

Trabalho 1 - Encaminhamento Estático

Miguel Ferreira
miguelferreira108@gmail.com
Vanessa Silva
up201305731@fc.up.pt

*Administração de Redes,
Departamento de Ciências de Computadores,
Faculdade de Ciências da Universidade do Porto*

29 de Março de 2016

Introdução

No âmbito da unidade curricular de Administração de Redes, implementamos a rede descrita na figura ??.

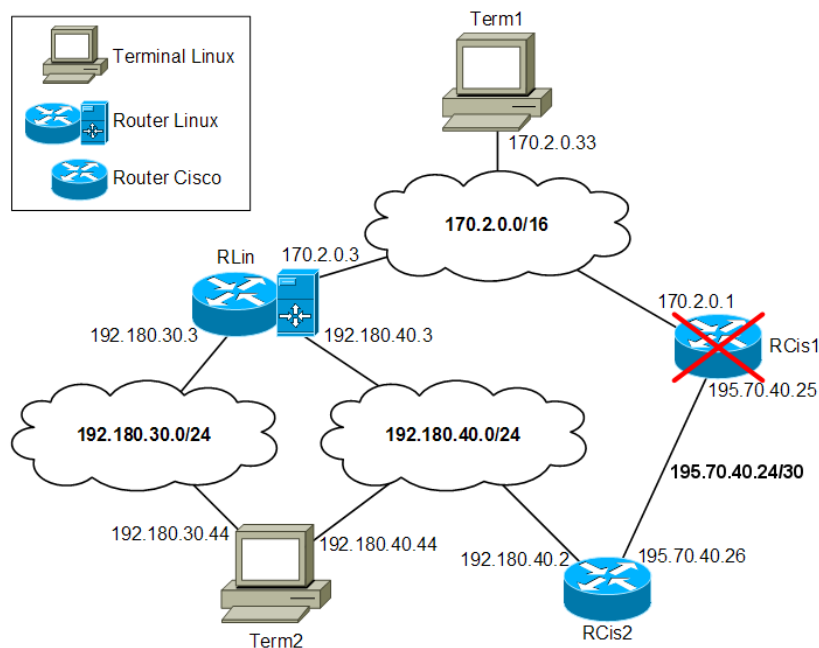


Figura 1: Rede implementada na aula.

As máquinas **Term1**, **Term2** e **RLin** foram concretizadas com 3 *workstations* Fedora 23, cada uma com as necessárias interfaces de rede. A máquina **RCis2** consiste num *router* Cisco 1841. A rede 170.2.0.0/16 consiste na ligação direta entre uma interface de **RLin** e **Term1**. A rede 192.180.30.0/24 foi implementada com uma ligação direta entre **RLin** e **Term2**. A rede 192.180.40.0/24 foi implementada com um *switch* ligado a **RLin**, **Term2** e **RCis2**.

1 Conectividade

a)

```
root@localhost ar]# traceroute -n -N 1 192.180.30.3
traceroute to 192.180.30.3 (192.180.30.3), 30 hops max, 60 byte packets
 1  192.180.30.3  0.458 ms  0.189 ms  0.187 ms
[root@localhost ar]# traceroute -n -N 1 192.180.40.3
traceroute to 192.180.40.3 (192.180.40.3), 30 hops max, 60 byte packets
 1  192.180.40.3  0.557 ms  0.071 ms  0.064 ms
[root@localhost ar]# traceroute -n -N 1 170.2.0.3
traceroute to 170.2.0.3 (170.2.0.3), 30 hops max, 60 byte packets
 1  192.180.40.2  0.462 ms  0.366 ms  0.382 ms
 2  * * *
 3  * * *
 4  * * *
 5  *
```

b)

```
[root@localhost ar]# traceroute -n -N 1 192.180.40.44
traceroute to 192.180.40.44 (192.180.40.44), 30 hops max, 60 byte packets
 1  170.2.0.33  3005.124 ms !H  3005.860 ms !H  3005.921 ms !H
[root@localhost ar]# traceroute -n -N 1 192.180.30.44
traceroute to 192.180.30.44 (192.180.30.44), 30 hops max, 60 byte packets
 1  170.2.0.3  0.331 ms  0.225 ms  0.233 ms
 2  * * *
 3  * * *
 4  * * *
 5  *
```

i)

```
[root@localhost ar]# traceroute -n -N 1 192.180.40.44
traceroute to 192.180.40.44 (192.180.40.44), 30 hops max, 60 byte packets
 1  170.2.0.33  3005.124 ms !H  3005.860 ms !H  3005.921 ms !H
[root@localhost ar]# traceroute -n -N 1 192.180.30.44
traceroute to 192.180.30.44 (192.180.30.44), 30 hops max, 60 byte packets
 1  170.2.0.3  0.331 ms  0.225 ms  0.233 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
```

ii) Comandos Cisco:

```
router_g04>enable
router_g04#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
router_g04(config)#ip route 170.2.0.0 255.255.0.0 195.70.40.25
router_g04(config)#end
```

