

Segundo Trabalho Laboratorial

Redes de Computadores



Mestrado Integrado em Engenharia Informática

Redes de computadores

Grupo:

Carolina Centeio Jorge - up201403090

João Fidalgo - up201303098

Mónica Fernandes - up201404789

Tiago Almeida - up201305665

Faculdade de Engenharia da Universidade do Porto
Rua Roberto Frias, sn, 4200-465 Porto, Portugal

23 de Dezembro de 2016

1 Summary

CHANGE THIS FOR THIS PROJECT This laboratory project was done in the scope of the subject Computer Networks of the Master in Computer Engineering and Informatics course in the school year of 2016/2017. All required knowledge and skills were provided both in the theoretical and practical classes of this subject, giving special attention to the Application Layer and Data Link Layer slides. Also, the laboratory guide was an essential tool for the development, as it contained all the requirements and all the necessary guides for this project.

This project's development was extremely important, since it allowed us to understand better how the theoretical concepts are actually applied to real world problems. We concluded that to successfully transfer files between two machines, we must guarantee the integrity of all transferred data and therefore, the usage of a data link protocol is crucial.

2 Introdução

Este projeto foi desenvolvido no âmbito da disciplina de Redes de Computadores do 3º ano do Mestrado Integrado em Engenharia Informática e Computadores da Faculdade de Engenharia da Universidade do Porto. Este projeto consiste em desenvolver uma aplicação através do protocolo FTP (descrito no RFC959) e realizar um conjunto de experiências com o objetivo de consolidar os conceitos lecionados nas aulas teóricas e perceber como configurar uma rede de computadores. Este projeto foi realizado em ambiente *Linux* e desenvolvido em C.

A aplicação reduz-se a fazer download de um ficheiro, passado como argumento pelo utilizador, de um servidor FTP assim como também é possível especificar o *username* e a *password* se for necessário fazer *login*, com a sintaxe descrita no RFC1738.

O propósito das experiências laboratoriais é dar a conhecer os comandos necessários para conseguir configurar uma rede de computadores. Esta segunda parte tem como objetivo configurar uma rede de computadores, todos ligados entre si através de duas VLANs diferentes configuradas num *switch* assim como também configurar um *router* comercial com *NAT* e utilizar *DNS* para a conversão de *hostnames* em *IP Addresses*.

O relatório tem uma secção dedicada a cada uma das duas partes deste projeto. O desenvolvimento da aplicação será explicado com grande detalhe na secção 3 e na secção 4 teremos subsecções para cada experiência realizada.

3 Aplicação de Download

Foi elaborado uma aplicação em C para fazer download de um ficheiro de um servidor FTP, usando ligações TCP. Esta aplicação implementa o protocolo FTP, como descrito em RFC959.

Esta aplicação é de um só argumento, sendo este um URL que segue a sintaxe descrita em RFC1738:

```
ftp://[<user>:>password>@]<host>/<url-path>
```

A aplicação começa por fazer parse desse URL, de forma a extrair a informação necessária para estabelecer a ligação TCP (user, password e host) e para, posteriormente, fazer download do ficheiro url-path. O user e password podem ser omitidos, sendo assumidos como “anonymous” e “mail@domain”, respetivamente. Seguidamente, usando a função *getip*, vamos buscar o ip correspondente ao endereço host escolhido pelo utilizador. Assim, já podemos abrir um socket TCP (chamemos-lhe A), usando a porta 21 e conectar ao servidor, através da função *ftpConnect*.

Se a ligação for bem sucedida, é enviado o user seguido da password em *ftpLogin*. Se a resposta aos comandos de login for, também, positiva, tenta-se estabelecer o Passive Mode com a função *ftpPasv*. Esta função, se não falhar, retorna a porta que vai servir para abrir o novo socket TCP (chamemos-lhe B). Neste novo socket, B, vai ser recebido o ficheiro que queremos fazer download, assim que, no primeiro socket aberto, A, enviarmos o comando “retr”, através da função *ftpDownload*. A *ftpDownload*,

lê também o que vai sendo recebido no socket B e guarda num ficheiro, de forma a reproduzir inteiramente o ficheiro que queremos transferir. No final, a ligação é terminada com o comando “quit” e são também fechados os sockets A e B e liberto o espaço em memória.

4 Experiências Laboratoriais

4.1 Configuração de um IP de Rede

Esta experiência permitiu-nos saber como distinguir os diferentes pacotes de dados assim como a sua funcionalidade e também como estabelecer ligados entre dois computadores.

