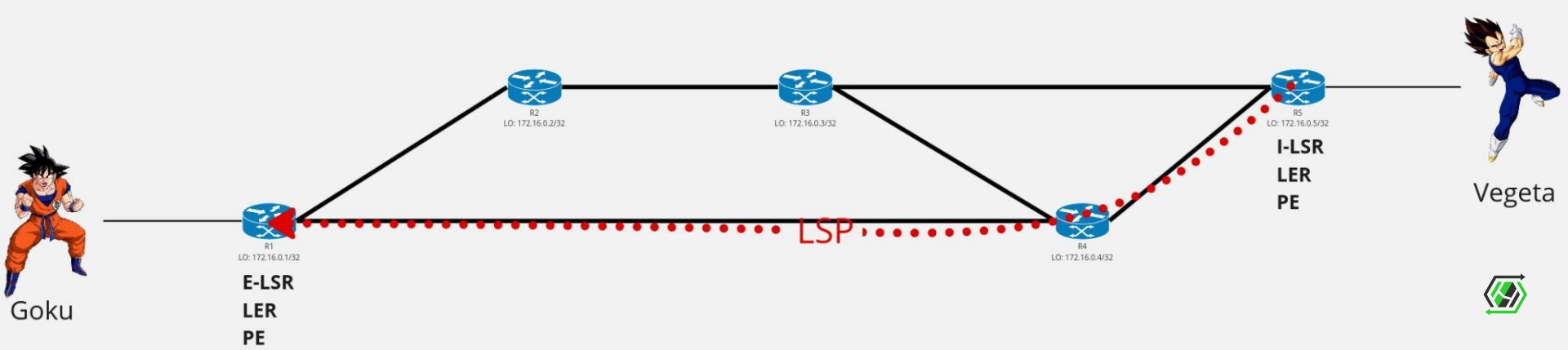
Também conhecido como PE, e em diversas literaturas citado como E-LSR ou I-LSR, dependendo da função empenhada na rede.



Label Edge Router / PE

Um Label Edge Router (LER) é um dispositivo situado na borda de uma rede MPLS, responsável por iniciar e terminar os caminhos de labels (LSPs).

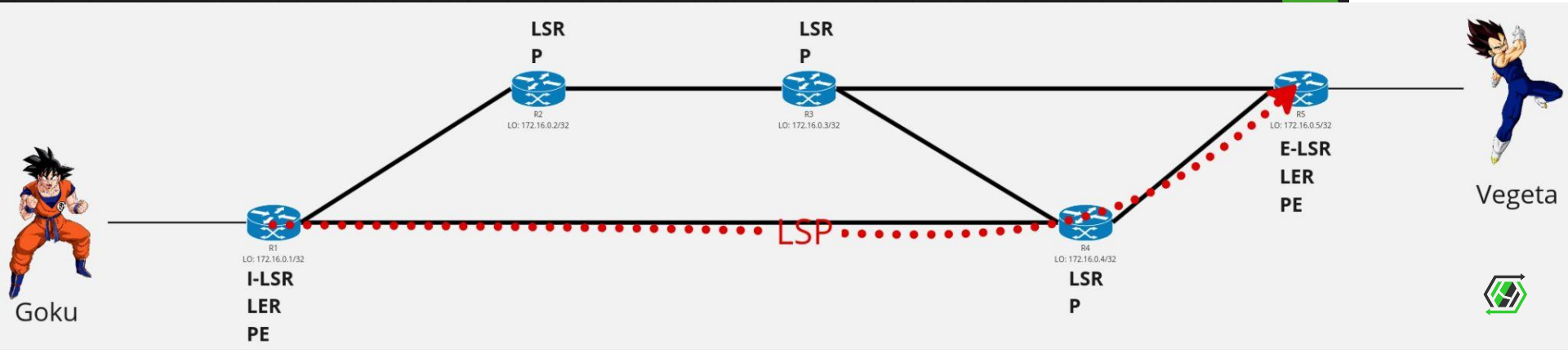
Também conhecido como PE, e em diversas literaturas citado como E-LSR ou I-LSR, dependendo da função empenhada na rede.



Label Switched Router / P

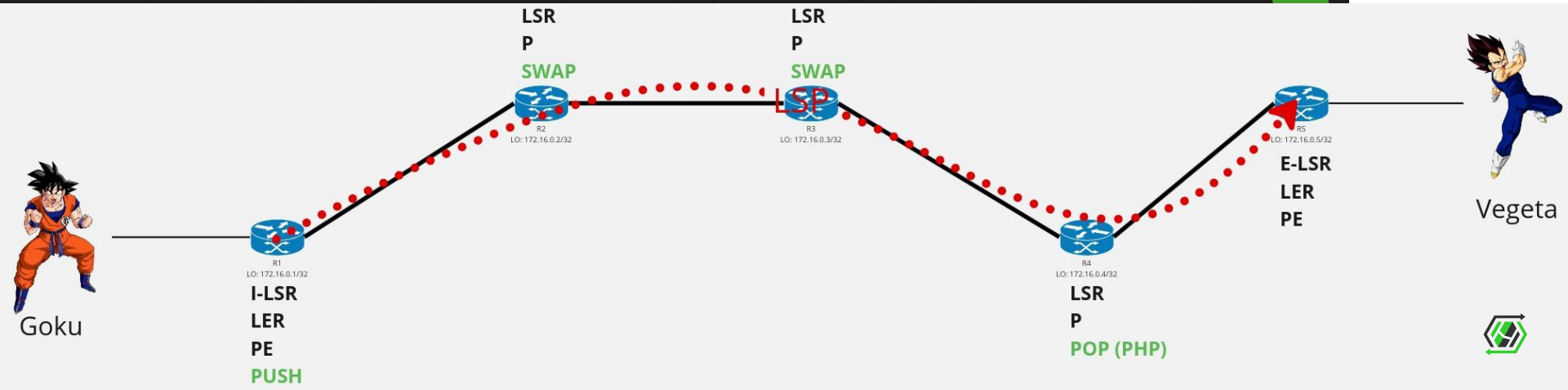
Um Label Switched Router (LSR) é um host que atua primordialmente realizando a operação SWAP.

