

Projeto de Arquitetura de Redes

João Bernardino
78022

Tomás Jacob
78034

Filipe Fernandes
78083

9 de Dezembro de 2016

Parte 2 do projeto

Após devidamente configurada toda a parte 1 do projeto, estivemos a analisar as sugestões dadas para a 2ª parte e optámos por arriscar no desenvolvimento de algo menos previsível. O primeiro passo foi vermos que tecnologias é que usávamos regularmente e sobre quais é que gostávamos de saber mais. Uma das opções era recrear uma aplicação de download de ficheiros com recurso a uma distribuição peer-to-peer, mas achámos que era de uma complexidade demasiado grande para a rede que configurámos.

Backup do Servidor DNS

Então optámos por complementar uma das tecnologias que já havíamos usado na primeira parte. Fazer um backup do servidor DNS configurado antes. Mas para não ficarmos por aqui, decidimos fazê-lo num Raspberry Pi.

A montagem ficou então dividida em três partes:

1. Fazer a ligação do Raspberry Pi ao computador

Usámos para a montagem um Raspberry Pi Model B, com uma porta Ethernet. A ligação foi feita a partir desta porta a um Macintosh. No Raspberry apenas tivemos de configurar o ficheiro relativo às interfaces do mesmo, localizado por default em `/etc/network/interfaces`. Alterámos precisamente a definição da porta Ethernet para a apropriada:

```
iface eth0 inet static
address 201.14.5.1
netmask 255.255.255.0
network 201.14.5.0
broadcast 201.14.5.255
gateway 201.14.5.254
```

No Macintosh definimos manualmente o IP da porta Ethernet Tunderbolt para o mesmo *IP*, *gateway* e *netmask* que no código acima. Desta maneira, o computador transforma a porta numa ligação “direta” ao raspberry. Neste momento, é pouco diferente de um dos Routers que utilizamos no GNS.