Um dos pacotes de dados usados nesta experiência são os pacotes ARP (*Address Resolution Protocol*). Este pacotes servem para obter o *MAC Address* (endereço que identifica um computador na rede) de um determinado *IP Address* (endereço que distingue um computador na rede). O computador onde é executado o comando *ping* envia um comando para todos os computadores com que tem ligação e pergunta que computador tem o um determinado *IP Address*. Este primeiro pacote envia o *IP Address* do computador que se pretende descobrir o *MAC Address* assim como o *IP Address* do computador para o qual se deve responder. Um segundo pacote ARP é enviado como resposta com o *IP Address* e o *MAC Address* do computador em questão. Este mecanismo pode ser visto no anexo 7.1 nas linhas 22 e 23.

Outro tipo de pacotes são os ICMP (*Internet Control Message Protocol*). Estes pacotes são gerados pelo comando ping e servem para enviar mensagens de erro ou de controlo para outros *hosts* ou *routers*. Quando um computador executa um comando ping, este envia um *echo request* para o outro computador, com o seu *IP Address* e o seu *MAC Address* assim como o *IP Address* e o *MAC Address* do computador de destino. Por sua vez, o computador de destino envia uma resposta (*echo reply*), com as mesmas informações. Este mecanismo pode ser confirmado no anexo 7.1 nas linhas 24 e 25.

Para decodificar estes pacotes é necessário analisar o *IP Diagram Format* descrito nas aulas teóricas. O campo *type of service* distingue se o pacote é ARP (0x0806) ou ICMP (0x0800). Isto pode ser visto no anexo 7.1 na segunda e terceira separação horizontal. O campo *length* representa o tamanho total do *datagram*. Este campo vem imediatamente a seguir ao campo *type of service* e como podemos na zona mencionada acima este campo tem 0x0054 como valor, o que representa 84 bytes.

O mecanismo de *loopback* é um mecanismo que reenvia uma mensagem de volta para o computador que a envia. Isto permite detetar erros na transmissão de dados assim como problemas nos cabos de transmissão.

4.2 Implementação de duas *Virtual LANS* num switch

Nesta experiência foi solicitada a criação de duas VLANs no switch. Para configurar uma VLAN temos de executar o comando “vlan x”, em que x corresponde ao número da vlan, na consola do switch. Depois de configurar as VLANs, foi necessário adicionar as respetivas portas no switch. Para adicionar a porta, basta selecioná-la e usar o comando “interface fastethernet 0/y”, em que y corresponde ao número da porta que queremos adicionar. Antes de adicionar a porta à VLAN com o comando “switchport access vlan x”, devemos alterar o seu modo com o comando “switchport mode access”.

Depois da observação dos logs da experiência 2, podemos concluir que existem 2 broadcast domains, um em cada VLAN. Isto porque foi visto que quando se fazia broadcast no tux1 apenas os tuxs dessa mesma VLAN recebiam os pacotes. Embora os tuxs estejam todos no mesmo switch, os pacotes só eram recebidos por tuxs dentro da mesma VLAN.

4.3 Configurar um *router* em Linux

“ARP messages” servem para informar o tux4 quais são os endereços MAC associados ao IP de destino. No tux1 temos a rota da interface eth0 que é criada quando é atribuído o IP a essa interface e uma rota para a rede 172.16.Y0.1 (Y representa o número da bancada) através da gateway 172.16.Y0.254.

No tux2 temos a rota da interface eth0 e uma rota para a rede 172.16.Y0.1 através da gateway 172.16.Y1.253.

No tux4 temos as rotas das interfaces eth0 e eth1, com este tux conseguimos alcançar as redes 172.16.Y0.1 e 172.16.Y1.1.

A tabela de forwarding contém a informação da rede de destino, a gateway por onde acedemos, a máscara dessa rede e a interface associada.

Visto que os ICMP packets têm sempre o mesmo IP de fonte e destino, estes são observados para quando passarem pelos vários tuxs saberem para onde têm de reencaminhar os pacotes.

4.4 Configurar um router comercial e implementar NAT

O objetivo desta experiência era configurar um *router* comercial com NAT (*Network Address Resolution*) e perceber a sua funcionalidade.