Também conhecido como P, e em diversas literaturas citado como Transit-LSR.



Colocando tudo Junto

Na topologia abaixo temos os acrônimos citados, mostrando inclusive as operações no MPLS.



Forwarding Equivalence Class

FEC (Forwarding Equivalence Class) é um conceito no MPLS que agrupa pacotes de dados que compartilham os mesmos requisitos de encaminhamento, como o mesmo destino, políticas de QoS ou rota.

Uma vez determinado, um rótulo MPLS é atribuído a cada FEC, permitindo que todos os pacotes dentro dessa classe sejam encaminhados da mesma maneira através da rede. Isso simplifica o processo de encaminhamento e melhora a eficiência na entrega de dados.

De certa forma, é análogo a uma rota, porém com label.

```
[~PE01-TONY-STARK-NE]dis mpls lsp
Flag after Out IF: (I) - RLFA Iterated LSP, (I*) - Normal and RLFA Iterated LSP
Flag after LDP FRR: (L) - Logic FRR LSP
------------------------------------------------------------------------------
                LSP Information: LDP LSP
------------------------------------------------------------------------------
FEC                 In/Out Label    In/Out IF                       Vrf Name
1.255.255.1/32      3/NULL          -/-
1.255.255.2/32      NULL/3          -/Eth1/0/1
1.255.255.2/32      48120/3         -/Eth1/0/1
```

Sumário L3VPN

- Underlay e overlay;
- VRF - Virtual Routing and Forwarding;
- VRF Lite;
- Route Distinguisher;
- Route Target;
- Otimizando uso de labels na VRF;
- Laboratório L3VPN;
- Laboratório 6VPE;
- Laboratório Hydra x S.H.I.E.L.D;

Underlay e Overlay

É importante entendermos um pouco de underlay e overlay antes de continuarmos.

Fazendo uma analogia, se considerarmos as ruas como o underlay, então as motos e carros poderiam ser nosso overlay! (e o Capitão América, quem sabe, um pacote?!)

O underlay é a base, a estrutura. No nosso caso, fibras, equipamentos, protocolos como OSPF e IS-IS, todos formam o underlay.

O overlay seriam os serviços operados sobre o underlay, como VXLAN, L2VPN, L3VPN, EVPN, por aí vai.

VRF + BGP = L3VPN?

VPNs MPLS de camada 3 fazem uso das chamadas VRF's (Virtual Routing and Forwarding), que podem ser resumidas a tabelas de roteamento virtualizadas e segregadas da tabela principal (main) do equipamento.

Ou seja, o comportamento padrão é o de rotas presentes numa VRF não serem vistas por aquelas que estiverem na tabela main e vice-versa. Vale ressaltar que há como fazer o leak (vazamento) de rotas entre as tabelas, porém isso quebra o conceito do isolamento e requer cuidado ao ser feito.

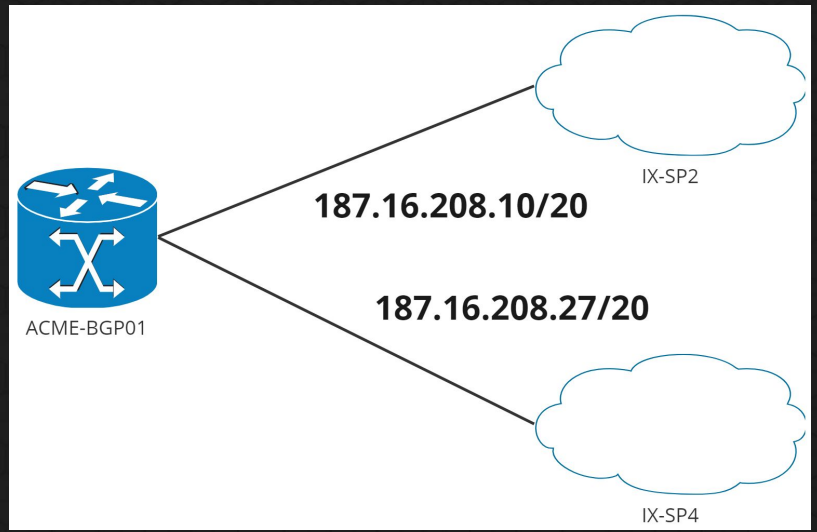
As L3VPNs também fazem uso do BGP para operarem, sendo definidas pelas famílias de endereço AFI 1 e SAFI 128 no IPv4 e AFI 2 SAFI 128 para IPv6.

