Practical Work Report

Arquitectura de Redes Avançada
2018/19

António Silva 76678
Filipe Grilo 76528

Introduction

In this report we explain our engineering decisions we took in order to meet the requirements defined for the development of the Practical Work. We try to take into account what would be an ISP best practice for optimization and simplicity.

Distribution of IPv4 addresses

ISP PT2		LoopBacks	
Core Network		10.3.1.1	Porto
10.0.0.0/30	Porto-Lisboa2	10.3.1.2	Lisboa2
10.0.0.4/30	Lisboa2-Aveiro	10.3.1.3	Aveiro
10.0.0.12/30	Porto-Aveiro	10.3.1.4	Oeiras
10.0.0.8/30	Lisboa<->Oeiras	10.3.1.5	EmpA
CONNECTION TO CLIENTS		10.3.1.6	EmpB
10.0.1.0/30	Aveiro<->EmpA1	**External BGP**	
10.0.1.4/30	Aveiro<->EmpB1	4.4.4.0/30	PORTO<->MADRID
10.0.2.192/30	Oeiras<->EmpA1	4.4.4.8/30	LISBOA2 <-> MADRID
10.0.2.196/30	Oeiras<->EmpB1	4.4.4.4/30	LISBOA2 <-> LISBOA1
SERVICES		4.4.4.12/30	LISBOA1 <-> MADRID
10.2.0.0/17	Lisboa<->DataCenter		
10.2.128.0/17	Aveiro<->DataCenter		
2.2.2.0/24	Porto <-> SIP Proxy 1		
3.3.3.0/24	Oeiras <-> BackUpSIP		

Decisions on deployment requirements

- Basic mechanisms and BGP

In order to establish connection between all the routers in ISP PT2, an OSPF process was activated on the interfaces that connect them.

BGP was used to route the traffic between, and inside the Autonomous Systems. More specifically, eBGP was used between Porto, Lisboa 1-2 and Madrid to connect the three Autonomous Systems and iBGP was used between the four stations inside ISP PT2 (Porto, Lisboa 2, Aveiro and Oeiras) that allows the exchange of external routes to the whole ISP PT2.

The following task where made to fill this requirement:

- Since they provide connectivity to the internet, Lisboa 1 and Madrid announce the default route through their neighbour relations with ISP PT2.

```
neighbor 4.4.4.1 default-originate
```

- The routes learned in the OSPF process that is running between the routers in ISP PT2 is being redistributed to the BGP.

```
redistribute ospf 1
```

- The exchange of private networks, default routes and external routes (since ISP PT2 in non-transit) on eBGP were filtered using a route-map.

```
! IP PREFIX LISTS
ip prefix-list ips-privados seq 10 permit 10.0.0.0/8
le 32
ip prefix-list ips-privados seq 12 permit
172.16.0.0/12 le 32
ip prefix-list ips-privados seq 14 permit
192.168.0.0/16 le 32
ip prefix-list ips=privados seq 16 permit 0.0.0.0/0


! Only allows as-path empty a.k.a local routes
ip as-path access-list 1 permit ^$
```

```
(...)
! ROUTE MAPS
!! DENY PRIVATE IPS AND PERMIT LOCAL ROUTES
route-map rm-ips-privados deny 10
match ip address prefix-list ips-privados
route-map rm-ips-privados permit 20
match as-path 1

neighbor 4.4.4.6 route-map rm-ips-privados out
```

Requirements:

IP traffic towards Internet should be preferably routed via ISP PT1:

- This was done by simply increasing Local Preference of the default information generated by ISP PT1

```
! default IP prefix-list
ip prefix-list default-gt seq 10 permit 0.0.0.0/0

! Increase local preference of AS40000 for traffic
to internet
route-map increase-40000 permit 10
match ip address prefix-list default-gt
set local-preference 200

!! LISBOA 1 (EGP)
neighbor 4.4.4.6 remote-as 40000
neighbor 4.4.4.6 route-map increase-40000 in
```

IP traffic towards all AS20000 networks, should be preferably routed via Porto from Aveiro, and via Lisboa2 from Oeiras:

- First, we decrease Local Preference of routes towards AS20000 on Lisboa2 and apply it to Porto and Aveiro.

```
ip as-path access-list 2 permit _20000$
```

```
route-map lower-as20000 permit 10
match as-path 2
set local-preference 50
route-map lower-as20000 permit 20
!! PORTO (IGP)
neighbor 10.3.1.1 remote-as 1000
neighbor 10.3.1.1 route-map lower-as20000 out

!! AVEIRO (IGP)
neighbor 10.3.1.3 remote-as 1000
neighbor 10.3.1.3 route-map lower-as20000 out
```

