
Internet Service Provider ARA Project

UNIVERSIDADE DE AVEIRO

DIOGO SILVA 60337
EDUARDO 68633

Internet Service Provider ARA Project
Arquitectura de Redes Avançada
Universidade de Aveiro

Diogo Silva 60337 Eduardo Sousa 68633

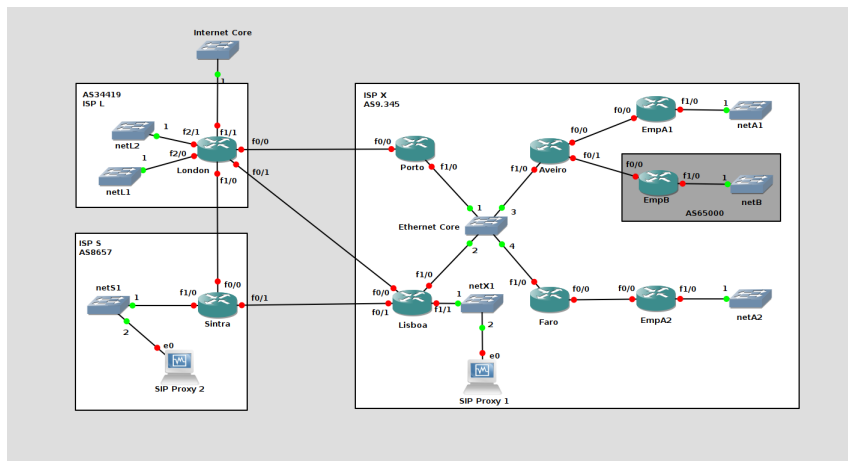
January 5, 2016

Contents

1	Basic Mechanisms and BGP	2
1.1	Internal BGP & OSPF Redistribution	2
1.2	External BGP	2
1.3	Private AS	2
1.4	Routing Constraints	2
1.4.1	Internet Traffic	3
1.4.2	Net L1 and Net L2 Preferences	3
1.4.3	SIP Proxy 2 Traffic	5
1.4.4	Non-Transit ISP-X	6
1.5	Changes for IPv6	7
2	MPLS	8
2.1	MPLS Tunnel for SIP Traffic	8
2.1.1	Validação do Túnel	11
2.2	MPLS VPN	12
2.2.1	Internal Connectivity	12
2.2.2	External Connectivity	13
3	VoIP SIP	15
3.1	Internal Extensions	15
3.2	PTSN Calls Support	16
3.3	Forward to SIP Proxy 2	17

Chapter 1

Basic Mechanisms and BGP



1.1 Internal BGP & OSPF Redistribution

#EDUARDO

1.2 External BGP

#EDUARDO

1.3 Private AS

#EDUARDO

1.4 Routing Constraints

Neste projecto todas as restrições de routing apresentadas a seguir foram efectuadas usando route-map para efectuar a respectiva regra, ou negar a

rota, ou aumentar a local preference da rede anunciada no iBGP.

1.4.1 Internet Traffic

“IP traffic towards Internet should be preferably routed via ISP S (Lisboa).”

Se a rota pertence à internet incrementa-se a preferência local (podia-se ter usado 0.0.0.0 para representar qualquer outra rede externa, ou seja, internet). No trecho de código seguinte podemos ver que se o ip da internet se verificar, coloca uma preferência local acima da default, caso não seja, anuncia a rota como veio.

```
1 access-list 5 permit 8.8.8.0 0.0.0.255
2
3 route-map INTERNET_LP permit 10
4   match ip address 5
5   set local-preference 200
6
7 route-map INTERNET_LP permit 20
```

Listing 1.1: Route-map para a Internet

Como se pretende dar mais preferência à ligação entre Sintra e Lisboa quando o tráfego vai para a internet, aplica-se o route-map a todas as rotas anunciadas por Sintra a Lisboa, sendo que se alguma dessas rotas anunciadas por Sintra pertencer a internet, a preferência local será aumentada.

```
1 router bgp 9.345
2   address-family ipv4
3   ...
4   neighbor 4.20.20.13 route-map INTERNET_LP in
```

Listing 1.2: Route-map da Internet no Neighbor Sintra no Router de Lisboa

1.4.2 Net L1 and Net L2 Preferences

“IP traffic towards netL1 and netL2, should be preferably routed via Porto from Aveiro, and via Lisboa from Faro.”