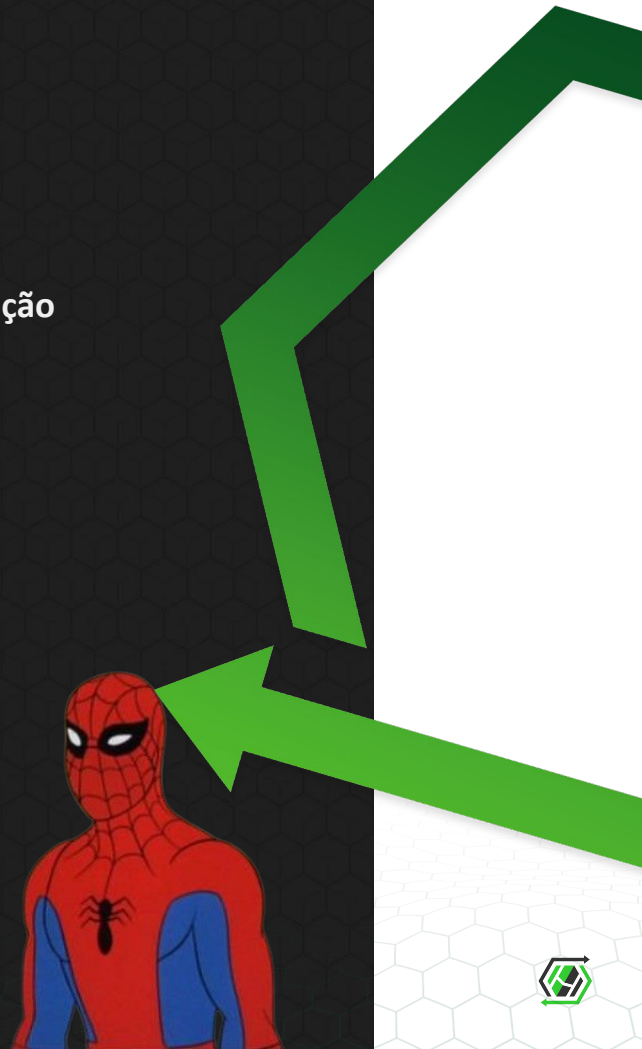
Muito importante: essa feature precisa de licença em alguns casos! Roteadores NE40 e NE8K são exemplos. Por outro lado, os switches Huawei que suportam MPLS já vêm com essa feature liberada de fábrica.

VRF Lite

As VRFs podem ser utilizadas em ambientes sem MPLS, sendo esta aplicação chamada de VRF Lite!

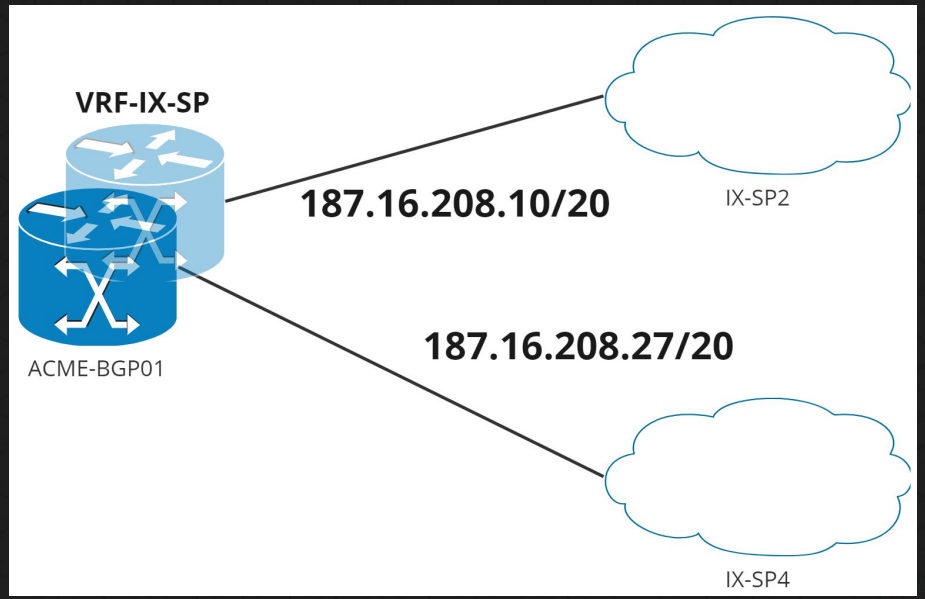
CASE: ativar 2 abordagens IX-SP na mesma caixa, é possível?





VRF Lite

Nesse caso, por exemplo, não podemos configurar a mesma sub-rede do IX em duas interfaces do roteador que façam parte da mesma tabela de roteamento! Solução: crie uma tabela separada (VRF-IX-SP) e configure lá!



Route Distinguisher

Rotas dentro de L3VPN's precisam ser diferenciadas umas das outras, o que é feito por meio dos RDs (Route Distinguishers), presentes tanto nos endereços VPNv4 quanto VPNv6. Eles possuem 64 bits e alguns formatos possíveis, sendo o mais comum aquele semelhante ao das communities BGP.