A primeira parte desta experiência consiste em configurar o *router* sem NAT. Para isso, configuramos cada uma das interfaces *gigabitethernet* com os seguintes comandos:

- *ip address [ip address] [mascara de rede]*
- *no shutdown*

e configuramos as rotas com o comando *ip route [rede de destino] [mascara de rede] [gateway IP]*. Com isto, foi possível configurar cada uma das interfaces de forma a que cada computador na rede privada tivesse acesso ao *router* comercial. Neste momento da experiência percebemos que os pacotes do comando *ping* entre o computador 2 e o computador 1 percorriam o seguinte caminho tux2 ->tux4 ->tux1. Isto acontecia, porque o tux2 tinha uma route para a outra VLAN a partir do tux4. Removendo essa rota, o caminho passa a ser tux2->router ->tux4 ->tux1. Sem *redirects*, este caminho é percorrido todas as vezes que é enviado um pacote, no entanto, se ativarmos os *redirects* este caminho é apenas percorrido uma vez e os seguintes já percorrem a rota tux2 ->tux4 ->tux1.

A segunda parte consiste em adicionar NAT ao router para que esta rede privada possa ter acesso a uma rede na *internet*. A NAT é uma funcionalidade que permite que o endereço de um computador numa rede privada seja traduzido num *IP Address* válido na *internet*. A NAT recebe o *IP Address* interno e a porta local do computador e gera um endereço de 16 bits usando uma tabela hash, sendo este endereço então escrito no campo da porta de origem. Com este mecanismo, o *router* é capaz de enviar pacotes de dados para o exterior da rede privada a partir de um *IP Address* global e do *IP Address* gerado com o mecanismo descrito anteriormente e quando recebe uma resposta, consegue descobrir qual será o computador que terá de receber a resposta.

Para configure a NAT no *router* basta adicionar *ip nat inside* à interface que está ligada à rede privada e *ip nat outside* à interface ligado à rede externa.

4.5 DNS

Com esta experiência percebemos como configurar o servidor *DNS* assim como o seu papel e a sua importância nas redes de computadores.

De uma forma breve, o servidor *DNS* serve para traduzir um *hostname* (www.google.com) num *IP Address* (194.210.238.155) atrás do *DNS Resolver*.

A configuração do servidor *DNS* trata-se apenas da edição do ficheiro */etc/resolv.conf*, adicionando uma linha a especificar o *search* (netlab.fe.up.pt) e o *nameserver* (172.16.1.1).

O primeiro pacote é enviado ao servidor *DNS* com o nome do *host* que por sua vez é passado como argumento ao *resolver* e este devolve o *IP Address* ligado a esse *hostname*. O segundo pacote envia um *IP Address* e espera pelo respetivo *hostname*. Este último mecanismo é chamado *reverse DNS lookup*.

4.6 Ligações TCP

Esta experiência pretende testar e analisar resultados relativos à aplicação FTP que elaborámos. A aplicação, como analisado anteriormente, estabelece duas ligações TCP. A primeira, através do socket a que chamámos A, onde é transportada a informação de controlo FTP através do comando “pasv”.

A ligação TCP divide-se em três fases e em todas é possível analisar um conjunto de parâmetros como a source port e a destination port (as mesmas durante toda a aplicação), o ARQ (por flag, neste caso ACK ou SYN), o sequence number, o Acknowledgment number e o window size. O mecanismo de ARQ em TCP funciona como Selective Repeat ARQ: o emissor envia o número de frames equivalente ao window size sem esperar por um ACK. Este window size é determinado pelo recetor, enquanto que a congestion window é determinada pelo emissor e não é visível nos logs. O recetor pode rejeitar frames individualmente, que posteriormente serão reenviados.

192.168.50.138	192.168.50.138	TCP	94 [TCP Dup ACK 2827899] 60164→21324 [ACK] Seq=1 Ack=4357033 Win=523648 Len=0 TSval=1353767 TSecr=1022560637
192.168.50.138	172.16.30.1	FTP-DA.	2962 FTP Data: 2896 bytes
172.16.30.1	192.168.50.138	TCP	94 [TCP Dup ACK 2827899] 60164→21324 [ACK] Seq=1 Ack=4357033 Win=523648 Len=0 TSval=1353767 TSecr=1022560637
192.168.50.138	172.16.30.1	FTP-DA.	2962 FTP Data: 2896 bytes
192.168.50.138	192.168.50.138	TCP	94 [TCP Dup ACK 2827899] 60164→21324 [ACK] Seq=1 Ack=4357033 Win=523648 Len=0 TSval=1353767 TSecr=1022560637
192.168.50.138	172.16.30.1	FTP-DA.	2962 [TCP Previous segment not captured] FTP Data: 2896 bytes
192.168.50.138	192.168.50.138	TCP	94 [TCP Dup ACK 2827899] 60164→21324 [ACK] Seq=1 Ack=4357033 Win=523648 Len=0 TSval=1353767 TSecr=1022560637
192.168.50.138	172.16.30.1	FTP-DA.	2962 [TCP Fast Retransmission] FTP Data: 2896 bytes
172.16.30.1	192.168.50.138	TCP	94 60164→21324 [ACK] Seq=1 Ack=4403369 Win=477312 Len=0 TSval=1353767 TSecr=1022560643 SLE=4636497 SRE=463939
192.168.50.138	172.16.30.1	FTP-DA.	2962 [TCP Previous segment not captured] FTP Data: 2896 bytes
172.16.30.1	192.168.50.138	TCP	94 [TCP Dup ACK 283181] 60164→21324 [ACK] Seq=1 Ack=4403369 Win=477312 Len=0 TSval=1353767 TSecr=1022560643 S
192.168.50.138	172.16.30.1	FTP-DA.	2962 FTP Data: 2896 bytes
172.16.30.1	192.168.50.138	TCP	94 [TCP Dup ACK 303182] 60164→21324 [ACK] Seq=1 Ack=4403369 Win=477312 Len=0 TSval=1353767 TSecr=1022560643 S