- But since Oeiras was still receiving routes from Lisboa and Porto with same Local Preference, we increased Local Preference of routes towards AS20000 on Lisboa2 and apply it to Oeiras.

```
ip as-path access-list 2 permit _20000$

route-map increase-as20000 permit 10
match as-path 2
set local-preference 200
route-map increase-as20000 permit 20

!! OEIRAS (IGP)
neighbor 10.3.1.4 remote-as 1000
neighbor 10.3.1.1 route-map increase-as20000 out
```

IP traffic for remote SIP proxy 2 (to network netL1) cannot be routed via Porto using the direct peering link to ISP ES:

- Done by denying routes to SIP proxy 2 (network netL1) in Porto

```
ip prefix-list rm-sipProxy-2 seq 10 deny
65.0.1.0/24 le 32
ip prefix-list rm-sipProxy-2 seq 20 permit
0.0.0.0/0 le 32

!! MADRID (EGP)
neighbor 4.4.4.2 remote-as 20000
neighbor 4.4.4.2 prefix-list rm-sipProxy-2 in
```

- MPLS

In order to fulfill the client's requests. MPLS-TE and RSVP-TE was activated on the core network of ISP PT2.

```
! Enable MPLS
ip cef
mpls traffic-eng tunnels
mpls ip

! Channel to PORTO
int atm3/0.1 point-to-point
mpls traffic-eng tunnels
mpls ip
ip rsvp bandwidth 102400 102400
```

Requirements:

Client B requested two bi-directional channels, between its two branches, with dedicated bandwidth of 30Mbps:

- Create two tunnels in Aveiro and Oeiras that connect with each other and take a reservation of 30Mbps

```
int Tunnel 20
ip unnumbered lo0
tunnel destination 10.3.1.4
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng bandwidth 30720
tunnel mpls traffic-eng path-option 1 dynamic
```

- Create Route-Maps that forwards client's B traffic to the tunnels

```
! PBR for EMPRESA B tunnels
access-list 100 permit ip 111.1.1.0 0.0.0.63
111.1.1.128 0.0.0.127
access-list 101 permit ip 111.1.1.64 0.0.0.63
111.1.1.192 0.0.0.127

route-map empresaB-tunnels permit 10
match ip address 100
set interface tunnel 20
```

```
route-map empresaB-tunnels permit 20
match ip address 101
set interface tunnel 21
```

Deploy a MPLS VPN for Corporate client A (interconnecting Aveiro and Oeiras branches):

- Create VRF for VPN to be used by client's A

```
ip vrf VPN-10
rd 1000:1
route-target export 1000:1
route-target import 1000:1
```

- Define interface to be used by VPN

```
router ospf 10 vrf VPN-10
! EmpA
int f1/0
no shut
ip vrf forwarding VPN-10
```

- Create static route for VPN's default gateway and another on to let the network know about the existence of the network on the VPN.

```
! Default GateWay for VPN10 (PORTO)
ip route vrf VPN-10 0.0.0.0 0.0.0.0 10.3.1.1 global

! Let Them Know About The Network on the VPN
ip route 110.1.1.0 255.255.255.128 f1/0

! Distribute static routes
router bgp 1000
redistribute static ip
```

- CDN

A CDN server is available at ISP PT2. This CDN has the main purpose of routing the clients to the closest datacenter based on their IP addresses.

As it was defined, clients in Aveiro should be routed to Aveiro Datacenter and all others to Lisboa Datacenter.

Requirements:

Deploy a CDN routing service (Conditional DNS) for corporate clients. The service DNS server is located in the Lisboa Datacenter, and must be able to redirect clients to the closest Datacenter according to their location:

- First, we start by creating ACL for each location possible:

```
acl "Aveiro" {
        111.1.1.0/25;
        10.2.128.0/17;
};

acl "EmpresaB" {
        111.1.1.0/24;
};

acl "EmpresaAAveiro" {
        110.1.1.0/25;
};

acl "EmpresaAOeiras" {
        110.1.1.128/25;
};
```

For this project, we divided the addresses between “Aveiro”, “EmpresaB”, “EmpresaAAveiro” and “EmpresaAOeiras”.