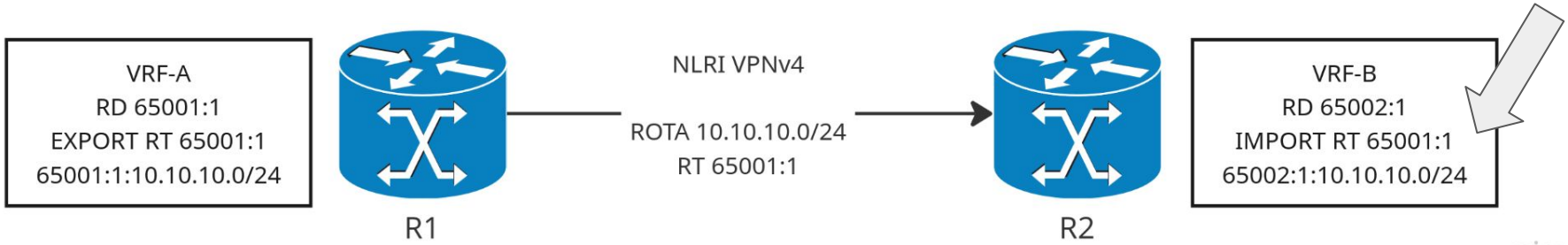
- Type 0 => 65000:100
- Type 1 => 192.168.10.1:10
- Type 2 => 919293:20 ou 20:919293

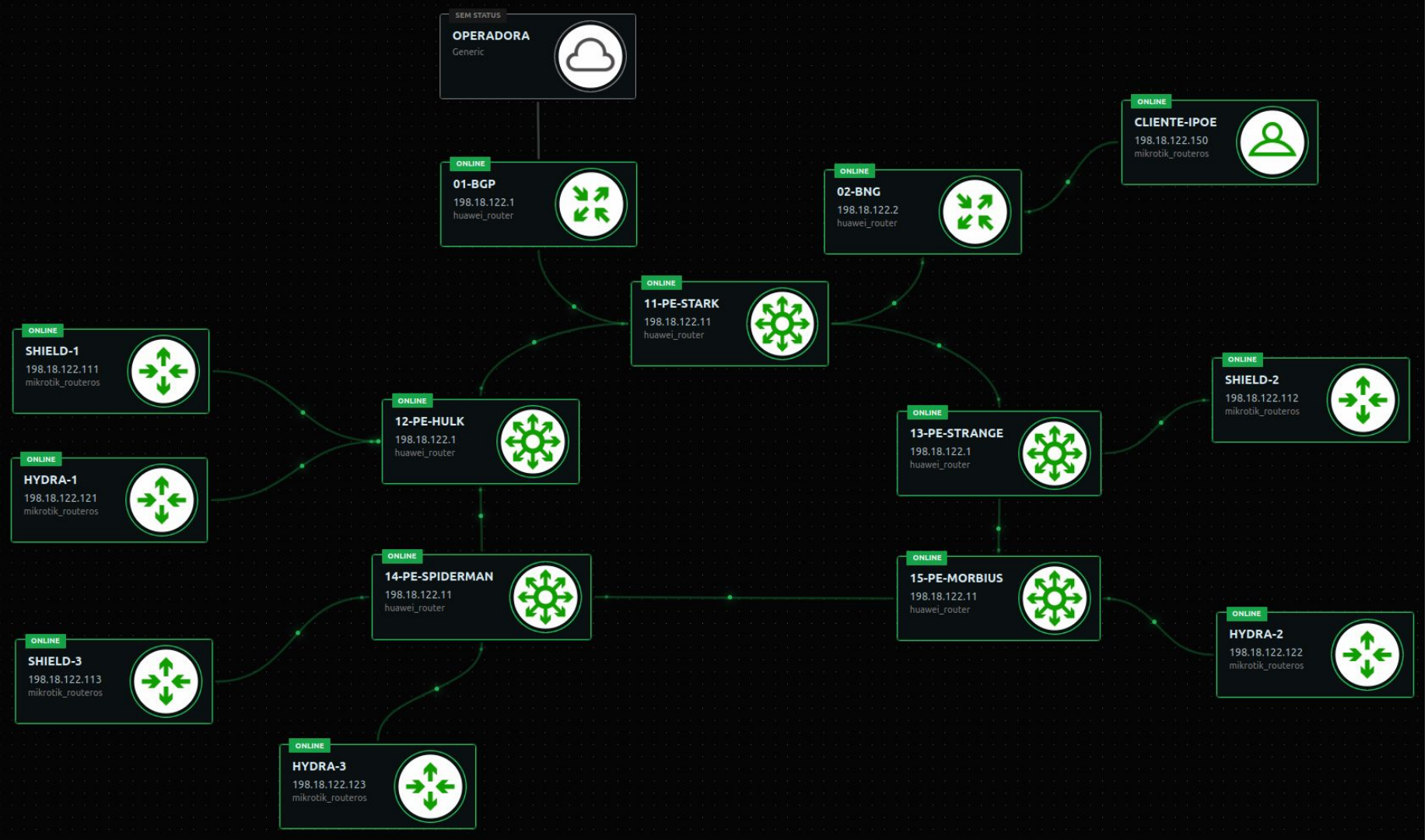
Uma consideração importante é a utilização de recursos do equipamento, visto que as rotas VPNv4 possuem 96 bits de tamanho, enquanto as rotas VPNv6 possuem 192 bits!

Route Target

O RT (Route Target) é uma extended community do BGP que possui 8 bytes de tamanho e funciona semelhantemente às communities tradicionais de 4 bytes, mas que desempenha papel especial na estrutura do L3VPN permitindo o "encaixe" das rotas nas respectivas tabelas quando elas são transportadas de um equipamento para outro via VPNv4 ou VPNv6.