Aqui, o parâmetro ACK terá sido menor que o último byte enviado pelo emissor. O recetor detetou a falha na sequência de números e gerou um ACK duplicado para pacote subsequente recebido nessa ligação, até que o pacote em falta tenha sido recebido com sucesso.

As três fases são, então:

- Estabelecimento de ligação: desde a abertura do socket, ao processamento dos dados de user e password

192.168.50.138	192.168.50.138	TCP	74 38494→21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1353615 TSecr=0 WS=128
192.168.50.138	172.16.30.1	TCP	74 21→38494 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1022560497 TSecr=1353615 WS=128
172.16.30.1	192.168.50.138	TCP	66 38494→21 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1353616 TSecr=1022560497
192.168.50.138	172.16.30.1	FTP	101 Response: 220 FTP for Alf/Tom/Crazy/Penguin
172.16.30.1	192.168.50.138	TCP	66 38494→21 [ACK] Seq=1 Ack=36 Win=29312 Len=0 TSval=1353617 TSecr=1022560498
172.16.30.1	192.168.50.138	FTP	83 Request: USER up081403000
192.168.50.138	172.16.30.1	TCP	66 21→38494 [ACK] Seq=36 Ack=18 Win=29056 Len=0 TSval=1022560499 TSecr=1353617
192.168.50.138	172.16.30.1	FTP	100 Response: 331 Please specify the password.
172.16.30.1	192.168.50.138	FTP	82 Request: PASS
192.168.50.138	172.16.30.1	TCP	66 21→38494 [ACK] Seq=70 Ack=34 Win=29056 Len=0 TSval=1022560509 TSecr=1353618
192.168.50.138	172.16.30.1	FTP	89 Response: 230 Login successful.
172.16.30.1	192.168.50.138	FTP	71 Request: pasv
192.168.50.138	172.16.30.1	TCP	66 21→38494 [ACK] Seq=93 Ack=39 Win=29056 Len=0 TSval=1022560541 TSecr=1353660
192.168.50.138	172.16.30.1	FTP	117 Response: 227 Entering Passive Mode (192,168,50,138,83,76)
172.16.30.1	192.168.50.138	TCP	74 60164→21324 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1353660 TSecr=0 WS=128
192.168.50.138	172.16.30.1	TCP	74 21324→60164 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1022560542 TSecr=1353660 WS=1

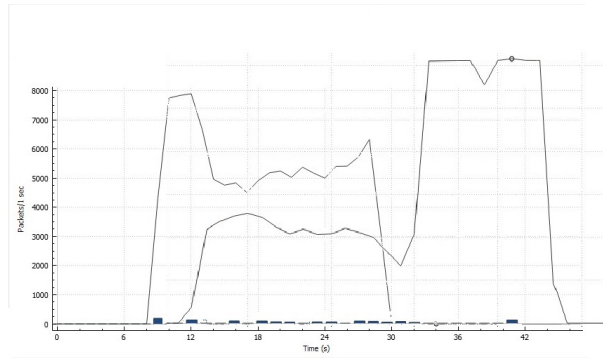
- Transferência de dados: desde o envio do comando “retr” até ao final da transferência do ficheiro em causa.