The division of Empresa A between Aveiro in Oeiras was needed to fill a requirement in VoIP module (explained on next chapter).

- Afterwards, we define what zone to return to the client based on the ACL he matches on.

```
view "aveiro" {
        match-clients { Aveiro; };
        recursion no;
        zone "cdn.pt" {
                type master;
                file "/etc/bind/cdn.pt-aveiro.db";
        };
};
```

In this example, we return “cdn.pt-aveiro.db” we a client in Aveiro requests “cdn.pt”.

- Lastly, we create the zones needed, for example, for aveiro.

```
$ORIGIN cdn.pt.
@         IN      SOA     ns1.cdn.pt.     adm.cdn.pt.     (
                          2          ; Serial
                     604800          ; Refresh
                      86400          ; Retry
                    2419200          ; Expire
                     604800 )        ; Negative Cache TTL
          IN      NS      ns1.cdn.pt.
          IN      A       10.2.128.2
ns1       IN      A       10.2.0.2
~
~
~
```

Improve the CDN routing service (Conditional DNS) by including a rotating decision for requests associated to Corporate B:

- First of all, it is needed to create an ACL to identity IPs coming from Empresa B.

```
acl "EmpresaB" {
        111.1.1.0/24;
};
```

- A rotating decision can be easily done by simply defining two A addresses in the zone file. Automatically, the decision is rotated.

```
$ORIGIN cdn.pt.
@         IN      SOA     ns1.cdn.pt.     adm.cdn.pt.     (
                          2          ; Serial
                     604800          ; Refresh
                      86400          ; Retry
                    2419200          ; Expire
                     604800 )        ; Negative Cache TTL
          IN      NS      ns1.cdn.pt.
          IN      A       10.2.128.2
          IN      A       10.2.0.2
ns1       IN      A       10.2.0.2
~
```

- VoIP - SIP

Three SIP Proxies were added to the project in order to provide VoIP service to clients. SIP Proxy 1 is located in ISP PT2 and is used as the primary proxy for ISP PT2 clients. SIP Proxy 2 was added in ISP PT1 mainly to forward PSTN phone numbers. Lastly, a backup proxy was added in ISP PT2 which is pretty much a replica of Proxy 1.

In order to test the configurations, there are 2 clients available in ISP PT2.

Asterisk was used for the development of this servers.

Requirements:

Deploy a VoIP - SIP service for all ISP PT2 corporate clients. The service provides VoIP connectivity (through ISP SIP Proxy 1) between internal clients and forwards all other calls (including PSTN numbers) to ISP PT1 SIP Proxy 2:

- First we start by adding the users so that a call can be establish.

```
[ManelZe]
type=friend
host=dynamic
secret=labcom
context=phones
allow=all
mailbox=2001@ara_voicemail

[ZeManel]
type=friend
host=dynamic
secret=labcom
context=phones
allow=all
mailbox=2002@ara_voicemail

[Proxy2]
type=peer
secret=labcom
username=Proxy1
host=65.0.1.100
```

"ManelZe" and "ZeManel" are two typical users with a voicemail available at extension 2001@ara_voicemail.

In order forward calls to SIP Proxy2, it is needed to define it as a peer and describe the username/secret and host to be used.

- With the users defined, we simply need to describe how the server will respond to each extension.

```
exten => 2001,1,Dial(SIP/ZeManel,10)
exten => 2001,2,VoiceMail(2001@ara_voicemail)
exten => 2001,3,PlayBack(vm-goodbye)
exten => 2001,4,HangUp()

exten => 2002,1,Dial(SIP/ManelZe,10)
exten => 2002,2,VoiceMail(2002@ara_voicemail)
exten => 2002,3,PlayBack(vm-goodbye)
exten => 2002,4,HangUp()

exten => _2[13][49]1[01]XXXX,1,Dial(SIP/${EXTEN}@Proxy22,10)
```

This results in a simple configuration. Extension 2001 will call user “ZeManel”, extension 2002 will call user “ManelZe” and in case of a number a PSTN number, the call is forwarded to Proxy2.

