Trabalho 3 - DHCP e NAT

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

9 de Maio de 2016

Introdução

No âmbito da unidade curricular de Administração de Redes, implementamos a rede com a seguinte configuração física:

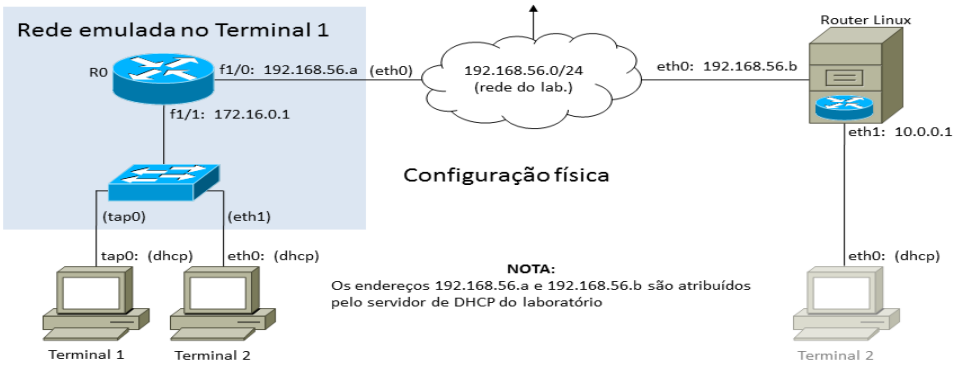

Figura 1: Configuração física da rede.

e com a seguinte configuração lógica:

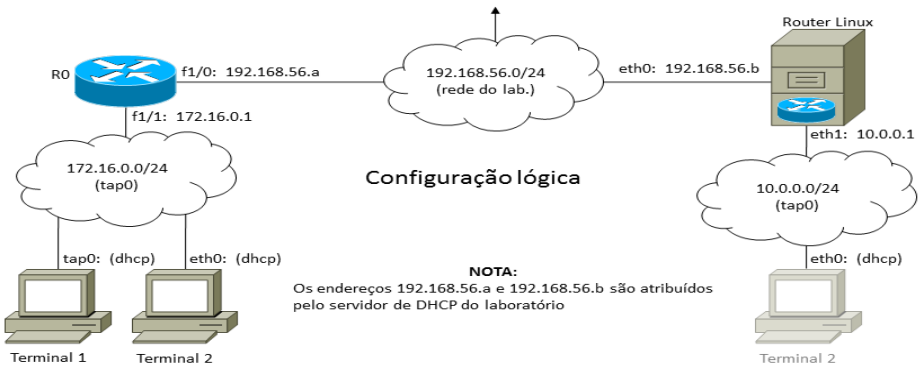

Figura 2: Configuração lógica da rede.

Questões

1.

a. Captura *wireshark* na interface f1/1 de R0, do pedido de IP para interface tap0 do terminal 1:

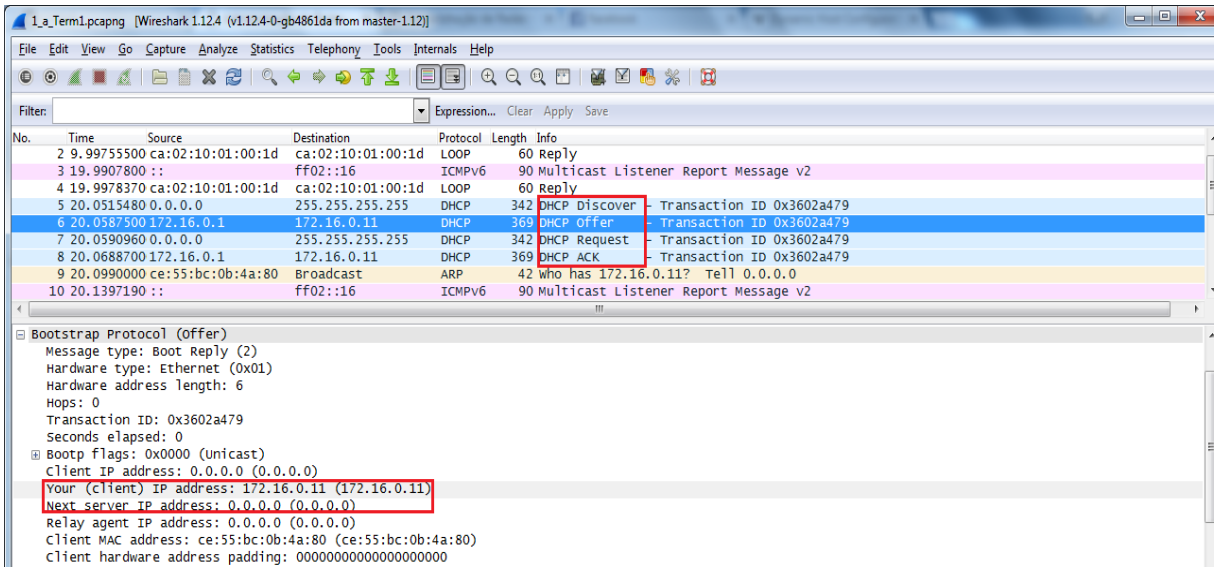

Figura 3: Terminal 1 obtém endereço IP por DHCP.

Captura *wireshark* na interface f1/1 de R0, do pedido de IP para interface eth0 do terminal 2:

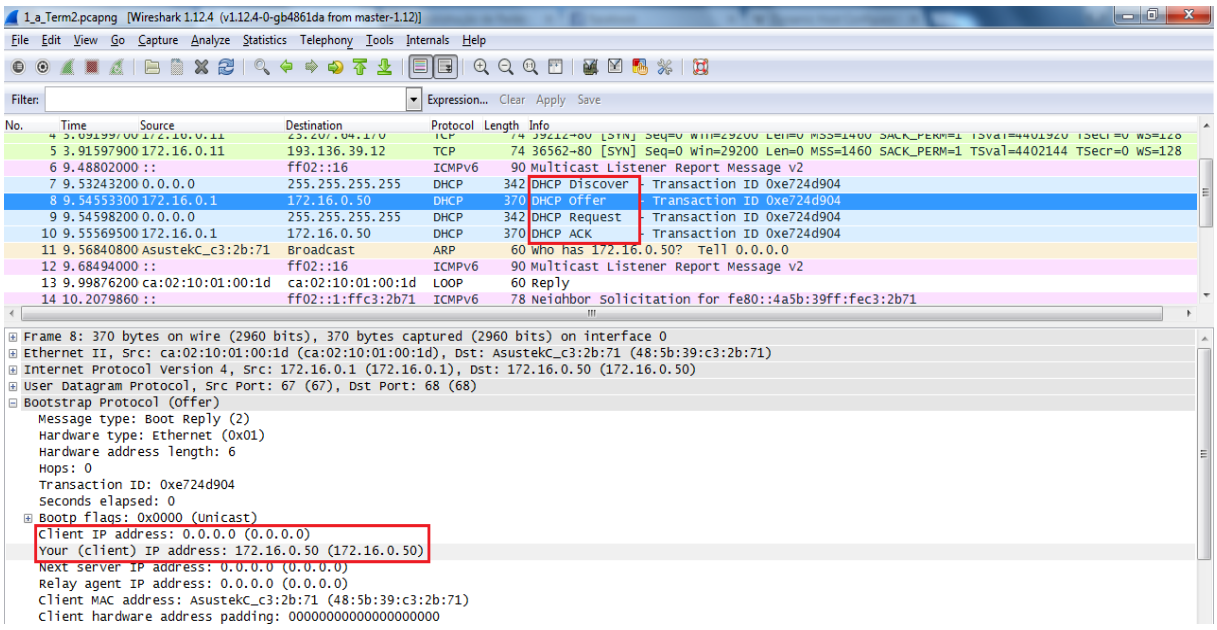

Figura 4: Terminal 2 obtém endereço IP por DHCP.

b. Não é possível, a partir das mensagens capturadas, saber qual das duas máquinas tem um endereço estático (reservado). As mensagens capturadas mostram uma sequência DHCP Discover, Offer, Request e Ack válida.

No entanto, sabendo que o intervalo da pool do DHCP não contem o IP atribuído a Terminal 1, podemos concluir que este IP lhe está reservado.

Se tivéssemos observado uma mensagem do tipo DHCP Inform, saberíamos que o atual cliente tem um IP estático definido por si e apenas pretende obter informação extra (Servidor DNS e Default Gateway, por exemplo).