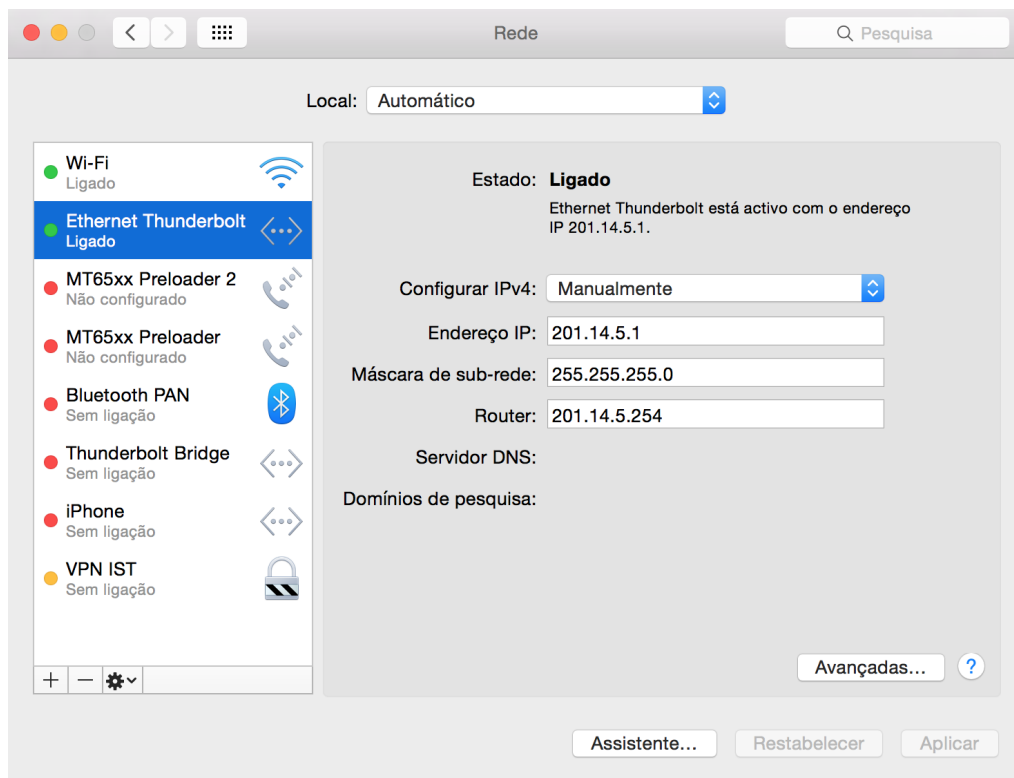


Figura 1: A configuração efetuado na secção de Rede do Machintosh

2. Ligar o Raspberry Pi à rede virtual no GNS3

Terminada a montagem física, partimos para a ligação ao GNS3. O mecanismo *Cloud* disponível no programa, representa uma rede exterior, que no nosso caso se adequa. Decidimos colocar a *cloud* ligada a um router R8, que por sua vez se ligou ao SW8 na entrada 2. Para facilitar, criámos a área OSPF 5 em que ficou o router e a *cloud*.

A configuração da *cloud* foi simples, apenas tivemos de a associar à porta Ethernet do Machintosh que preparámos anteriormente.

3. Configurar o servidor DNS no Raspberry Pi

Neste momento, apenas nos falta preparar o Raspberry para ser o backup DNS. E aqui a configuração é em tudo parecida com a que efetuámos também anteriormente na máquina virtual Ubuntu. O sistema operativo dos Raspberry Pi's utiliza o kernel de Linux, graças a isto, é como se tivéssemos um autêntico computador à nossa disposição.

O código que utilizámos foi o seguinte.

No ficheiro `/etc/bind/named.conf.local`:

```
zone 'ar.org' {
    type slave;
    file 'db.ar.org';
    masters{ 201.14.1.1; };
};

zone '14.201.in-addr.arpa' {
    type slave;
    file 'db.201.14';
    masters { 201.14.1.1; };
};
```

No ficheiro `/etc/bind/named.conf.options`:

```
acl 'trusted' {  
    201.14.0.0/16 ;  
};  
  
options {  
    directory '/var/cache/bind';  
  
    recursion yes;  
    allow-recursion { trusted };  
    listen-on { 201.14.5.1 };  
    allow-transfer { none; };  
  
    forwarders {  
        8.8.8.8;  
        8.8.4.4;  
    };  
};
```

No ficheiro `/etc/default/bind9`:

```
OPTIONS='-4 -u bind'
```

Aqui foi necessário também alterar a máquina virtual que suportava o servidor primário, para permitir a existência de um backup.

Neste ponto, a única alteração que ainda tínhamos de fazer era dar a conhecer a todos os routers que agora existia um backup de DNS. Usando o comando: “nameserver 201.14.1.1 201.14.5.1”