Definiu-se a seguinte route-map em Aveiro e Faro, tendo em conta que ambos querem aumentar a preferência para a route-map na netL1 e netL2, a única diferença é por onde querer ir (só muda onde é aplicada a route-map), então definiu-se a mesma para os dois.

```
1 access-list 10 permit 82.84.100.0 0.0.0.255
2 access-list 10 permit 82.84.200.0 0.0.0.255
3
4 route-map LNET_LP permit 25
5   match ip address 10
6   set local-preference 210
```

```
7 route-map LNET_LP permit 30
```

Listing 1.3: Route-map para a netL1 e netL2

Depois de definida a route-map, aplicou-se a rota ao neighbor respectivo. Se Aveiro receber uma rota anunciada pelo Porto que cumpra a route-map, aumenta-lhe a preferência. Em Faro caso receba uma rota anunciada por Lisboa que cumpra a route-map, aumenta-lhe a preferência local. Isso fez-se através do seguinte código.

```
1 neighbor 192.172.100.1 route-map LNET_LP in
```

Listing 1.4: Route-map LNET_LP no Neighbor Porto no Router de Aveiro

```
1 neighbor 192.172.100.2 route-map LNET_LP in
```

Listing 1.5: Route-map LNET_LP no Neighbor Lisboa no Router de Faro

BGP IP Route - Faro e Aveiro para a rede 82.84.X.0

Na imagem seguinte pode-se verificar que Aveiro tem uma preferencia local de 210 para a rede 82.84.100.0 e 82.84.100.0 por 192.172.100.1 (Porto) e Faro tem uma preferencia local de 210 por 192.172.100.2 (Lisboa), tal como previsto.

Faro					
Network	Next Hop	Metric	LocPrf	Weight	Path
*>18.8.8.0/24	192.172.100.2	0	200	0	8657 34419 1
* 180.172.100.128/26					
	192.172.100.3	0	100	0	?
*>	10.1.0.3	20		32768	?
* 180.172.100.192/26					
	192.172.100.3	20	100	0	?
*>	0.0.0.0	0		32768	?
*>181.84.100.0/24	192.172.100.3	0	100	0	65000 1
*>182.84.100.0/24	192.172.100.2	0	210	0	34419 1
* 1	192.172.100.1	0	100	0	34419 1
*>182.84.200.0/24	192.172.100.2	0	210	0	34419 1
* 1	192.172.100.1	0	100	0	34419 1
* i192.172.100.1/32	192.172.100.3	2	100	0	?
*>	10.1.0.1	2		32768	?
* i192.172.100.2/32	192.172.100.3	2	100	0	?
*>	10.1.0.2	2		32768	?
* i192.172.100.3/32	192.172.100.3	0	100	0	?
*>	10.1.0.3	2		32768	?
* i192.172.100.4/32	192.172.100.3	2	100	0	?
*>	0.0.0.0	0		32768	?
* i192.172.100.128/25					
	192.172.100.2	0	100	0	1
* 1	192.172.100.3	2	100	0	?
*>	10.1.0.2	2		32768	?
*>1200.1.100.0	192.172.100.2	0	100	0	8657 1
Aveiro					
Network	Next Hop	Metric	LocPrf	Weight	Path
*>18.8.8.0/24	192.172.100.2	0	200	0	8657 34419 1
* 80.172.100.128/26					
	0.0.0.0	0		32768	?
* 180.172.100.192/26					
	192.172.100.4	0	100	0	?
*>	10.1.0.4	20		32768	?
*> 81.84.100.0/24	10.1.100.10	0		0	65000 1
* 182.84.100.0/24	192.172.100.2	0	100	0	34419 1
*>1	192.172.100.1	0	210	0	34419 1
* 182.84.200.0/24	192.172.100.2	0	100	0	34419 1
*>1	192.172.100.1	0	210	0	34419 1
* i192.172.100.1/32	10.1.0.1	2	100	0	?
*>	10.1.0.1	2		32768	?
* i192.172.100.2/32	10.1.0.2	2	100	0	?
*>	10.1.0.2	2		32768	?
*> 192.172.100.3/32	0.0.0.0	0		32768	?
* i192.172.100.4/32	192.172.100.4	0	100	0	?
*>	10.1.0.4	2		32768	?
* i192.172.100.128/25					
	10.1.0.2	2	100	0	?
* 1	192.172.100.2	0	100	0	1
*>	10.1.0.2	2		32768	?
*>1200.1.100.0	192.172.100.2	0	100	0	8657 1
Aveiro#					

Figure 1.1: BGP IP Route - Faro e Aveiro (Local Preferences para a rede 82.84.x.0)

1.4.3 SIP Proxy 2 Traffic

“IP traffic for remote SIP proxy 2 (to network netS1) should be routed only via Lisboa using the direct peering link to ISP S.”