2.

```
Router#debug ip nat detailed
IP NAT detailed debugging is on
Router#
*May  4 15:48:33.015: NAT: Allocated Port for 172.16.0.11 -> 192.168.56.178:
 wanted 39340 got 39340
*May  4 15:48:33.015: NAT*: i: tcp (172.16.0.11, 39340) -> (193.136.39.12, 80)
 [44778]
*May  4 15:48:33.015: NAT*: i: tcp (172.16.0.11, 39340) -> (193.136.39.12, 80)
 [44778]
*May  4 15:48:33.019: NAT*: s=172.16.0.11->192.168.56.178, d=193.136.39.12
 [44778]
*May  4 15:48:33.051: NAT*: o: tcp (193.136.39.12, 80) -> (192.168.56.178,
 39340) [0]
*May  4 15:48:33.051: NAT*: s=193.136.39.12, d=192.168.56.178->172.16.0.11 [0]
*May  4 15:48:33.059: NAT*: i: tcp (172.16.0.11, 39340) -> (193.136.39.12, 80)
 [44779]
*May  4 15:48:33.059: NAT*: s=172.16.0.11->192.168.56.178, d=193.136.39.12
 [44779]
*May  4 15:48:33.059: NAT*: i: tcp (172.16.0.11, 39340) -> (193.136.39.12, 80)
 [44780]
*May  4 15:48:33.059: NAT*: s=172.16.0.11->192.168.56.178, d=193.136.39.12
 [44780]
*May  4 15:48:33.075: NAT*: o: tcp (193.136.39.12, 80) -> (192.168.56.178,
 39340) [32364]
*May  4 15:48:33.075: NAT*: s=193.136.39.12, d=192.168.56.178->172.16.0.11
 [32364]
*May  4 15:48:33.291: NAT*: o: tcp (193.136.39.12, 80) -> (192.168.56.178,
 39340) [32365]
*May  4 15:48:33.291: NAT*: s=193.136.39.12, d=192.168.56.178->172.16.0.11
 [32365]
*May  4 15:48:33.291: NAT*: o: tcp (193.136.39.12, 80) -> (192.168.56.178,
 39340) [32366]
*May  4 15:48:33.291: NAT*: s=193.136.39.12, d=192.168.56.178->172.16.0.11
 [32366]
*May  4 15:48:33.295: NAT*: o: tcp (193.136.39.12, 80) -> (192.168.56.178,
 39340) [32367]
*May  4 15:48:33.295: NAT*: s=193.136.39.12, d=192.168.56.178->172.16.0.11
 [32367]
*May  4 15:48:33.295: NAT*: o: tcp (193.136.39.12, 80) -> (192.168.56.178,
 39340) [32368]
*May  4 15:48:33.299: NAT*: s=193.136.39.12, d=192.168.56.178->172.16.0.11
```

```
 [32368]
*May  4 15:48:33.323: NAT*: i: tcp (172.16.0.11, 39340) -> (193.136.39.12, 80)
 [44781]
*May  4 15:48:33.323: NAT*: s=172.16.0.11->192.168.56.178, d=193.136.39.12
 [44781]
*May  4 15:48:33.323: NAT*: i: tcp (172.16.0.11, 39340) -> (193.136.39.12, 80)
 [44782]
*May  4 15:48:33.323: NAT*: s=172.16.0.11->192.168.56.178, d=193.136.39.12
 [44782]
*May  4 15:48:33.323: NAT*: i: tcp (172.16.0.11, 39340) -> (193.136.39.12, 80)
 [44783]
*May  4 15:48:33.323: NAT*: s=172.16.0.11->192.168.56.178, d=193.136.39.12
 [44783]
*May  4 15:48:33.335: NAT*: o: tcp (193.136.39.12, 80) -> (192.168.56.178,
 39340) [32368]
*May  4 15:48:33.335: NAT*: s=193.136.39.12, d=192.168.56.178->172.16.0.11
 [32368]
*May  4 15:48:33.335: NAT*: o: tcp (193.136.39.12, 80) -> (192.168.56.178,
 39340) [32369]
```

...

O trabalho do NAT é realizar uma tradução de endereços de rede privado (no nosso caso, 172.16.0.11), em um endereço público (192.168.56.178). Isto permite que um computador de uma rede interna (privada) tenha acesso ao exterior, rede pública.