Abaixo, o R2 encaixa a rota VPNv4 10.10.10.0/24 na VRF-B por conta do import do RT 65001:1!




SEM STATUS
OPERADORA
Generic
ONLINE
01-BGP
198.18.122.1
huawei_router
ONLINE
02-BNG
198.18.122.2
huawei_router
ONLINE
CLIENTE-IPOE
198.18.122.150
mikrotik_routeros
ONLINE
11-PE-STARK
198.18.122.11
huawei_router
ONLINE
SHIELD-1
198.18.122.111
mikrotik_routeros
ONLINE
HYDRA-1
198.18.122.121
mikrotik_routeros
ONLINE
12-PE-HULK
198.18.122.1
huawei_router
ONLINE
13-PE-STRANGE
198.18.122.1
huawei_router
ONLINE
SHIELD-2
198.18.122.112
mikrotik_routeros
ONLINE
14-PE-SPIDERMAN
198.18.122.11
huawei_router
ONLINE
15-PE-MORBIUS
198.18.122.11
huawei_router
ONLINE
SHIELD-3
198.18.122.113
mikrotik_routeros
ONLINE
HYDRA-2
198.18.122.122
mikrotik_routeros
ONLINE
HYDRA-3
198.18.122.123
mikrotik_routeros

Laboratório

Vamos fechar VPNv4 do lado do BNG01 com o BGP01:

BNG01:

```
bgp 1
 undo default ipv4-unicast <<< desativa ipv4-unicast default
 group RR internal
 peer RR connect-interface loopback 0
 peer 172.16.0.1 group RR
 peer 172.16.0.1 description BGP01
 ipv4-family vpnv4
  peer RR enable
  peer RR advertise-community
  peer 172.16.0.1 group RR
```

Verificar: display bgp vpnv4 all peer / display bgp all summary
display current-configuration configuration bgp

Laboratório

Vamos simular o BNG com DHCP:

BNG01:

```
dhcp enable

ip vpn-instance VRF-INTERNET
 ipv4-family
 route-distinguisher 65001:0
 vpn-target 65001:0 both <<< import + export

ip pool clientes server
 vpn-instance VRF-INTERNET
 gateway 1.0.0.1 255.255.255.0
 section 10 1.0.0.2 1.0.0.254
 lease 0 0 10
```

Laboratório

Vamos simular o BNG com DHCP:

BNG01:

```
bgp 1
 ipv4-family vpn-instance VRF-INTERNET
  import-route direct <<< redistribuindo rotas diretamente conectadas

interface Ethernet1/0/0
 description CLIENTES
 ip binding vpn-instance VRF-INTERNET
 ip address 1.0.0.1 255.255.255.0
```

Verificar:

```
display arp interface ethernet 1/0/0
display ip-pool pool-usage all
ping -vpn-instance VRF-INTERNET x.x.x.x
```

Laboratório

Vamos fechar agora VPNv4 do BGP01 com o BNG01:

BGP01:

```
bgp 1
 undo default ipv4-unicast
 group BNG internal
 peer BNG connect-interface loopback0
 peer 172.16.0.2 group BNG
 peer 172.16.0.2 description BNG01
 ipv4-family vpnv4
  peer BNG enable
  peer BNG reflect-client
  peer BNG advertise-community
  peer 172.16.0.2 group BNG
```

Verificar: display bgp vpnv4 all peer / display bgp all summary

Laboratório

Agora vamos trazer Internet para a VRF:

BGP01:

```
ip vpn-instance VRF-INTERNET
 ipv4-family
 route-distinguisher 65001:0
 vpn-target 65001:0 both <<< import + export

interface Ethernet 1/0/1
 description OPERADORA
 ip binding vpn-instance VRF-INTERNET
 ip address 203.0.113.2 30
```

Laboratório

Adicionemos o IP na interface:

BGP01

```
ip route-static vpn-instance VRF-INTERNET 1.0.0.0 8 null 0

bgp 1
 ipv4-family vpn-instance VRF-INTERNET
  peer 203.0.113.1 as-number 2
  network 1.0.0.0 8
```

Verificar:

```
display bgp vpnv4 all peer / display bgp all summary
display bgp vpnv4 vpn-instance VRF-INTERNET routing-table peer
203.0.113.1 advertised-routes
```

Wireshark, Baby!

R02-BNG_Ethernet1/0/1

```
▸ Frame 136: 103 bytes on wire (824 bits), 103 bytes captured (824 bits) on interface eth0, id 0
▸ Ethernet II, Src: 00:7c:59:83:bb:01 (00:7c:59:83:bb:01), Dst: 00:66:03:0d:01:01 (00:66:03:0d:01:01)
▸ MultiProtocol Label Switching Header, Label: 48120, Exp: 6, S: 1, TTL: 255
▸ Internet Protocol Version 4, Src: 172.16.0.2, Dst: 172.16.0.1
▸ Transmission Control Protocol, Src Port: 52198, Dst Port: 179, Seq: 1, Ack: 1, Len: 45
▾ Border Gateway Protocol - OPEN Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 45
    Type: OPEN Message (1)
    Version: 4
    My AS: 1
    Hold Time: 180
    BGP Identifier: 10.2.11.2
    Optional Parameters Length: 16
  ▾ Optional Parameters
    ▾ Optional Parameter: Capability
        Parameter Type: Capability (2)
        Parameter Length: 14
      ▾ Capability: Multiprotocol extensions capability
          Type: Multiprotocol extensions capability (1)
          Length: 4
          AFI: IPv4 (1)
          Reserved: 00
          SAFI: Labeled VPN Unicast (128)
      ▾ Capability: Route refresh capability
          Type: Route refresh capability (2)
          Length: 0
      ▾ Capability: Support for 4-octet AS number capability
          Type: Support for 4-octet AS number capability (65)
          Length: 4
          AS Number: 1
```

Otimizando o Uso de Labels na L3VPN

O comportamento default da L3VPN é de gerar labels para cada uma das FECs, entretanto isso pode levar a um grande consumo de recursos se você tiver muitas rotas VPNv4/v6.