192.168.50.138	172.16.30.1	FTP-DA.	2962 FTP Data: 2896 bytes
172.16.30.1	192.168.50.138	TCP	66 60164→21324 [ACK] Seq=1 Ack=4345 Win=37888 Len=0 TSval=1353667 TSecr=1022560548
192.168.50.138	172.16.30.1	FTP-DA.	2962 FTP Data: 2896 bytes
172.16.30.1	192.168.50.138	TCP	66 60164→21324 [ACK] Seq=1 Ack=7241 Win=43776 Len=0 TSval=1353667 TSecr=1022560548
192.168.50.138	172.16.30.1	FTP-DA.	4410 FTP Data: 4344 bytes
192.168.50.138	172.16.30.1	FTP-DA.	2962 FTP Data: 2896 bytes
192.168.50.138	192.168.50.138	TCP	66 60164→21324 [ACK] Seq=1 Ack=14481 Win=58240 Len=0 TSval=1353668 TSecr=1022560548
192.168.50.138	172.16.30.1	FTP-DA.	1514 FTP Data: 1440 bytes
172.16.30.1	192.168.50.138	TCP	66 60164→21324 [ACK] Seq=1 Ack=15929 Win=61056 Len=0 TSval=1353668 TSecr=1022560549
192.168.50.138	172.16.30.1	FTP-DA.	4410 FTP Data: 4344 bytes
172.16.30.1	192.168.50.138	TCP	66 60164→21324 [ACK] Seq=1 Ack=20273 Win=69760 Len=0 TSval=1353668 TSecr=1022560549
192.168.50.138	172.16.30.1	FTP-DA.	2962 FTP Data: 2896 bytes
172.16.30.1	192.168.50.138	TCP	66 60164→21324 [ACK] Seq=1 Ack=23169 Win=75648 Len=0 TSval=1353668 TSecr=1022560549

- Término de ligação: o envio do quit seguido da receção do “Goodbye” termina a ligação

1154.. 29.523995	172.16.30.1	192.168.50.138	FTP	71 Request: quit
1154.. 29.524003	172.16.30.1	192.168.50.138	FTP-DA.	71 FTP Data: 5 bytes
1154.. 29.524012	172.16.30.1	192.168.50.138	TCP	66 38494→21 [FIN, ACK] Seq=70 Ack=253 Win=29312 Len=0 TSval=1358687 TSecr=1022565567
1154.. 29.524025	172.16.30.1	192.168.50.138	TCP	66 60164→21324 [FIN, ACK] Seq=70 Ack=172389436 Win=1130800 Len=0 TSval=1358687 TSecr=1022565552
1154.. 29.527624	192.168.50.138	172.16.30.1	TCP	60 21324→60164 [RST] Seq=172389436 Win=0 Len=0
1154.. 29.527637	192.168.50.138	172.16.30.1	TCP	60 21324→60164 [RST] Seq=172389436 Win=0 Len=0
1154.. 29.528259	192.168.50.138	172.16.30.1	FTP	80 Response: 221 Goodbye.
1154.. 29.528259	172.16.30.1	192.168.50.138	TCP	66 38494→21 [RST] Seq=71 Win=0 Len=0
1154.. 29.528880	192.168.50.138	172.16.30.1	TCP	66 21→38494 [FIN, ACK] Seq=267 Ack=71 Win=29056 Len=0 TSval=1022565568 TSecr=1358687
1154.. 29.528891	172.16.30.1	192.168.50.138	TCP	64 38494→21 [RST] Seq=71 Win=0 Len=0

É notável que o número de pacotes transferidos com sucesso, por segundo, diminui quando outra ligação TCP é iniciada. Nesta segunda, notamos também um aumento significativo quando a primeira transferência acaba. Tentámos juntar num só gráfico o que acabámos de explicar:



5 Conclusão

Todas as experiências foram realizadas com sucesso e com elas fomos capazes de aplicar os conceitos dados nas aulas teóricas.

Conclui-se que a configuração de uma rede de pequena dimensão é um problema sem grande complexidade, no entanto para a configuração de uma rede de maior dimensão é necessário aprofundar certos conceitos e aprender novos que não foram lecionados nesta unidade curricular.

Falar das conclusões da experiência 6, alterações no tráfego quando duas aplicações estão a ser corridas ao mesmo tempo, etc....