- Proxy2 will simply dictate the number called as a test

```
[phones]
exten => _2[13][49]1[01]XXXX,1,Answer(500)
exten => _2[13][49]1[01]XXXX,2,Playback(vm-received)
exten => _2[13][49]1[01]XXXX,3,SayDigits(${EXTEN:3})
exten => _2[13][49]1[01]XXXX,n,PlayBack(vm-goodbye)
exten => _2[13][49]1[01]XXXX,n,Hangup()
```

Extend this VoIP-SIP service for corporate Client A by using alternatively both SIP Proxies:

We accomplished this by what we thought to be the most realistic, yet lazy, way. Assuming that clients will connect to the SIP proxy via a domain (sip.pt, for example), the DNS server will return a zone similar to the one discussed before, one where two A addresses are defined giving it an rotating decision between Proxy 1 and BackUp.

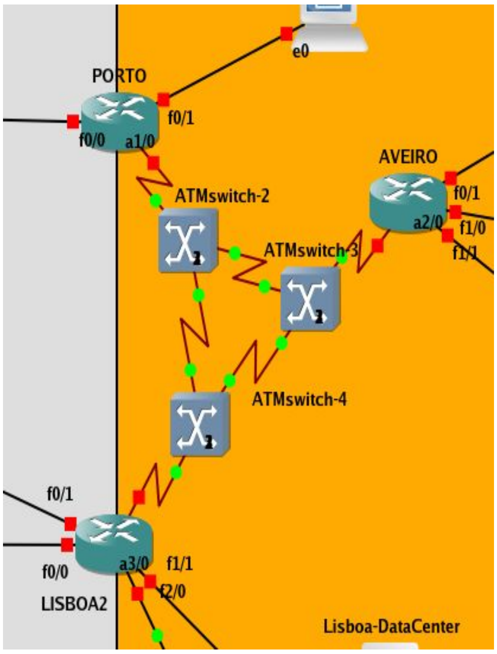
- Just like it was done to the datacenters, we created the zone files needed and returned to the users of Empresa A in zone that gives a rotating decision on the SIP Proxies.

```
$ORIGIN sip.pt.
@        IN      SOA     ns1.sip.pt.     adm.sip.pt.     (
                         2               ; Serial
                    604800               ; Refresh
                     86400               ; Retry
                   2419200               ; Expire
                    604800 )             ; Negative Cache TTL
         IN      NS      ns1.sip.pt.
         IN      A       2.2.2.2
         IN      A       3.3.3.3
ns1      IN      A       10.2.0.2
```

- ATM Networks

In ISP PT2 we installed two ATM networks with point-to-point connections which gives a more realistic scenario of an ISP's autonomous system.

- The first one was installed in the core network between the ISP stations in Aveiro, Porto and Lisboa 2.



For simplicity, we used VPI and ignored VCI (always 0). At each ATM Switch, it routes the local station to the other two. This gives a total of 3 VP tunnels:

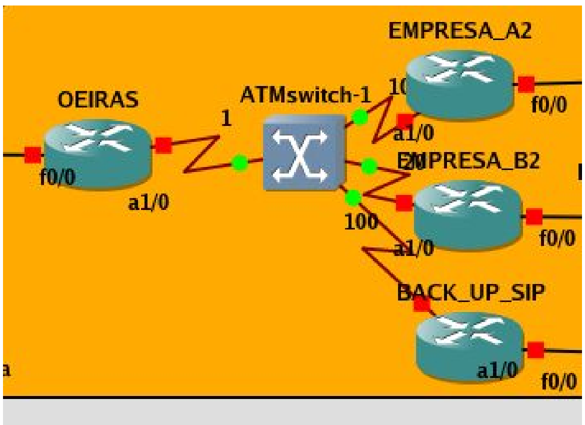
- VPI 1 - PORTO <-> Lisboa2
- VPI 2 - PORTO <-> Aveiro
- VPI 3 - Lisboa2 <-> Aveiro

Example of configuration on ATMswitch-2 that routes PORTO station:

Mapping

Port:VPI:VCI	Port:VPI:VCI
1:1:0	2:1:0
1:2:0	3:2:0

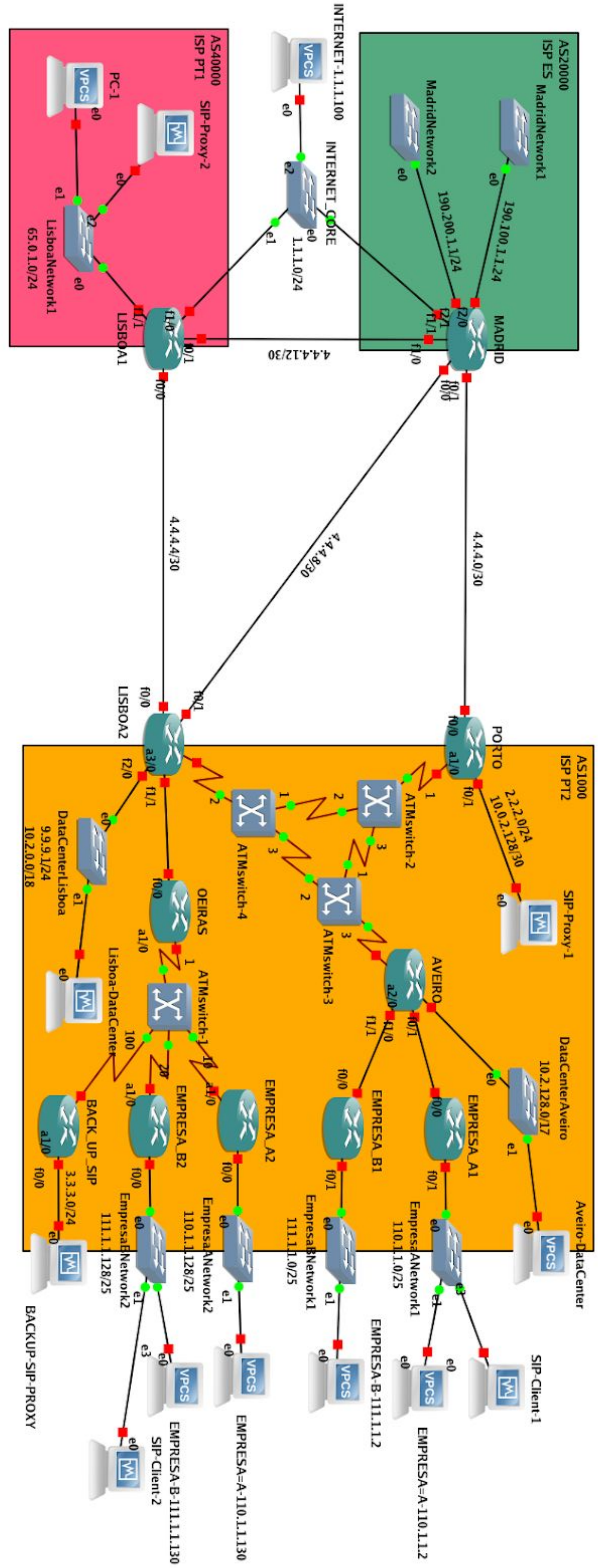
- Since netD1 was described as a ADSL network, some kind of switching was needed to simulate a DSLAM. For that, we used an ATM Switch.



Again, the ATM Switch is simply using VPI for simplicity. We have three VP tunnels that connect Oeiras to the three endpoints via point-to-point connections.

Mapping

Port:VPI:VCI	Port:VPI:VCI
1:10:0	10:10:0
1:100:0	100:100:0
1:20:0	20:20:0


AS40000
ISP PT1
PC-1
SIP-Proxy-2
LisboaNetwork1
65.0.1.0/24
LISBOA1
INTERNET-1.1.1.100
INTERNET_CORE
1.1.1.0/24
AS20000
ISP ES
MadridNetwork1
MadridNetwork2
190.100.1.1/24
190.200.1.1/24
MADRID
4.4.4.12/30
4.4.4.4/30
4.4.4.8/30
4.4.4.0/30
LISBOA2
PORTO
AS1000
ISP PT2
2.2.2.0/24
10.0.2.128/30
SIP-Proxy-1
ATMswitch-2
ATMswitch-4
ATMswitch-3
ATMswitch-1
OEIRAS
AVEIRO
DataCenterLisboa
9.9.9.1/24
10.2.0.0/18
Lisboa-DataCenter
DataCenterAveiro
10.2.128.0/17
Aveiro-DataCenter
EMPRESA_A1
EMPRESA_B1
EMPRESA_A2
EMPRESA_B2
BACK_UP_SIP
3.3.3.0/24
EmpresaANetwork1
110.1.1.0/25
EmpresaBNetwork1
111.1.1.0/25
EmpresaANetwork2
110.1.1.128/25
EmpresaBNetwork2
111.1.1.128/25
BACKUP-SIP-PROXY
SIP-Client-1
SIP-Client-2
EMPRESA=A-110.1.1.2
EMPRESA-B-111.1.1.2
EMPRESA=A-110.1.1.130
EMPRESA-B-111.1.1.130