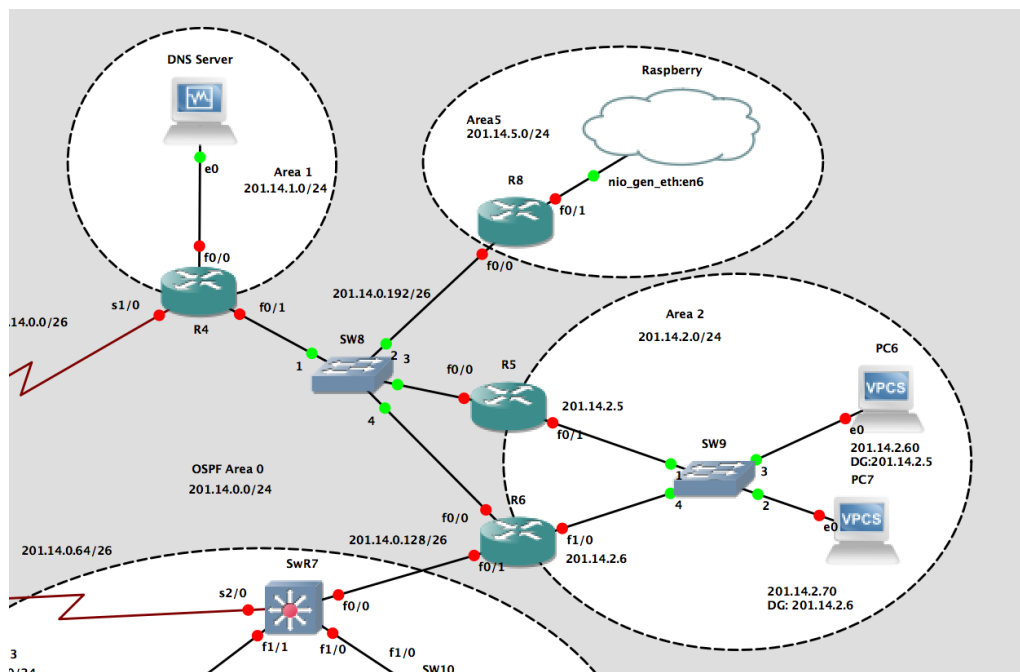


Figura 2: O aspeto final da montagem no GNS3, onde é possível ver a área OSPF 5, o seu router e a cloud.

Para verificarmos se o sistema está a funcionar como pretendido, pode-se efetuar um simples teste. Se desligarmos (comando shut) a interface f0/0 de R?? o servidor de DNS fica inacessível. Quando algum outro router tentar traduzir um endereço, vai ser chamado o principal, e de seguida o secundário. Na linha de comandos do router, é visível a chamada ao backup, com o endereço 201.14.5.1 .

```

Filipe — R3 — telnet — 80x24

0 IA 201.14.5.0/24 [110/94] via 201.14.4.129, 00:00:02, FastEthernet0/0
  201.14.4.0/26 is subnetted, 4 subnets
C    201.14.4.192 is directly connected, FastEthernet1/0
C    201.14.4.128 is directly connected, FastEthernet0/0
C    201.14.4.64 is directly connected, FastEthernet0/1
0    201.14.4.0 [110/20] via 201.14.4.129, 00:00:02, FastEthernet0/0
0 IA 201.14.1.0/24 [110/84] via 201.14.4.129, 00:00:02, FastEthernet0/0
  201.14.0.0/26 is subnetted, 3 subnets
0 IA  201.14.0.192 [110/84] via 201.14.4.129, 00:00:03, FastEthernet0/0
0 IA  201.14.0.128 [110/94] via 201.14.4.129, 00:00:03, FastEthernet0/0
0 IA  201.14.0.0 [110/74] via 201.14.4.129, 00:00:03, FastEthernet0/0
0 IA 201.14.2.0/24 [110/85] via 201.14.4.129, 00:00:03, FastEthernet0/0
R3#ping h6.ar.org
Translating "h6.ar.org"...domain server (201.14.1.1) (201.14.5.1)
Translating "h6.ar.org"...domain server (201.14.1.1) (201.14.5.1) [OK]
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 201.14.2.60, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 80/85/88 ms
R3#
R3#
  
```

Figura 3: Aqui é visível R3 a recorrer ao servidor de DNS secundário para traduzir um endereço

Servidor FTP

Para complementar esta segunda parte, decidimos também criar um servidor FTP. Para evitar interferir com a programação do servidor DNS primário, que estava na máquina virtual, decidimos usar novamente o Raspberry como *host*.

Por defeito, o sistema operativo do Raspberry já permite acessos FTP, no entanto, apenas o utilizador root o pode fazer e fica com um acesso virtualmente ilimitado ao seu conteúdo. Portanto preferimos instalar um *package* chamado *Pure-FTPD*.

A configuração revolveu muito à volta das propriedades que o package autorizava. O primeiro passo foi criar um tipo de utilizadores, e configurar as suas permissões, à semelhança do que acontece em sistemas operativos como o Linux. Aqui, criámos um utilizador e incluímo-lo no grupo anterior. Por fim, no sistema de ficheiros do Raspberry fizemos um diretório base, que é o acessível pelos clientes do servidor.