Para fazer com que Lisboa -> Sintra fosse o único peering possível do ISP X para a NetS1, optou-se pela estratégia oposta, negar todas as saídas possíveis, ou seja, tudo o que anunciar a NetS1 e que não seja aquele link não é aceite. Sendo definida a seguinte rota:

```
1 access-list 6 permit 200.1.100.0 0.0.0.255
2
3 route-map SIP_ROUTE deny 11
4   match ip address 6
5 route-map SIP_ROUTE permit 21
```

Listing 1.6: Route-map SIP_ROUTE para cancelar rotas

Sendo que foi preciso definir nos neighbors as rotas, neste caso, em Lisboa e Porto, rotas para a NetS1 recebidas pelo link de London.

```
1 neighbor 4.20.20.5 route-map SIP_ROUTE in
```

Listing 1.7: Cancelar rota para NetS1 recebida em Lisboa por London

```
1 neighbor 4.20.20.1 route-map SIP_ROUTE in
```

Listing 1.8: Cancelar rota para NetS1 recebida no Porto por London

1.4.4 Non-Transit ISP-X

Para efectuar o Non-Transit foi preciso definir uma verificação no as-path, se o AS-PATH for vazio, ou contiver apenas o AS 65000 (privado) significa que é uma rota interna e pode ser anunciada para que os outros saibam que aquele neighbor pode ser usado para chegar a rede anunciada, sendo que redes que contenham outros AS-PATH (Sistemas autonomos externos não serão anunciados para fora).

Teve-se de verificar para o AS 65000 porque ele atravessa pelo AS 9.345 para sair para London ou Sintra, e o rm-private-as é apenas retirado depois da validação das route-maps definidas no neighbor.

Sendo que se usou a seguinte route-map em todas as saídas possíveis do AS 9.345:

```
1 ip as-path access-list 4 permit ^$
2 ip as-path access-list 4 permit ^65000$
3 route-map NON_TRANSIT permit 10
4   match as-path 4
```

Listing 1.9: Tornar ISP X num AS Non-Transit

Sendo depois aplica-se esta route-map em Lisboa e no Porto em todos os neighbors ao anunciar rotas para fora (neighbor VIZINHO route-map NON_TRANSIT out)

1.5 Changes for IPv6

#EDUARDO

Chapter 2

MPLS

2.1 MPLS Tunnel for SIP Traffic

Para o SIP Traffic é importante a existência de Tuncis entre as empresas e o SIP Proxy, sendo que definiu-se que o Tunnel em Aveiro até Lisboa seria suficiente, porque as ligações de Aveiro a empA1 é uma ligação ponto-a-ponto e Aveiro é que pertence ao Core do ISP (também podemos verificar que o limite do iBGP só chega até Aveiro, a empA1 já não tem BGP), daí termos escolhido Aveiro como um dos extremos do túnel, o mesmo se verifica para Faro.