E faz sentido ter um label para cada rota da VRF? Provavelmente não, exceto no caso de haver a necessidade de tratar algumas delas individualmente.

Uma forma de otimizar o uso do recurso de labels é aplicar o comando apply-label per-instance, fazendo com que um label seja utilizado para todas as rotas de determinada L3VPN. Isso pode ser muito interessante no cenário onde o "Full Route" estiver sendo recebido dentro da L3VPN, constituindo uma topologia de rede extremamente escalável quando bem configurada.

Otimizando o Uso de Labels na L3VPN

```
ip vpn-instance VRF-CLIENTES
 ipv4-family
  route-distinguisher 65000:65000
  apply-label per-instance
  vpn-target 65000:65000 export-extcommunity
  vpn-target 65000:65000 import-extcommunity
```

6VPE

O atual protocolo da Internet, o IPv6, não poderia ficar de fora, não é mesmo? E como implementamos o suporte ao IPv6 no L3VPN?

De forma semelhante ao que é feito no 6PE, podemos reaproveitar os backbones IPv4-only para estabelecer a comunicação da família VPNv6 entre os participantes!

A recursão ocorre para NEXT_HOPs IPv4-mapped IPv6 e o encaminhamento se dá através de LSPs MPLS

- ::FFFF:X.0.0.Y
- ::FFFF:1.0.0.4

Laboratório

Ativando a família VPNv6 entre o BGP01 e o BNG01:

BNG01

```
bgp 1
 ipv6-family vpnv6
  peer RR enable
  peer RR advertise-community
  peer 172.16.0.1 group RR
```

BGP01

```
bgp 1
 ipv6-family vpnv6
  peer BNG enable
  peer BNG reflect-client
  peer BNG advertise-community
  peer 172.16.0.2 group BNG
```

*Ativar nova família = flap na sessão BGP, ok!

Verificar: display bgp vpnv4 all peer / display bgp all summary

Laboratório

Ativando o IPv6 na VRF-INTERNET no BNG01:

BNG01:

```
ip vpn-instance VRF-INTERNET
 ipv6-family
  route-distinguisher 65001:0
  vpn-target 65001:0 export-extcommunity
  vpn-target 65001:0 import-extcommunity

bgp 1
 ipv6-family vpn-instance VRF-INTERNET
  import-route direct << redistribuindo rotas diretamente conectadas*

interface Ethernet 1/0/0
 ipv6 enable
 ipv6 address 2001:1:beba:cafe::1/64
```

Laboratório

Ativando o IPv6 na VRF-INTERNET no BGP01:

BGP01:

```
ipv6 route-static vpn-instance VRF-INTERNET 2001:1:: 32 null0

ip vpn-instance VRF-INTERNET
 ipv6-family
  route-distinguisher 65001:0
  vpn-target 65001:0 export-extcommunity
  vpn-target 65001:0 import-extcommunity

bgp 1
 ipv6-family vpn-instance VRF-INTERNET
  network 2001:1:: 32
  peer 2001:2:cafe::1 as-number 2
```

Laboratório

Ativando o IPv6 na VRF-INTERNET no BGP01:

BGP01:

```
interface Ethernet 1/0/1
 ipv6 enable
 ipv6 address 2001:2:cafe::2/64
```