6 Contribuições

7 Anexos

7.1 Experiência 1

No.	Time	Source	Destination	Protocol	Length	Info
16	24.057821	CiscoInc_3a:fa:83	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/30/fc:fb:3a:fa:80 Cost = 0 Port = 0x8003
17	25.397218	CiscoInc_3a:fa:83	CDP/VTP/DTP/PAGP/UD...	CDP	435	Device ID: tux-sw3 Port ID: FastEthernet0/1
18	26.067802	CiscoInc_3a:fa:83	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/30/fc:fb:3a:fa:80 Cost = 0 Port = 0x8003
19	28.067858	CiscoInc_3a:fa:83	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/30/fc:fb:3a:fa:80 Cost = 0 Port = 0x8003
20	30.072230	CiscoInc_3a:fa:83	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/30/fc:fb:3a:fa:80 Cost = 0 Port = 0x8003
21	30.324124	CiscoInc_3a:fa:83	CiscoInc_3a:fa:83	LOOP	60	Reply
22	30.667221	G-ProCom_8b:e4:4d	Broadcast	ARP	42	Who has 172.16.30.254? Tell 172.16.30.1
23	30.667564	HewlettP_5a:7d:74	G-ProCom_8b:e4:4d	ARP	60	172.16.30.254 is at 00:21:5a:5a:7d:74
24	30.667574	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x2831, seq=1/256, ttl=64 (reply in 25)
25	30.667828	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x2831, seq=1/256, ttl=64 (request in 24)
26	31.666225	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x2831, seq=2/512, ttl=64 (reply in 27)
27	31.666568	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x2831, seq=2/512, ttl=64 (request in 26)
28	32.082024	CiscoInc_3a:fa:83	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/30/fc:fb:3a:fa:80 Cost = 0 Port = 0x8003
29	32.665221	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x2831, seq=3/768, ttl=64 (reply in 30)
30	32.665221	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x2831, seq=3/768, ttl=64 (request in 29)
[Coloring Rule String: icmp icmpv6]						
▼ Ethernet II, Src: G-ProCom_8b:e4:4d (00:0f:fe:8b:e4:4d), Dst: HewlettP_5a:7d:74 (00:21:5a:5a:7d:74)						
▼ Destination: HewlettP_5a:7d:74 (00:21:5a:5a:7d:74)						
Address: HewlettP_5a:7d:74 (00:21:5a:5a:7d:74)						
.....0..... = LG bit: Globally unique address (factory default)						
.....0..... = IG bit: Individual address (unicast)						
▼ Source: G-ProCom_8b:e4:4d (00:0f:fe:8b:e4:4d)						
Address: G-ProCom_8b:e4:4d (00:0f:fe:8b:e4:4d)						
.....0..... = LG bit: Globally unique address (factory default)						
.....0..... = IG bit: Individual address (unicast)						
Type: IPv4 (0x0800)						
0000	00 21 5a 7d 74 00 0f fe 8b e4 4d 08 00 45 00	.1.22)t...E				
0010	00 54 7f fb 40 00 40 01 25 8e ac 10 1e 01 ac 10	.T..0.0.%.....				
0020	1e fe 08 00 0e 47 28 31 00 01 84 d3 2d 58 91 57	...G(1....-X.W				
0030	03 00 08 00 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15				
0040	16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25!*%\$				
0050	26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35	&'()*+,-./012345				
0060	36 37	67				

7.2 Experiência 4

7.2.1 Passo 4 - Sem Redirect

No.	Time	Source	Destination	Protocol	Length	Info
5	6.014810	CiscoInc.3a:fa:84	Spanning-tree-(for...	STP	60	Conf. Root = 32768/31/fc:fb:3a:fa:80 Cost = 0 Port = 0x8004
6	7.483499	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x1894, seq=1/256, ttl=64 (reply in 8)
7	7.483881	172.16.31.254	172.16.31.1	ICMP	70	Redirect (Redirect for host)
8	7.484159	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) reply id=0x1894, seq=1/256, ttl=63 (request in 6)
9	8.019793	CiscoInc.3a:fa:84	Spanning-tree-(for...	STP	60	Conf. Root = 32768/31/fc:fb:3a:fa:80 Cost = 0 Port = 0x8004
10	8.482504	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x1894, seq=2/512, ttl=64 (reply in 12)
11	8.482815	172.16.31.254	172.16.31.1	ICMP	70	Redirect (Redirect for host)
12	8.483011	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) reply id=0x1894, seq=2/512, ttl=63 (request in 10)
13	9.481501	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x1894, seq=3/768, ttl=64 (reply in 15)
14	9.481813	172.16.31.254	172.16.31.1	ICMP	70	Redirect (Redirect for host)
15	9.482088	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) reply id=0x1894, seq=3/768, ttl=63 (request in 13)
16	10.024038	CiscoInc.3a:fa:84	Spanning-tree-(for...	STP	60	Conf. Root = 32768/31/fc:fb:3a:fa:80 Cost = 0 Port = 0x8004
17	10.481409	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x1894, seq=4/1024, ttl=64 (reply in 19)
18	10.481714	172.16.31.254	172.16.31.1	ICMP	70	Redirect (Redirect for host)
19	10.482103	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) reply id=0x1894, seq=4/1024, ttl=63 (request in 17)
20	11.481390	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x1894, seq=5/1280, ttl=64 (reply in 22)
21	11.481692	172.16.31.254	172.16.31.1	ICMP	70	Redirect (Redirect for host)
22	11.482006	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) reply id=0x1894, seq=5/1280, ttl=63 (request in 20)
23	12.029565	CiscoInc.3a:fa:84	Spanning-tree-(for...	STP	60	Conf. Root = 32768/31/fc:fb:3a:fa:80 Cost = 0 Port = 0x8004
24	12.481392	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x1894, seq=6/1536, ttl=64 (reply in 26)
25	12.481690	172.16.31.254	172.16.31.1	ICMP	70	Redirect (Redirect for host)
26	13.485008	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) reply id=0x1894, seq=6/1536, ttl=63 (request in 24)