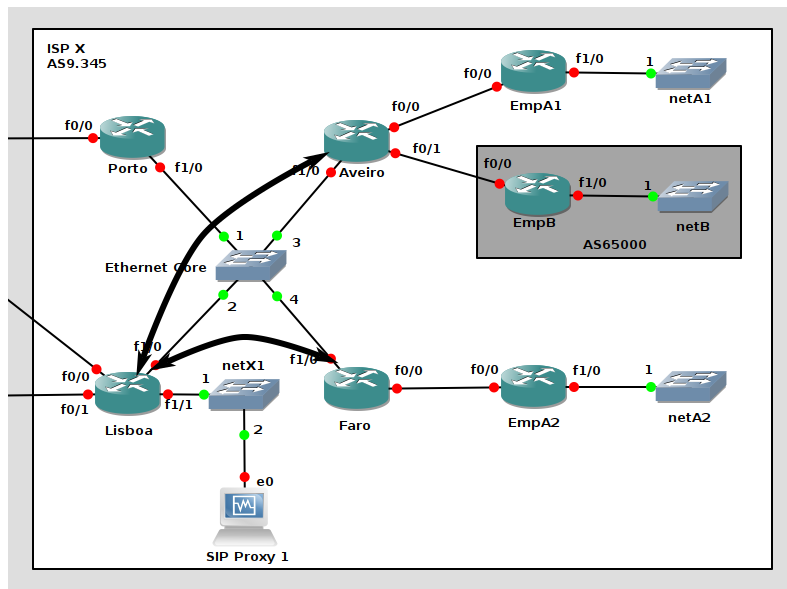


Figure 2.1: MPLS Tunnel entre SIP Proxy 1 e Aveiro (empA1 e EmpB) e o SIP Proxy e Faro (empA2)

Sendo assim criou-se dois túneis, um túnel (1) para fazer a transição do tráfico da Empresa A1 (ramo de Aveiro) e da Empresa B até ao SIP Proxy 1 e vice-versa. Depois criou-se outro túnel para o ramo da Empresa A em Faro (empA2) até ao SIP Proxy 1. Tal como mostra a imagem anterior.

De forma a activar o MPLS com RSVP-TE sobre a rede entre Aveiro/Faro e Lisboa foi preciso activar na configuração OSPF de forma a ele se propagar (Loopbacks e OSPF já estavam definidos anteriormente, foi só activar o MPLS traffic engineering), usando o seguinte comando (em todas as interfaces físicas e na configuração geral do router, isto entre Aveiro/Faro e Lisboa):

```
1 mpls traffic-eng area 0
2 mpls traffic-eng router-id Loopback 0
```

Listing 2.1: MPLS Tunnel - Activar MPLS TE no OSPF

Para activar o RSVP ainda é preciso definir a largura de banda do RSVP, tendo em conta que ambos os túneis vão usar o mesmo RSVP e pretende-se 1Mbits em cada túnel MPLS, é preciso definir no mínimo 2Mbits na largura de banda RSVP para ter os dois túneis activos ao mesmo tempo, no entanto definiu-se 8Mbits para se ter uma pequena margem. Usando assim o seguinte comando nas interfaces físicas.

```
1 ip rsvp bandwidth 8096 8096
```

Listing 2.2: MPLS Tunnel - Activar RSVP

Para configurar o Túnel em Aveiro usou-se a seguinte sequência de comandos:

```
1 interface Tunnel1
2 ip unnumbered Loopback0
3 tunnel mode mpls traffic-eng
4 tunnel destination 192.172.100.2
5 tunnel mpls traffic-eng bandwidth 1024
6 tunnel mpls traffic-eng path-option 1 dynamic
```

Listing 2.3: MPLS Tunnel - Aveiro

Sendo que em Lisboa foi definido um Túnel para Aveiro também com a respectiva interface de Loopback de Aveiro, o mesmo foi efectuado para Faro.

Depois de ter definido todas as interfaces e activar correctamente o MPLS com RSVP, é preciso indicar como é que é possível entrar no túnel. Sendo que foi definido o seguinte código no router de Lisboa para permitir a entrada no túnel:

```

1 interface FastEthernet1/1      !interface do SIP Proxy 1
2 ip policy route-map SIP_PORT
3
4 route-map SIP_PORT permit 12          !Aveiro
5 match ip address VoIP-TRAFFIC 7      !SIP e ACL7
6 set interface Tunnel1
7
8 route-map SIP_PORT permit 13          !Faro
9 match ip address VoIP-TRAFFIC 8      !SIP e ACL 8
10 set interface Tunnel2
11
12 access-list 7 permit 10.1.1.0 0.0.0.127 !EmpA1
13 access-list 7 permit 81.84.100.0 0.0.0.255 !EmpB
14 access-list 8 permit 10.1.1.128 0.0.0.127 !EmpA2
15
16 ip access-list extended VoIP-TRAFFIC
17 permit udp any any range 16384 32767
18 permit udp any any range 16384 32767 any
19 permit udp any any eq 5060
20 permit tcp any any eq 5060
21 permit udp any eq 5060 any
22 permit tcp any eq 5060 any
23 permit udp any any eq 5061
24 permit tcp any any eq 5061
25 permit udp any eq 5061 any
26 permit tcp any eq 5061 any

```

Listing 2.4: MPLS Tunnel - Lisboa rota de entrada no túnel

Este route-map é aplicado na interface que está do lado do SIP Proxy 1, ou seja, só o tráfego que vem daquela interface com destino as redes das empresas e com um porto de tráfego VoIP é que consegue atravessar por dentro do túnel, caso contrário, consegue comunicar mas não por dentro do túnel (acabando por perder os seus benefícios).