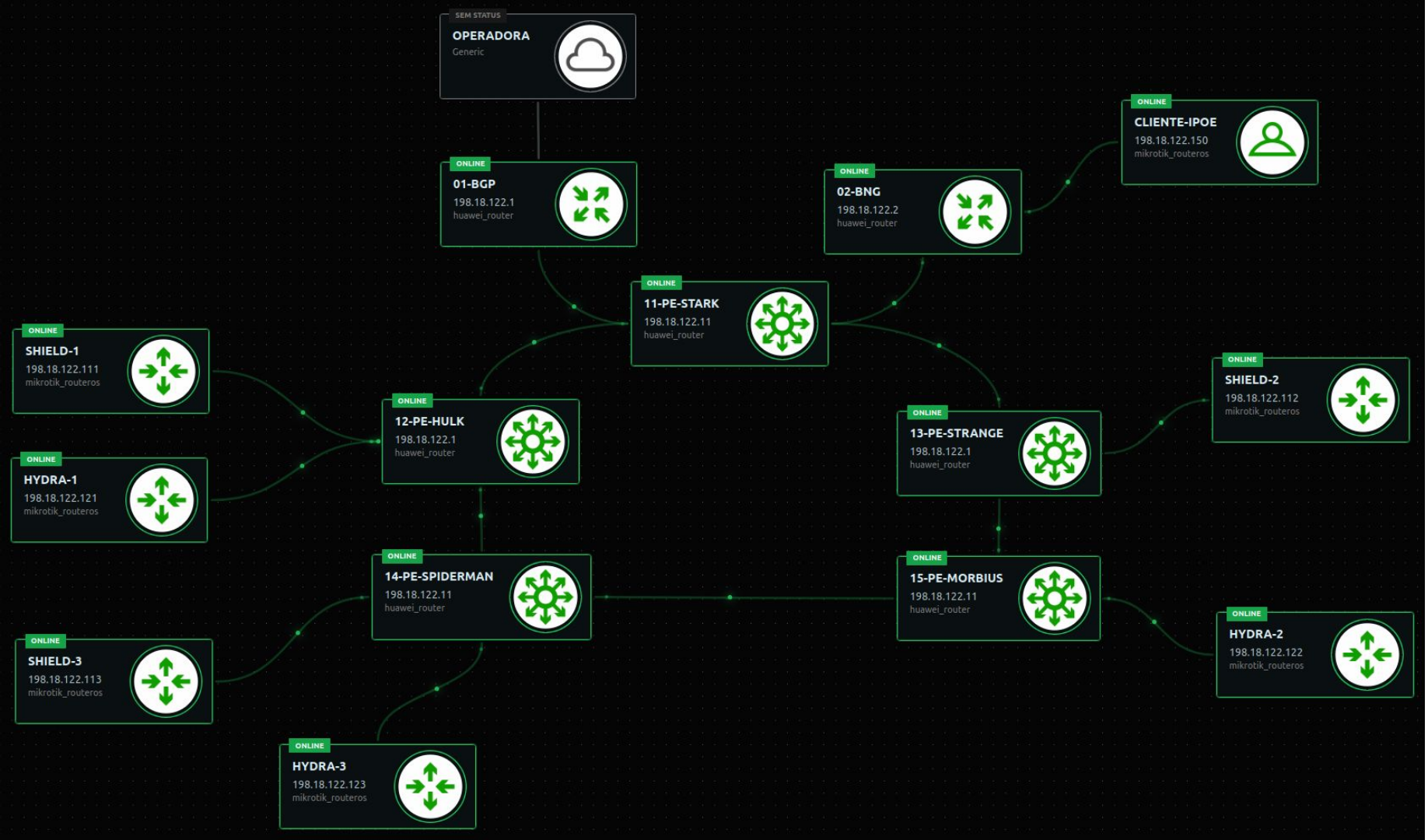
Verificar:

```
display bgp vpnv6 all peer / display bgp all summary
display bgp vpnv6 vpn-instance VRF-INTERNET routing-table
```

S.H.I.E.L.D x Hydra

A S.H.I.E.L.D e a Hydra contrataram nosso provedor para comunicarmos suas unidades! Como são inimigas mortais, suas informações de roteamento precisam ser isoladas umas das outras, caso contrário o caos poderia se instalar.

Vamos implementar isso utilizando L3VPN?


SEM STATUS
OPERADORA
Generic
ONLINE
01-BGP
198.18.122.1
huawei_router
ONLINE
02-BNG
198.18.122.2
huawei_router
ONLINE
CLIENTE-IPOE
198.18.122.150
mikrotik_routeros
ONLINE
11-PE-STARK
198.18.122.11
huawei_router
ONLINE
SHIELD-1
198.18.122.111
mikrotik_routeros
ONLINE
HYDRA-1
198.18.122.121
mikrotik_routeros
ONLINE
12-PE-HULK
198.18.122.1
huawei_router
ONLINE
13-PE-STRANGE
198.18.122.1
huawei_router
ONLINE
SHIELD-2
198.18.122.112
mikrotik_routeros
ONLINE
14-PE-SPIDERMAN
198.18.122.11
huawei_router
ONLINE
15-PE-MORBIUS
198.18.122.11
huawei_router
ONLINE
SHIELD-3
198.18.122.113
mikrotik_routeros
ONLINE
HYDRA-2
198.18.122.122
mikrotik_routeros
ONLINE
HYDRA-3
198.18.122.123
mikrotik_routeros

Laboratório

Primeiro, vamos estabelecer a comunicação do refletor BGP01 com os PE's:

BGP01:

```
bgp 1
 group PE internal
 peer PE connect-interface loopback0
 peer 172.16.0.11 group PE
 peer 172.16.0.11 description PE11-STARK
 peer 172.16.0.12 group PE
 peer 172.16.0.12 description PE12-HULK
 peer 172.16.0.13 group PE
 peer 172.16.0.13 description PE13-STRANGE
 peer 172.16.0.14 group PE
 peer 172.16.0.14 description PE14-SPIDERMAN
 peer 172.16.0.15 group PE
 peer 172.16.0.15 description PE15-MORBIUS
```

Laboratório

Primeiro, vamos estabelecer a comunicação do refletor BGP01 com os PE's:

BGP01:

```
bgp 1
 ipv4-family vpnv4
  undo policy vpn-target <<< para instalar rotas das VRF's sem precisar criá-las;
  peer PE enable
  peer PE reflect-client
  peer PE advertise-community
  peer 172.16.0.11 group PE
  peer 172.16.0.12 group PE
  peer 172.16.0.13 group PE
  peer 172.16.0.14 group PE
  peer 172.16.0.15  group PE
```