Agora para acedermos ao servidor, utilizámos a máquina virtual já existente. Através do seu terminal podemos descarregar um ficheiro, como na seguinte demonstração:

```
filipe@filipe-VirtualBox: ~
filipe@filipe-VirtualBox:~$
filipe@filipe-VirtualBox:~$ ftp 201.14.5.1
Connected to 201.14.5.1.
220----- Welcome to Pure-FTPd [privsep] [TLS] -----
220-You are user number 1 of 50 allowed.
220-Local time is now 19:25. Server port: 21.
220-This is a private system - No anonymous login
220 You will be disconnected after 15 minutes of inactivity.
Name (201.14.5.1:filipe): ar
331 User ar OK. Password required
Password:
230 OK. Current directory is /
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful
150 Connecting to port 53833
-rw-r--r-- 1 0 0 13 Dec 8 18:14 hello.txt
226-Options: -l
226 1 matches total
ftp> get hello.txt
local: hello.txt remote: hello.txt
200 PORT command successful
150 Connecting to port 50771
226-File successfully transferred
226 0.001 seconds (measured here), 16.64 Kbytes per second
13 bytes received in 0.02 secs (0.5239 kB/s)
ftp>
```

Figura 4: O utilizador AR, fez download do ficheiro hello.txt

Capturing from Standard input — R8 FastEthernet0/1 to Raspberry nio_gen_eth:en6

Apply a display filter ... <f>

No.	Time	Source	Destination	Protocol	Length	Info
4	8.893428	201.14.5.1	201.14.1.1	TCP	74	21-33118 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_P...
5	8.926348	201.14.1.1	201.14.5.1	TCP	66	33118->21 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=241788 TSecr=28...
6	8.945562	201.14.5.1	201.14.1.1	DNS	83	Standard query 0x7bde PTR 1.1.14.201.in-addr.arpa
7	8.975192	201.14.1.1	201.14.5.1	DNS	171	Standard query response 0x7bde PTR 1.1.14.201.in-addr.arpa PTR n...
8	8.986604	201.14.5.1	201.14.1.1	FTP	331	Response: 220----- Welcome to Pure-FTPd [privsep] [TLS] ---
9	9.012185	201.14.1.1	201.14.5.1	TCP	66	33118->21 [ACK] Seq=1 Ack=266 Win=30336 Len=0 TSval=241808 TSecr=...
10	12.407038	201.14.1.1	201.14.5.1	FTP	75	Request: USER ar
11	12.417179	201.14.5.1	201.14.1.1	TCP	66	21-33118 [ACK] Seq=266 Ack=10 Win=28992 Len=0 TSval=281717 TSecr=...
12	12.417229	201.14.5.1	201.14.1.1	FTP	101	Response: 331 User ar OK. Password required
13	12.467808	201.14.1.1	201.14.5.1	TCP	66	33118->21 [ACK] Seq=10 Ack=301 Win=30336 Len=0 TSval=242671 TSecr=...
14	15.063819	c2:0e:22:0f:00:01	c2:0e:22:0f:00:01	LOOP	60	Reply
15	18.982644	201.14.1.1	201.14.5.1	FTP	82	Request: PASS raspberry
16	19.014585	201.14.5.1	201.14.1.1	FTP	98	Response: 230 OK. Current directory is /
17	19.041700	201.14.1.1	201.14.5.1	TCP	66	33118->21 [ACK] Seq=26 Ack=333 Win=30336 Len=0 TSval=244317 TSecr=...

Frame 15: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0

Ethernet II, Src: c2:0e:22:0f:00:01 (c2:0e:22:0f:00:01), Dst: Raspberr_db:3a:d5 (b8:27:eb:db:3a:d5)

Internet Protocol Version 4, Src: 201.14.1.1, Dst: 201.14.5.1

Transmission Control Protocol, Src Port: 33118, Dst Port: 21, Seq: 10, Ack: 301, Len: 16

File Transfer Protocol (FTP)

PASS raspberry\r\n

Request command: PASS

Request arg: raspberry

Figura 5: Uma das razões pela qual o protocolo FTP já não é utilizado regularmente, é a transmissão das mensagens descriptadas o que pode causar problemas de segurança.