Terminal:

```
[root@localhost ar]# traceroute -n -N 1 192.180.40.44
traceroute to 192.180.40.44 (192.180.40.44), 30 hops max, 60 byte packets
 1  170.2.0.33  3004.524 ms !H  3005.865 ms !H  3005.837 ms !H
[root@localhost ar]# traceroute -n -N 1 192.180.30.44
traceroute to 192.180.30.44 (192.180.30.44), 30 hops max, 60 byte packets
 1  170.2.0.3  0.330 ms  0.071 ms  0.068 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
```

iii) Comandos Cisco:

```
router_g04>enable
router_g04#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
router_g04(config)#ip route 170.2.0.0 255.255.0.0 192.180.40.3
router_g04(config)#end
```

Terminal:

```
[root@localhost ar]# traceroute -n -N 1 192.180.40.44
traceroute to 192.180.40.44 (192.180.40.44), 30 hops max, 60 byte packets
 1  170.2.0.33  3005.787 ms !H  3005.895 ms !H  3005.845 ms !H
[root@localhost ar]# traceroute -n -N 1 192.180.30.44
traceroute to 192.180.30.44 (192.180.30.44), 30 hops max, 60 byte packets
 1  170.2.0.3  0.115 ms  0.067 ms  0.067 ms
 2  192.180.30.44  0.406 ms  0.684 ms  0.303 ms
```

c) Na alínea b) pretendemos testar a conectividade entre as máquinas *Term1* e *Term2*. Em b) i) os pacotes chegam à máquina *Term2* pela interface *enp1s6* (192.180.30.44) que envia os pacotes, como resposta, pela interface *enp0s7* (192.180.40.44) para o router *RCis2* (192.180.40.2), uma vez que é a sua rota por defeito, como *RCis2* não tem rota para a rede 170.2.0.0/16 os pacotes são perdidos.

Em b) ii) os pacotes chegam a *Term2* pela interface *enp1s6* (192.180.30.44) que envia os pacotes, como resposta, pela interface *enp0s7* (192.180.40.44) para o router *RCis2*, (rota por defeito), como este é configurado com uma rota para a rede 170.2.0.0/16 através de *RCis1*, que é um router fisicamente inexistente, a resposta nunca chega a *Term1*.

Em b) iii) o percurso dos pacotes é idêntico ao descrito para a alínea b) ii), à exceção de que, neste caso, o router *RCis2* é configurado com uma rota para a rede 170.2.0.0/16 através da máquina *RLin*. *RCis2* envia então os pacotes para *RLin* (192.180.40.3) que reenvia-os para a máquina de origem.

Em todos os casos, b) i), b) ii) e b) iii), quando a máquina *Term1* envia os pacotes para a interface *enp0s7*, que tem o endereço IP 192.180.40.44, os pacotes não chegam ao destino. Isto acontece pois *Term1* tem uma rota para a rede 192.180.40.0/24 através de *RCis1*, que é um router fisicamente inexistente.

Rotas da máquina Term1:

```
[root@localhost ar]# route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
0.0.0.0          170.2.0.3       0.0.0.0          UG      0      0      0 enp0s7
169.254.0.0      0.0.0.0         255.255.0.0      U       1002   0      0 enp0s7
170.2.0.0        0.0.0.0         255.255.0.0      U       0      0      0 enp0s7
192.180.40.0     170.2.0.1       255.255.255.0    UG      0      0      0 enp0s7
```

Rotas da máquina Term2:

```
[root@localhost ar]# route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
0.0.0.0          192.180.40.2    0.0.0.0          UG      0      0      0 enp0s7
169.254.0.0      0.0.0.0         255.255.0.0      U       1002   0      0 enp0s7
169.254.0.0      0.0.0.0         255.255.0.0      U       1003   0      0 enp1s6
192.180.30.0     0.0.0.0         255.255.255.0    U       0      0      0 enp1s6
192.180.40.0     0.0.0.0         255.255.255.0    U       0      0      0 enp0s7
```

d) Como vimos, perante algumas condições estipuladas, os pacotes enviados da máquina Term1 à Term2 foram perdidos, ou não chegaram ao destino ou a resposta não chegou à origem.