7.2.2 Passo 4 - Com Redirect

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	CiscoInc.3a:fa:84	Spanning-tree-(for...	STP	60	Conf. Root = 32768/31/fc:fb:3a:fa:80 Cost = 0 Port = 0x8004
2	0.461789	CiscoInc.3a:fa:84	CiscoInc.3a:fa:84	LOOP	60	Reply
3	1.258621	CiscoInc.3a:fa:84	CDP/VTP/DTP/PagP/UD...	CDP	435	Device ID: tux-sw3 Port ID: FastEthernet0/2
4	2.004965	CiscoInc.3a:fa:84	Spanning-tree-(for...	STP	60	Conf. Root = 32768/31/fc:fb:3a:fa:80 Cost = 0 Port = 0x8004
5	4.000905	CiscoInc.3a:fa:84	Spanning-tree-(for...	STP	60	Conf. Root = 32768/31/fc:fb:3a:fa:80 Cost = 0 Port = 0x8004
6	5.626150	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x1a46, seq=1/256, ttl=64 (reply in 8)
7	5.626548	172.16.31.254	172.16.31.1	ICMP	70	Redirect (Redirect for host)
8	5.626973	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) reply id=0x1a46, seq=1/256, ttl=63 (request in 6)
9	6.014825	CiscoInc.3a:fa:84	Spanning-tree-(for...	STP	60	Conf. Root = 32768/31/fc:fb:3a:fa:80 Cost = 0 Port = 0x8004
10	6.625159	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x1a46, seq=2/512, ttl=64 (reply in 11)
11	6.625613	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) reply id=0x1a46, seq=2/512, ttl=63 (request in 10)
12	7.624159	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x1a46, seq=3/768, ttl=64 (reply in 13)
13	7.624569	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) reply id=0x1a46, seq=3/768, ttl=63 (request in 12)
14	8.019652	CiscoInc.3a:fa:84	Spanning-tree-(for...	STP	60	Conf. Root = 32768/31/fc:fb:3a:fa:80 Cost = 0 Port = 0x8004
15	8.624117	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x1a46, seq=4/1024, ttl=64 (reply in 16)
16	8.624557	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) reply id=0x1a46, seq=4/1024, ttl=63 (request in 15)
17	9.624106	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x1a46, seq=5/1280, ttl=64 (reply in 18)
18	9.624520	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) reply id=0x1a46, seq=5/1280, ttl=63 (request in 17)

7.2.3 Passo 5 - NAT

No.	Time	Source	Destination	Protocol	Length	Info
7	9.305644	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x0d47, seq=1/256, ttl=64 (reply in 8)
8	9.306768	172.16.1.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0d47, seq=1/256, ttl=62 (request in 7)
9	10.029111	CiscoInc.3a:fa:83	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/fc:fb:3a:fa:80 Cost = 0 Port = 0x8003
10	10.306881	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x0d47, seq=2/512, ttl=64 (reply in 11)
11	10.307699	172.16.1.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0d47, seq=2/512, ttl=62 (request in 10)
12	11.306431	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x0d47, seq=3/768, ttl=64 (reply in 13)
13	11.307232	172.16.1.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0d47, seq=3/768, ttl=62 (request in 12)
14	12.028879	CiscoInc.3a:fa:83	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/fc:fb:3a:fa:80 Cost = 0 Port = 0x8003
15	12.306440	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x0d47, seq=4/1024, ttl=64 (reply in 16)
16	12.307241	172.16.1.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0d47, seq=4/1024, ttl=62 (request in 15)
17	13.306430	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x0d47, seq=5/1280, ttl=64 (reply in 18)
18	13.307242	172.16.1.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0d47, seq=5/1280, ttl=62 (request in 17)
19	14.033085	CiscoInc.3a:fa:83	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/fc:fb:3a:fa:80 Cost = 0 Port = 0x8003
20	14.306454	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x0d47, seq=6/1536, ttl=64 (reply in 21)
21	14.307276	172.16.1.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0d47, seq=6/1536, ttl=62 (request in 20)
22	14.316878	HewlettP.5a:7d:74	G-ProCom.8b:e4:4d	ARP	60	Who has 172.16.30.1? Tell 172.16.30.254
23	14.316890	G-ProCom.8b:e4:4d	HewlettP.5a:7d:74	ARP	42	172.16.30.1 is at 00:0f:fe:8b:e4:4d
24	15.306435	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x0d47, seq=7/1792, ttl=64 (reply in 25)
25	15.307256	172.16.1.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0d47, seq=7/1792, ttl=62 (request in 24)