3. Quando usamos o comando `clear ip nat translation *`, deixamos de receber os "hello". "Quando usamos NAT com tradução de portas, os fluxos de pacotes passam de *connectionless* para *connection-oriented*", ou seja, o *router* Cisco tem uma tabela de traduções de IPs internos para externos e vice-versa, (mantém informação sobre as conexões).
O comando acima permite-nos limpar todas as traduções dinâmicas da tabela de tradução do *router*, perdendo a conexão, uma vez que apagamos toda a informação da tabela, voltando assim para *connectionless*, sendo por esta razão que deixamos de receber os "hello".

4.

a.

```
[root@Labs5616 ar]# ssh 192.168.56.20
```

b. Os IPs privados numa rede local não são acessíveis fora da rede, a partir da internet. O *port forwarding* serve para permitir que uma dada máquina chegue (usando NAT) a uma porta de um IP privado, mesmo fora dessa rede, este atribui uma porta a um dispositivo.
Usando uma porta não-standard quando temos mais que um terminal numa sub-rede, teríamos de usar portas diferentes para aceder de uma rede externa a uma rede interna, neste caso, ao *router* Linux e ao Terminal 2.

5.

a. Captura *wireshark* nas interfaces do Router Linux:

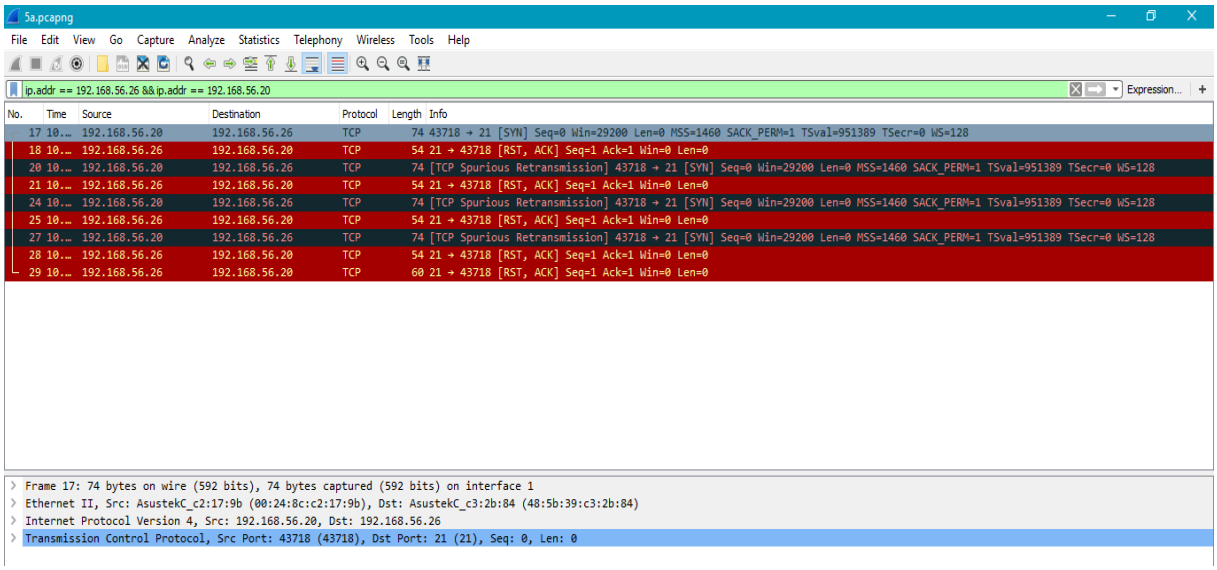

Figura 5: Captura *wireshark* nas interfaces de Router Linux.