No encaminhamento dinâmico, tal como o nome indica, as rotas (caminhos) são calculados dinamicamente recorrendo a protocolos de encaminhamento dinâmico que se adaptam a possíveis alterações que acontecem na rede. Estes protocolos envolvem a troca de informação de controlo entre os nós (terminais e routers) encaminhadores, são desenvolvidos para trocar para uma rota alternativa quando a rota primária se torna inoperável e para decidir qual é a melhor rota para o destino.

Com este tipo de encaminhamento os pacotes perdidos, muito provavelmente, não seria perdidos, uma vez que os protocolos de encaminhamento dinâmico atualizam automaticamente as tabelas de encaminhamento dos routers, de modo a encontrarem a rota ideal para chegar ao destino.

e)

i)

```
[root@localhost ~]# traceroute -n -N 1 170.2.0.33
traceroute to 170.2.0.33 (170.2.0.33), 30 hops max, 60 byte packets
 1  192.180.40.3  0.093 ms  0.064 ms  0.063 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * *
```

ii) Comandos Cisco:

```
router_g04>enable
router_g04#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
router_g04(config)#ip route 170.2.0.0 255.255.0.0 195.70.40.25
router_g04(config)#end
```

Terminal:

```
[root@localhost ~]# traceroute -n -N 1 170.2.0.33
traceroute to 170.2.0.33 (170.2.0.33), 30 hops max, 60 byte packets
 1  192.180.40.3  0.294 ms  0.165 ms  0.072 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * *
```

iii) *Comandos Cisco:*

```
router_g04>enable
router_g04#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
router_g04(config)#ip route 170.2.0.0 255.255.0.0
router_g04(config)#ip route 170.2.0.0 255.255.0.0 192.180.40.3
router_g04(config)#end
```

Terminal:

```
[root@localhost ~]# traceroute -n -N 1 170.2.0.33
traceroute to 170.2.0.33 (170.2.0.33), 30 hops max, 60 byte packets
 1  192.180.40.3  0.307 ms  0.247 ms  0.065 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * *
```

f) *Sim, seria útil ter encaminhamento dinâmico nos routes, pois mais uma vez vemos que os pacotes, em e) i) e e) ii), perdem-se, e em e) iii) chegam ao destino mas, neste caso a resposta é enviada para o router RCis1 que não existe fisicamente e por isso também se perdem. Com o encaminhamento dinâmico os routers escolhiam sempre a melhor rota, como explicado na alínea d), de modo a que estes problemas se resolvessem.*

g)