Verificar:

```
display bgp vpnv6 all peer / display bgp all summary
display bgp vpnv6 vpn-instance VRF-INTERNET routing-table
```

Laboratório

Agora vamos fechar cada PE com o refletor BGP01:

PE:

```
bgp 1
 undo default ipv4-unicast
 group RR internal
 peer RR connect-interface loopback0
 peer 172.16.0.1 group RR
 peer 172.16.0.1 description BGP01
 ipv4-family vpnv4
  peer RR enable
  peer RR advertise-community
  peer 172.16.0.1 group RR
```

Verificar:

```
display bgp vpnv6 all peer / display bgp all summary
display bgp vpnv6 vpn-instance VRF-INTERNET routing-table
```

Laboratório

Estabelecido o VPNv4, vamos criar as VRFs e redistribuir as rotas:

PE12-HULK e PE14-SPIDERMAN

```
ip vpn-instance VRF-SHIELD
 ipv4-family
  route-distinguisher 65001:3
  vpn-target 65001:3 export-extcommunity
  vpn-target 65001:3 import-extcommunity

ip vpn-instance VRF-HYDRA
 ipv4-family
  route-distinguisher 65001:4
  vpn-target 65001:4 export-extcommunity
  vpn-target 65001:4 import-extcommunity
```

```
bgp 1
 ipv4-family vpn-instance VRF-SHIELD
 import-route direct
 ipv4-family vpn-instance VRF-HYDRA
 import-route direct
```

Laboratório

Estabelecido o VPNv4, vamos criar as VRFs e redistribuir as rotas:

PE12-HULK

```
interface ethernet 1/0/3
 description SHIELD-1
 ip binding vpn-instance VRF-SHIELD
 ip address 10.12.3.1 24

interface ethernet 1/0/4
 description HYDRA-1
 ip binding vpn-instance VRF-HYDRA
 ip address 10.12.4.1 24
```

PE14-SPIDERMAN

```
interface ethernet 1/0/3
 description SHIELD-1
 ip binding vpn-instance VRF-SHIELD
 ip address 10.14.3.1 24

interface ethernet 1/0/4
 description HYDRA-1
 ip binding vpn-instance VRF-HYDRA
 ip address 10.14.4.1 24
```

Laboratório

Estabelecido o VPNv4, vamos criar as VRFs e redistribuir as rotas:

PE13-STRANGE

```
ip vpn-instance VRF-SHIELD
 ipv4-family
  route-distinguisher 65001:3
  vpn-target 65001:3 export-extcommunity
  vpn-target 65001:3 import-extcommunity

bgp 1
 ipv4-family vpn-instance VRF-SHIELD
 import-route direct

interface ethernet 1/0/4
 description SHIELD-2
 ip binding vpn-instance VRF-SHIELD
 ip address 10.13.3.1 24
```

Laboratório

Estabelecido o VPNv4, vamos criar as VRFs e redistribuir as rotas:

PE15-MORBIUS

```
ip vpn-instance VRF-HYDRA
 ipv4-family
  route-distinguisher 65001:4
  vpn-target 65001:4 export-extcommunity
  vpn-target 65001:4 import-extcommunity

bgp 1
 ipv4-family vpn-instance VRF-HYDRA
 import-route direct

interface ethernet 1/0/4
 description HYDRA-2
 ip binding vpn-instance VRF-HYDRA
 ip address 10.15.4.1 24
```

Laboratório

Conferindo:

PE's:
display bgp all summary
display bgp vpnv4 vpn-instance VRF-SHIELD routing-table
display bgp vpnv4 vpn-instance VRF-HYDRA routing-table

SHIELD-1:
/tool ping 10.14.3.2
/tool ping 10.15.3.2
/tool traceroute 10.14.3.2
/tool traceroute 10.15.3.2

HYDRA-1:
/tool ping 10.14.4.2
/tool ping 10.15.4.2
/tool traceroute 10.14.4.2
/tool traceroute 10.15.4.2

Laboratório

O isolamento realmente existe?!

SHIELD-1:
/tool ping 10.14.4.2 ????
/tool ping 10.15.4.2 ????

HYDRA-1:
/tool ping 10.14.3.2 ????
/tool ping 10.15.3.2 ????

Laboratório

O isolamento realmente existe?! Como pudemos ver, sim, ele existe!

Por estarem em VRF's diferentes, a S.H.I.E.L.D não "enxerga" as rotas da Hydra e vice-versa!

Só porque funciona...

...não quer dizer que não possa ser melhorado! A icônica frase da irmã do T'Challa também se aplica ao L3VPN! Isto porque podemos fazer muito mais, por exemplo:

- **Utilizar o "mode pipe" para não alterar o TTL na rede MPLS;**
- **Implementar o L3VPN Inter-AS na integração de redes;**
- **Manipular fluxos de pacotes no L3VPN por meio de túneis de engenharia de tráfego (MPLS-TE), usufruindo dos seus benefícios como Fast Reroute, reserva de banda, entre outros;**

É isso, não se limitem! Isso é apenas o início, portanto continuem estudando para se tornarem profissionais de excelência!

Meus **Contatos e links:**

E-mail: andredias@hexanetworks.com.br
Linkedin: https://www.linkedin.com/in/andreldias/

E-mail wallace@hexanetworks.com.br
Linkedin: www.linkedin.com/in/wallacemariadeandrade

Telefone: +55 (11) 4395-5806

Dúvidas ou Sugestões?