Do lado de Aveiro/Faro foi preciso fazer uma configuração similar para entrar no túnel, no entanto não se verificou o IP destino, desde que seja SIP entra pelo túnel:

```

1 interface FastEthernet0/0          !Empresa A1
2 ip vrf forwarding VPN-ClientA
3 ip address 10.1.100.1 255.255.255.252
4 ip policy route-map SIP_PORT
5
6 interface FastEthernet0/1          !Empresa B
7 ip address 10.1.100.9 255.255.255.252
8 ip policy route-map SIP_PORT
9
10 route-map SIP_PORT permit 11
11 match ip address VoIP-TRAFFIC
12 set interface Tunnel1

```

Listing 2.5: MPLS Tunnel - Aveiro rota de entrada no túnel

Em Faro foi feita uma configuração similar a de Aveiro, sendo isto o necessário para ter ambos os túneis a funcionar correctamente e em simultâneo.

2.1.1 Validação do Túnel

A validação deste túnel não pode ser feita através de GNS3 (pelo menos totalmente) porque não se via nenhum label MPLS o que está correcto, Lisboa é o penultimo router do Tunnel para Aveiro, logo é suposto ser o último pop do label.

Para confirmar que de facto estava a entrar dentro do tunnel usou o comando: `show interfaces Tunnel 1`

E verificou-se o estado do Túnel como também o número de pacotes que tinham lá passado, após verificar o número de pacotes efectuou-se um ping com um VPCS (são os unicos que permitem pings usando UDP e o porto) no porto 5060 em UDP e verificou-se que ao efectuar show interfaces Tunnel 1 novamente, o número de pacotes que tinha atravessado o túnel tinha sido 5.

A imagem seguinte mostra a validação efectuada, sendo 93 o número de pacotes inicial e 98 após o envio de pacotes UDP para a network da empresa A.

```

Tunnel receive bandwidth 8000 (Kbps)
Last input never, output 00:06:29, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/0 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts (0 IP multicasts)
0 runs, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
93 packets output, 5126 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 unknown protocol drops
0 output buffer failures, 0 output buffers swapped out
Liisboa(config-if)#
Liisboa(config-if)#
Liisboa(config-if)#do show interfaces Tunnel 1
Tunnel1 is up, line protocol is up
Hardware is Tunnel
Interface is unnumbered. Using address of Loopback0 (192.172.100.2)
MTU 17940 bytes, BW 100 Kbit/sec, DLY 500000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 192.172.100.2, destination 10.1.0.3
Tunnel protocol/transport Label Switching
Tunnel transmit bandwidth 8000 (Kbps)
Tunnel receive bandwidth 8000 (Kbps)
Last input never, output 00:07:31, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/0 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts (0 IP multicasts)
0 runs, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
90 packets output, 5511 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 unknown protocol drops
0 output buffer failures, 0 output buffers swapped out
Liisboa(config-if)#
Liisboa(config-if)#
```

Figure 2.2: MPLS Tunnel - Teste de VPCS em Lisboa para interface da EmpA1 (Pacote UDP)

2.2 MPLS VPN

2.2.1 Internal Connectivity

Para configurar a VPN é preciso criar uma VRF que é o que vai permitir ter um encaminhamento específico para a network da empresa A, neste caso vai ter um routing table específica para a rede da empresa A, sendo esse o panel da VRF (Routing e Forwarding).