Este procedimento não foi bem sucedido pois o *router* Linux não oferece nenhum serviço na porta 21. Se tivéssemos tentado conectar-nos ao 10.0.0.22, o gateway do Terminal 1 não saberia a rota para o IP.
Com o redirecionamento da porta 21 do *router* Linux para Terminal 2, iremos conseguir entregar-lhe pacotes e com MASQUERADING teremos uma comunicação FTP bidirecional efetiva como veremos nas seguintes alíneas.

b.

```
[root@Labs5616 Trab3]# iptables -t nat -A PREROUTING -p tcp -i enp0s7
  -d 192.168.56.26 --dport 21 -j DNAT --to 10.0.0.22:21
[root@Labs5616 Trab3]# iptables -t nat -A POSTROUTING -o enp0s7 -j
  MASQUERADE

[root@Labs5616 Trab3]# iptables-save
# Generated by iptables-save v1.4.21 on Wed May  4 17:54:11 2016
*nat
:PREROUTING ACCEPT [5:626]
:INPUT ACCEPT [1:328]
:OUTPUT ACCEPT [16:1194]
:POSTROUTING ACCEPT [3:234]
-A PREROUTING -d 192.168.56.26/32 -i enp0s7 -p tcp -m tcp --dport 21
  -j DNAT --to-destination 10.0.0.22:21
-A POSTROUTING -o enp0s7 -j MASQUERADE
COMMIT
# Completed on Wed May  4 17:54:11 2016
# Generated by iptables-save v1.4.21 on Wed May  4 17:54:11 2016
*filter
:INPUT ACCEPT [21329:24200039]
```

```
:FORWARD ACCEPT [99770:70460139]
:OUTPUT ACCEPT [13750:876654]
COMMIT
# Completed on Wed May  4 17:54:11 2016
[root@Labs5616 Trab3]#
```

c. Captura *wireshark* na interface enp0s7 - 192.168.56.26 do Router Linux:

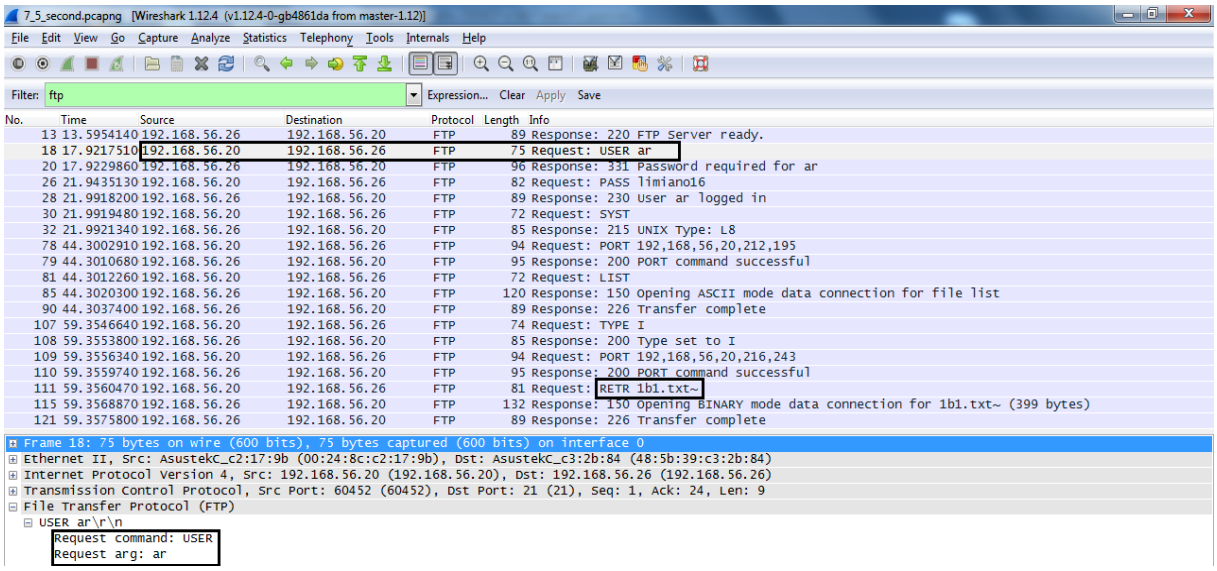

Figura 6: Captura *wireshark* na interface 192.168.56.26.