i)

```
[root@localhost ar]# traceroute -n -N 1 170.2.0.15
traceroute to 170.2.0.15 (170.2.0.15), 30 hops max, 60 byte packets
 1  192.180.40.2  0.491 ms  0.376 ms  0.384 ms
 2  192.180.40.3  0.601 ms  3005.145 ms !H  3005.706 ms !H
[root@localhost ar]# traceroute -n -N 1 170.2.0.15
traceroute to 170.2.0.15 (170.2.0.15), 30 hops max, 60 byte packets
 1  192.180.40.3  0.100 ms  0.070 ms  0.069 ms
 2  192.180.40.3  3005.710 ms !H  3005.788 ms !H  3005.939 ms !H
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	CiscoInc_78:e0:95	CDP/VTP/DTLP/PAgP/LDLD	CDP	343	Device ID: router_g04 Port ID: FastEthernet0/1
2	2.937129000	192.180.40.44	170.2.0.15	UDP	74	Source port: 54318 Destination port: 33434
3	2.937600000	192.180.40.2	192.180.40.44	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
4	2.937719000	192.180.40.44	170.2.0.15	UDP	74	Source port: 43891 Destination port: 33435
5	2.938085000	192.180.40.2	192.180.40.44	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
6	2.938119000	192.180.40.44	170.2.0.15	UDP	74	Source port: 37468 Destination port: 33436
7	2.938500000	192.180.40.2	192.180.40.44	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
8	2.938529000	192.180.40.44	170.2.0.15	UDP	74	Source port: 41765 Destination port: 33437
9	2.938985000	192.180.40.2	192.180.40.44	ICMP	70	Redirect (Redirect for network)
10	2.939126000	192.180.40.3	192.180.40.44	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
11	2.939187000	192.180.40.44	170.2.0.15	UDP	74	Source port: 47944 Destination port: 33438
12	5.944324000	192.180.40.3	192.180.40.44	ICMP	102	Destination unreachable (Host unreachable)
13	5.944702000	192.180.40.44	170.2.0.15	UDP	74	Source port: 53291 Destination port: 33439
14	7.944340000	Asustek_C3:2b:84	Asustek_C3:2b:84	ARP	60	Who has 192.180.40.44? Tell 192.180.40.3
15	7.944364000	Asustek_C3:2b:84	Asustek_C2:17:9b	ARP	42	192.180.40.44 is at 48:5b:39:c3:2b:84
16	7.948017000	Asustek_C3:2b:84	CiscoInc_78:e0:95	ARP	42	Who has 192.180.40.2? Tell 192.180.40.44
17	7.948463000	CiscoInc_78:e0:95	Asustek_C3:2b:84	ARP	60	192.180.40.2 is at 00:17:59:78:e0:95
18	8.950396000	192.180.40.3	192.180.40.44	ICMP	102	Destination unreachable (Host unreachable)
19	11.384591000	192.180.40.44	170.2.0.15	UDP	74	Source port: 46389 Destination port: 33434
20	11.384672000	192.180.40.3	192.180.40.44	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
21	11.384749000	192.180.40.44	170.2.0.15	UDP	74	Source port: 47119 Destination port: 33435
22	11.384815000	192.180.40.3	192.180.40.44	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
23	11.384850000	192.180.40.44	170.2.0.15	UDP	74	Source port: 41768 Destination port: 33436
24	11.384916000	192.180.40.3	192.180.40.44	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
25	11.384947000	192.180.40.44	170.2.0.15	UDP	74	Source port: 38413 Destination port: 33437
26	14.390654000	192.180.40.3	192.180.40.44	ICMP	102	Destination unreachable (Host unreachable)
27	14.390814000	192.180.40.44	170.2.0.15	UDP	74	Source port: 49365 Destination port: 33438
28	16.396022000	Asustek_C3:2b:84	Asustek_C2:17:9b	ARP	42	Who has 192.180.40.2? Tell 192.180.40.44
29	16.396089000	Asustek_C2:17:9b	Asustek_C3:2b:84	ARP	60	192.180.40.3 is at 00:24:8c:c2:17:9b

Frame 1: 343 bytes on wire (2744 bits), 343 bytes captured (2744 bits) on interface 0

- IEEE 802.3 Ethernet
- Logical-Link Control
- Cisco Discovery Protocol

Figura 2: Screenshot da captura de pacotes na interface enp0s7 (IP 192.180.40.44) no wireshark.

ii) No capture do wireshark, da figura anterior, podemos verificar que o router Cisco envia um ICMP redirect message para a máquina Term2, de modo a comunicar-lhe que existe uma melhor rota. Com esta informação o Term2 atualiza a sua rota para o IP comunicado (192.180.40.3) e na chamada seguinte do traceroute, envia já diretamente para RLin (192.180.40.3), em vez de passar por RCis2, rota "antiga"(192.180.40.2).

O Internet Control Message Protocol(ICMP) é usado, entre outras coisas, para os terminais e routers trocarem informação sobre o seu funcionamento, controlo do fluxo de informação e para trocarem mensagens de controlo, bem como informação relativa à escolha de rotas até um determinado destino.

Um router pode detetar uma rota alternativa entre o emissor e o recetor e enviar uma mensagem ao emissor, usando o ICMP, para o informar que existe uma melhor rota (ICMP redirect message).

2 ARP

a) Resultado do ping:

```

ar@localhost:/home/ar
File Edit View Search Terminal Help
[root@localhost ar]# ping -c 1 192.180.40.55
PING 192.180.40.55 (192.180.40.55) 56(84) bytes of data:
From 192.180.40.44 icmp_seq=1 Destination Host Unreachable