Para além disso cada VRF tem de ter um identificador que a permite distinguir de outras VRF existentes (neste caso só existe a da empresa A), e é preciso indicar que todas as rotas desta VRF serão exportadas por VRF com o mesmo identificador e também irá importar rotas de VRFs com o mesmo identificador. Sendo assim definida a seguinte VRF:

```
1 ip vrf VPN-ClientA
2   rd 9345:1
3   route-target export 9345:1
4   route-target import 9345:1
```

Listing 2.6: VPN - Criar uma VRF e associar um router distinguisher

Depois de definir a VRF é preciso indicar a interface a qual a VRF estará associada, neste caso será na interface FastEthernet0/0 em Aveiro e FastEthernet0/0 em Faro sendo a outra ponta.

```
1 interface FastEthernet0/0
2   ip vrf forwarding VPN-ClientA
```

Listing 2.7: VPN - Associar a VRF a uma interface

Para além disso é preciso identificar aos Provider Edge (PE), neste caso, Aveiro e Faro, quais são os pacotes que pertencem as VPNs, para isso efectua-se o seguinte comando.

```
1 router bgp 9.345
2   address-family vpnv4
3     neighbor 192.172.100.4 activate
4     neighbor 192.172.100.4 send-community both
5   address-family ipv4 vrf VPN-ClientA
6     redistribute connected
```

Listing 2.8: VPN - Anunciar os labels das VPNs

Após efectuar este pequeno comando reparou-se que apenas se tinha ligação entre a rede directamente ligada a Aveiro e a rede directamente ligada a Faro, com a VPN a funcionar apesar de tudo. Isto deve-se a VRF estar completamente vazia apenas com as directamente ligadas!

O que se fez para resolver este problema e ter conectividade não só com a rede directamente ligada a Aveiro mas também com a rede da Empresa A1 e da Empresa A2 foi activar um processo OSPF na VRF como mostra a imagem seguinte rodeado com círculos pretos:

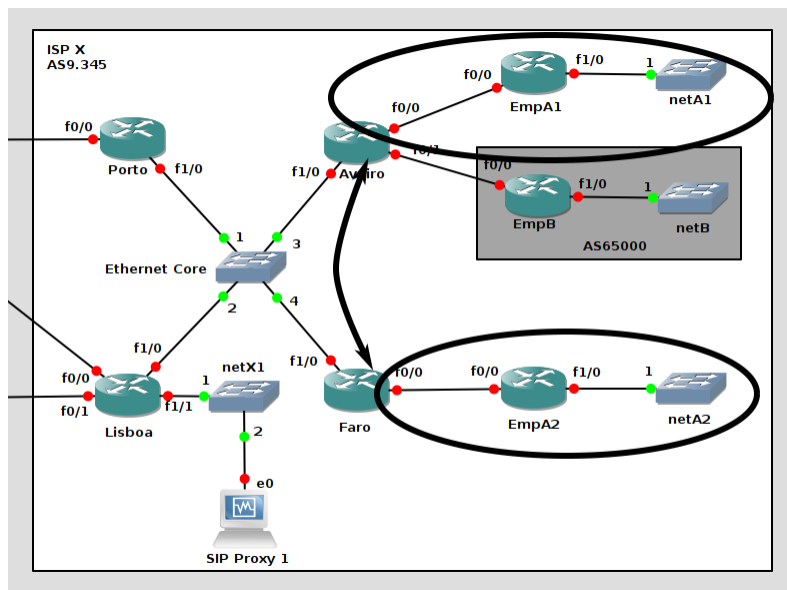


Figure 2.3: MPLS VPN entre Aveiro e Faro (empresas A1 e A2)

Para activar o OSPF na VRF foi preciso executar os seguintes comandos:

```
1 router ospf 200 vrf VPN-ClientA
2 network 10.1.0.0 0.0.255.255 area 0
```

Listing 2.9: VPN - Suporte de OSPF entre o PE e o CE

Depois de activar OSPF na VRF do router de Aveiro, Faro, EmpA1 e EmpA2 passou-se a ter total conectividade dentro da VPN.

2.2.2 External Connectivity

Para obter conectividade externa é preciso indicar a rede da VPN como sair para fora da rede interna, neste caso faz-se com uma rota estática (que é distribuída pelo OSPF da VRF) definida na VRF indicando a interface de saída.

```
1 ip route vrf VPN-ClientA 0.0.0.0 0.0.0.0 192.172.100.1 global
```

Listing 2.10: VPN - Rota de saída da VPN-ClientA

E é preciso indicar às redes de fora (ou seja, a todo o iBGP e eBGP) quais são as redes que estão disponíveis dentro da VRF.

```
1 ip route 10.1.1.0 255.255.255.128 FastEthernet0/0
2 ip route 80.172.100.128 255.255.255.192 FastEthernet0/0
```

Listing 2.11: VPN - Anunciar rotas disponíveis dentro da VPN do lado de Aveiro

Sendo estas rotas distribuídas para o OSPF para funcionar na rede de interna, e para o BGP para funcionar externamente de igual forma, existindo assim tanto conectividade dentro da VPN como de fora para dentro e vice-versa.