Captura *wireshark* na interface enp1s6 - 10.0.0.1 do Router Linux:

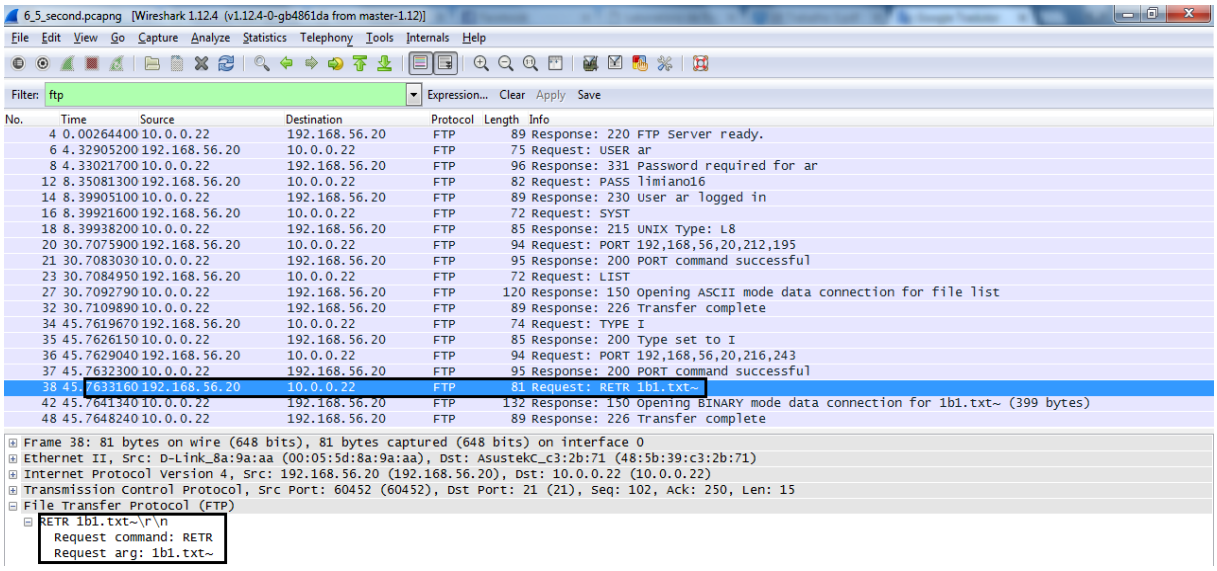

Figura 7: Captura *wireshark* na interface 10.0.0.1.

Com *port-forwarding* (entrada estática na tabela NAT) todos os pacotes para um IP+porta exterior (inside global) são sempre traduzidos para um IP+porta interior (inside local) o que significa que quando o Terminal 1 envia pacotes (ftp) à porta 21 do *router* Linux, estes estão a ser encaminhados para o 10.0.0.22:21 (TERMINAL 2), destino inacessível pelo **Terminal 1**

diretamente. O *router* Linux reescreve os cabeçalhos dos pacotes com destino a 192.168.56.26:21 e origem IP:PORT e encaminha-os para 10.0.0.22 desta vez com cabeçalho destino 10.0.0.22 e a mesma origem.

Como 10.0.0.22 responde a IP:PORT (192.168.56.20) e sabe a rota para esse destino, a comunicação é bem sucedida.

d. Captura *wireshark* nas interfaces do Router Linux:

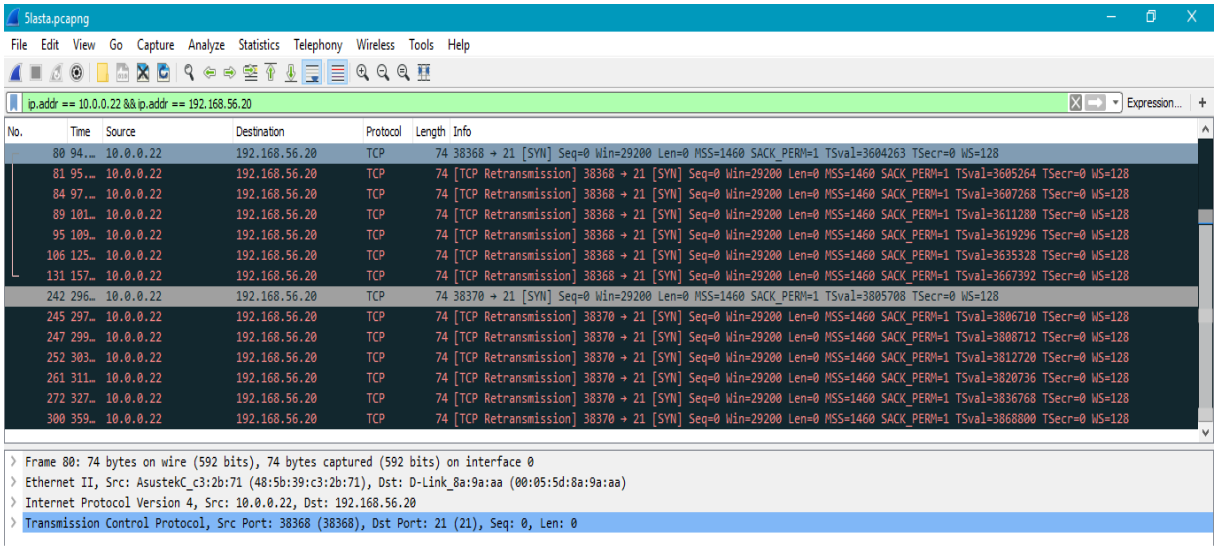

Figura 8: Captura *wireshark* nas interfaces de Router Linux.

A conexão não é bem sucedida pois o *router* Linux limita-se a retransmitir os pacotes do Terminal 2 para o Terminal 1 mantendo os cabeçalhos nos pacotes. Como Terminal 1 não sabe a rota para o endereço do Terminal 2 escrito no cabeçalho (10.0.0.22), a conexão não é bidirecional, logo não é bem sucedida.

e. Com a instalação do módulo `nf_nat_ftp`, o *router* Linux reescreve o cabeçalho dos pacotes que por si passam, permitindo que o Terminal 1 responda ao Terminal 2, através do *router* Linux e uma porta dedicada para esse efeito (neste caso 38402 é a Dst port no Terminal 1 e no *router* Linux).

Como podemos ver pelo dump do Wireshark ordenado temporalmente, do Terminal 2 para o Terminal 1 altera-se o campo *source* para o Terminal 1 conseguir responder; do Terminal 1 para o Terminal 2 altera-se o campo *destination* para o pacote ser aceite pelo Terminal 2.

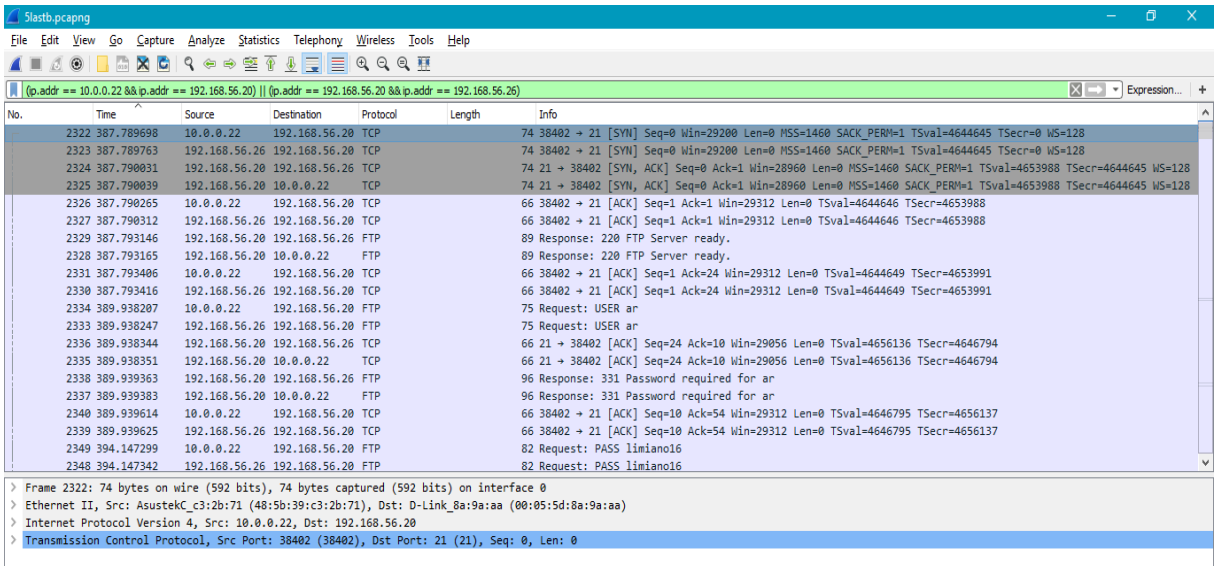

Figura 9: Captura *wireshark* nas interfaces de Router Linux.