--- 192.180.40.55 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

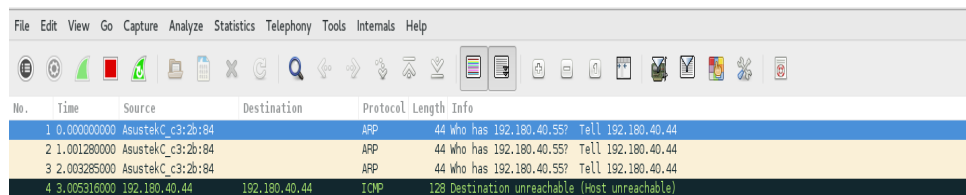
[root@localhost ar]#

```

Figura 3: Resultado de ping -c 1 192.180.40.55.

b) Como podemos verificar no screenshot abaixo, capturou-se 3 pacotes ARP de modo a "resolver" o endereço IP 192.180.40.55. Entre as tentativas houve um timeout de aproximadamente 1 segundo, uma vez que entre elas (retransmissões) não se conseguiu "resolver" o endereço IP e, depois da terceira tentativa sem resposta, é enviado um pacote ICMP Host Unreachable ao IP de origem.

É, exatamente, desta forma que funciona o protocolo ARP, ou seja, quando a tradução de um endereço IP não se encontra na cache de ARP é necessário "resolver" esse endereço. Para tal, é enviado uma trama para o endereço MAC de difusão (pacote ARP), todos os nós (terminais e/ou routers) recebem e processam a trama, mas apenas o que reconhece o endereço pretendido responde e a informação é adicionada à cache de ARP. Se não ocorrer a resposta no espaço de 1 segundo (timeout), a trama é retransmitida e, se à **terceira** tentativa não receber resposta, "desiste" e é enviado um ICMP Host Unreachable (!H) ao IP de origem do pacote.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	Asustek_C3:2b:84	192.180.40.44	ARP	44	who has 192.180.40.55? Tell 192.180.40.44
2	1.00128000	Asustek_C3:2b:84	192.180.40.44	ARP	44	who has 192.180.40.55? Tell 192.180.40.44
3	2.00328500	Asustek_C3:2b:84	192.180.40.44	ARP	44	who has 192.180.40.55? Tell 192.180.40.44
4	3.00531600	192.180.40.44	192.180.40.44	ICMP	128	Destination unreachable (Host unreachable)

Figura 4: Screenshot do wireshark a correr na máquina Term2.

3 Captura

a) Comando com o filtro de captura pretendido:

```
tcpdump -nn 'tcp[tcpflags] & tcp-push == tcp-push' and 'ip[2:2] < 128' > ssh.dump
```

O resultado da captura é o seguinte:

```
11:00:59.019099 IP 170.2.0.33.41460 > 192.180.30.44.22: Flags [P.],
seq 3235348810:3235348854, ack 1796210900, win 288, options [nop,nop,
TS val 2288580 ecr 2411765], length 44
11:00:59.503156 IP 170.2.0.33.41460 > 192.180.30.44.22: Flags [P.],
seq 44:80, ack 93, win 288, options [nop,nop,TS val 2289064 ecr 2470611], length 36
11:00:59.503611 IP 192.180.30.44.22 > 170.2.0.33.41460: Flags [P.], seq 93:129,
ack 80, win 340, options [nop,nop,TS val 2471095 ecr 2289064], length 36
11:00:59.503638 IP 192.180.30.44.22 > 170.2.0.33.41460: Flags [P.], seq 129:173,
ack 80, win 340, options [nop,nop,TS val 2471095 ecr 2289064], length 44
11:01:00.504865 IP 192.180.30.44.22 > 170.2.0.33.41460: Flags [P.], seq 173:217,
ack 80, win 340, options [nop,nop,TS val 2472096 ecr 2289064], length 44
11:01:01.505841 IP 192.180.30.44.22 > 170.2.0.33.41460: Flags [P.], seq 217:261,
ack 80, win 340, options [nop,nop,TS val 2473097 ecr 2290066], length 44
11:01:02.506759 IP 192.180.30.44.22 > 170.2.0.33.41460: Flags [P.], seq 261:305,
ack 80, win 340, options [nop,nop,TS val 2474098 ecr 2291067], length 44
11:01:03.507958 IP 192.180.30.44.22 > 170.2.0.33.41460: Flags [P.], seq 305:349,
ack 80, win 340, options [nop,nop,TS val 2475099 ecr 2292068], length 44
11:01:04.508887 IP 192.180.30.44.22 > 170.2.0.33.41460: Flags [P.], seq 349:393,
ack 80, win 340, options [nop,nop,TS val 2476100 ecr 2293069], length 44
11:01:05.509815 IP 192.180.30.44.22 > 170.2.0.33.41460: Flags [P.], seq 393:437,
ack 80, win 340, options [nop,nop,TS val 2477101 ecr 2294070], length 44
```


b) *Filtro de visualização pretendido:*

`tcp.flags.push == 1 && ip.len < 128`

Na figura abaixo, podemos ver o resultado da captura, no wireshark, usando este filtro.

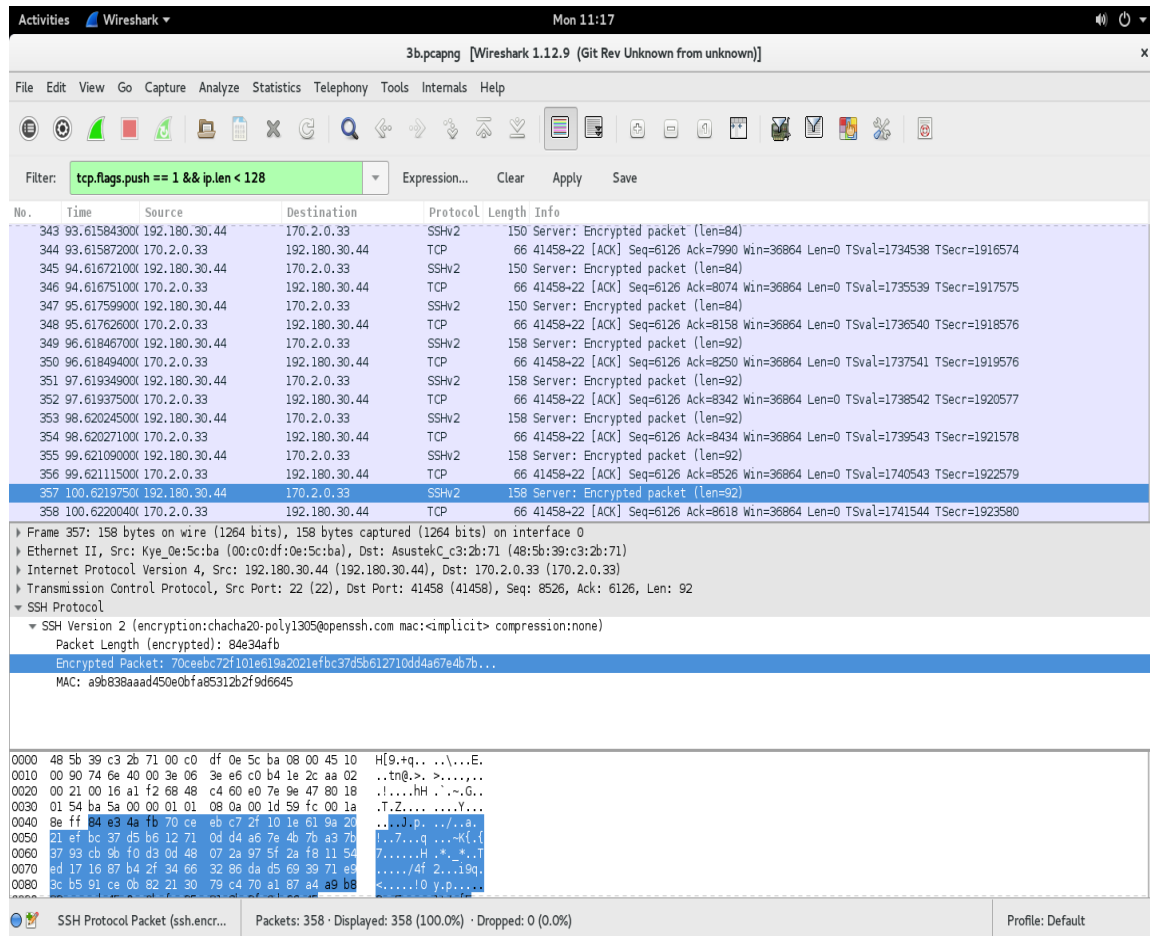


Figura 7: Resultado do filtro de visualização do *wireshark*.

c) *Para calcular o comprimento de um pacote IP subtrai-se o seu tamanho total (cabeçalho + dados), encontrado nos bytes 2 a 3 (indexado a 0), pelo comprimento do próprio cabeçalho, encontrado no primeiro byte, nos últimos 4 bits (mais à direita).*