Chapter 3

VoIP SIP

Neste projeto foi pedido que existisse um serviço de SIP-VoIP para todos os clientes empresariais do ISP X. Esse serviço VoIP deve fornecer conectividade entre todos os clientes empresariais, bem como fornecer compatibilidade com chamadas oriundas de PTSN, encaminhadas pelo SIP Proxy 2. O serviço de VoIP deve também encaminhar as chamadas para redes externas utilizando o SIP Proxy 2.

3.1 Internal Extensions

A configuração dos clientes é a seguinte:

```
1 [EmpA1]
2 type=friend
3 host=dynamic
4 secret=labcom
5 context=phones
6 allow=all
7
8 [EmpA2]
9 type=friend
10 host=dynamic
11 secret=labcom
12 context=phones
13 allow=all
14
15 [EmpB]
16 type=friend
17 host=dynamic
18 secret=labcom
19 context=phones
20 allow=all
```

Listing 3.1: SIP Proxy 1 - /etc/asterisk/sip.conf

Apenas existe uma conta por empresa, para ser possível testar a conectividade entre elas. Na realidade poderiam existir muitas mais sem que existem

problemas.

De seguida configurou-se a extensões dos clientes, sendo elas:

```
1 [phones]
2 exten => 1000,1,Answer(500)
3 exten => 1000,2,PlayBack(demo-congrats)
4 exten => 1000,3,PlayBack(vm-goodbye)
5 exten => 1000,4,HangUp()
6
7 exten => 2000,1,Dial(SIP/EmpA1)
8 exten => 2000,2,voicemail(3000@corporations_voicemail)
9 exten => 2000,3,PlayBack(vm-goodbye)
10 exten => 2000,4,HangUp()
11
12 exten => 2001,1,Dial(SIP/EmpA2)
13 exten => 2001,2,voicemail(3001@corporations_voicemail)
14 exten => 2001,3,PlayBack(vm-goodbye)
15 exten => 2001,4,HangUp()
16
17 exten => 2002,1,Dial(SIP/EmpB)
18 exten => 2002,2,voicemail(3002@corporations_voicemail)
19 exten => 2002,3,PlayBack(vm-goodbye)
20 exten => 2002,4,HangUp()
21
22 exten => 3000,1,voicemail(3000@corporations_voicemail)
23 exten => 3001,1,voicemail(3001@corporations_voicemail)
24 exten => 3002,1,voicemail(3002@corporations_voicemail)
```

Listing 3.2: SIP Proxy 1 - /etc/asterisk/extensions.conf

A extensão 1000 serve para testar a conectividade com o servidor. A extensão 2000 é a extensão do polo 1 da empresa A. A extensão 2001 é a extensão do polo 2 da empresa A. A extensão 2002 é a extensão da empresa B. As extensões 3000, 3001 e 3002 dão acesso ao voicemail das empresas. Todas as extensões 2000, 2001 e 2002 caso estejam ocupadas vão parar ao voicemail.

As configurações do voicemail são as seguintes:

```
1 [corporations_voicemail]
2 3000 => 1212,EmpA1
3 3001 => 1212,EmpA2
4 3002 => 1212,EmpB
```

Listing 3.3: SIP Proxy 1 - /etc/asterisk/voicemail.conf

3.2 PTSN Calls Support

As chamadas provenientes da rede PTSN vão utilizar a seguinte configuração para serem reencaminhadas para as respetivas empresas.

```
1 [phones]
```

```

2 ...
3 exten => _2341000XX,1,Dial(SIP/EmpA1)
4 exten => _2341000XX,2,voicemail(3000@corporations_voicemail)
5 exten => _2341000XX,3,PlayBack(vm-goodbye)
6 exten => _2341000XX,4,HangUp()
7
8 exten => _2891001XX,1,Dial(SIP/EmpA2)
9 exten => _2891001XX,2,voicemail(3001@corporations_voicemail)
10 exten => _2891001XX,3,PlayBack(vm-goodbye)
11 exten => _2891001XX,4,HangUp()
12
13 exten => _2341002XX,1,Dial(SIP/EmpB)
14 exten => _2341002XX,2,voicemail(3002@corporations_voicemail)
15 exten => _2341002XX,3,PlayBack(vm-goodbye)
16 exten => _2341002XX,4,HangUp()

```

Listing 3.4: SIP Proxy 1 - /etc/asterisk/extensions.conf

O underscore indica o início do padrão que queremos utilizar e o X indica números de 0 a 9. De resto a configuração é semelhante a anterior.

3.3 Forward to SIP Proxy 2

Para fazer reencaminhar as chamadas VoIP para redes externas é preciso alterar as configurações dos dois servidores SIP.

```

1 ...
2 [Server2]
3 type=peer
4 host=200.1.100.2
5 secret=labcom
6 username=Server1

```

Listing 3.5: SIP Proxy 1 - /etc/asterisk/sip.conf

```

1 [Server1]
2 type=peer
3 host=192.172.100.130
4 secret=labcom
5 context=phones

```

Listing 3.6: SIP Proxy 2 - /etc/asterisk/sip.conf

```

1 ...
2 exten => _X!,1,Dial(SIP/${EXTEN}@Server2,10)
3 exten => _X!,2,HangUp()

```

Listing 3.7: SIP Proxy 1 - /etc/asterisk/extensions.conf

```

1 [phones]
2 exten => _X!,1,Answer(500)
3 exten => _X!,2,PlayBack(demo-congrats)
4 exten => _X!,3,PlayBack(vm-goodbye)

```

```
5 exten => _X! ,4 ,HangUp()
```

Listing 3.8: SIP Proxy 2 - /etc/asterisk/extensions.conf

No servidor 1 foi preciso criar um peer para permitir que o mesmo se consiga ligar ao servidor 2. Foi também preciso criar uma nova regra nas extensões do servidor 1 que reencaminhe todas as chamadas VoIP com destino em extensões externas para o servidor 2, assim sendo foi criado o padrão _X!, onde o underscore indica o início do padrão, o X indica qualquer número e o ! todas as extensões não definidas anteriormente. Caso se encontre alguma chamada para uma extensão deste tipo, o servidor 1 deve fazer uma SIP URI dial para o servidor 2 para onde envia a extensão.

No servidor 2 foi preciso criar um peer para indicar que o servidor 1 se vai ligar a ele. As extensões são idênticas às definidas anteriormente.

Listings

1.1	Route-map para a Internet	3
1.2	Route-map da Internet no Neighbor Sintra no Router de Lisboa	3
1.3	Route-map para a netL1 e netL2	3
1.4	Route-map LNET_LP no Neighbor Porto no Router de Aveiro	4
1.5	Route-map LNET_LP no Neighbor Lisboa no Router de Faro	4
1.6	Route-map SIP_ROUTE para cancelar rotas	6
1.7	Cancelar rota para NetS1 recebida em Lisboa por London	6
1.8	Cancelar rota para NetS1 recebida no Porto por London	6
1.9	Tornar ISP X num AS Non-Transit	6
2.1	MPLS Tunnel - Activar MPLS TE no OSPF	9
2.2	MPLS Tunnel - Activar RSVP	9
2.3	MPLS Tunnel - Aveiro	9
2.4	MPLS Tunnel - Lisboa rota de entrada no túnel	10
2.5	MPLS Tunnel - Aveiro rota de entrada no túnel	10
2.6	VPN - Criar uma VRF e associar um router distinguisher	12
2.7	VPN - Associar a VRF a uma interface	12
2.8	VPN - Anunciar os labels das VPNs	12
2.9	VPN - Suporte de OSPF entre o PE e o CE	13
2.10	VPN - Rota de saída da VPN-ClientA	13
2.11	VPN - Anunciar rotas disponiveis dentro da VPN do lado de Aveiro	14
3.1	SIP Proxy 1 - /etc/asterisk/sip.conf	15
3.2	SIP Proxy 1 - /etc/asterisk/extensions.conf	16
3.3	SIP Proxy 1 - /etc/asterisk/voicemail.conf	16
3.4	SIP Proxy 1 - /etc/asterisk/extensions.conf	16
3.5	SIP Proxy 1 - /etc/asterisk/sip.conf	17
3.6	SIP Proxy 2 - /etc/asterisk/sip.conf	17
3.7	SIP Proxy 1 - /etc/asterisk/extensions.conf	17
3.8	SIP Proxy 2 - /etc/asterisk/extensions.conf	17