7.3 Experiência 5

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.20.1	172.16.20.1	DNS	471	Standard query 0x0fbc A google.pt
2	0.001431	172.16.1.1	172.16.20.1	DNS	471	Standard query response 0x0fbc A google.pt A 194.210.238.155 A 194.210.238.159 A 194.210.238.163 A 194.210.238.165 A 194.210.238.166 A 194.210.238.167 A 194.210.238.168 A 194.210.238.169 A 194.210.238.170 A 194.210.238.171 A 194.210.238.172 A 194.210.238.173 A 194.210.238.174 A 194.210.238.175 A 194.210.238.176 A 194.210.238.177 A 194.210.238.178 A 194.210.238.179 A 194.210.238.180 A 194.210.238.181 A 194.210.238.182 A 194.210.238.183 A 194.210.238.184 A 194.210.238.185 A 194.210.238.186 A 194.210.238.187 A 194.210.238.188 A 194.210.238.189 A 194.210.238.190 A 194.210.238.191 A 194.210.238.192 A 194.210.238.193 A 194.210.238.194 A 194.210.238.195 A 194.210.238.196 A 194.210.238.197 A 194.210.238.198 A 194.210.238.199 A 194.210.238.200 A 194.210.238.201 A 194.210.238.202 A 194.210.238.203 A 194.210.238.204 A 194.210.238.205 A 194.210.238.206 A 194.210.238.207 A 194.210.238.208 A 194.210.238.209 A 194.210.238.210 A 194.210.238.211 A 194.210.238.212 A 194.210.238.213 A 194.210.238.214 A 194.210.238.215 A 194.210.238.216 A 194.210.238.217 A 194.210.238.218 A 194.210.238.219 A 194.210.238.220 A 194.210.238.221 A 194.210.238.222 A 194.210.238.223 A 194.210.238.224 A 194.210.238.225 A 194.210.238.226 A 194.210.238.227 A 194.210.238.228 A 194.210.238.229 A 194.210.238.230 A 194.210.238.231 A 194.210.238.232 A 194.210.238.233 A 194.210.238.234 A 194.210.238.235 A 194.210.238.236 A 194.210.238.237 A 194.210.238.238 A 194.210.238.239 A 194.210.238.240 A 194.210.238.241 A 194.210.238.242 A 194.210.238.243 A 194.210.238.244 A 194.210.238.245 A 194.210.238.246 A 194.210.238.247 A 194.210.238.248 A 194.210.238.249 A 194.210.238.250 A 194.210.238.251 A 194.210.238.252 A 194.210.238.253 A 194.210.238.254 A 194.210.238.255 A 194.210.238.256 A 194.210.238.257 A 194.210.238.258 A 194.210.238.259 A 194.210.238.260 A 194.210.238.261 A 194.210.238.262 A 194.210.238.263 A 194.210.238.264 A 194.210.238.265 A 194.210.238.266 A 194.210.238.267 A 194.210.238.268 A 194.210.238.269 A 194.210.238.270 A 194.210.238.271 A 194.210.238.272 A 194.210.238.273 A 194.210.238.274 A 194.210.238.275 A 194.210.238.276 A 194.210.238.277 A 194.210.238.278 A 194.210.238.279 A 194.210.238.280 A 194.210.238.281 A 194.210.238.282 A 194.210.238.283 A 194.210.238.284 A 194.210.238.285 A 194.210.238.286 A 194.210.238.287 A 194.210.238.288 A 194.210.238.289 A 194.210.238.290 A 194.210.238.291 A 194.210.238.292 A 194.210.238.293 A 194.210.238.294 A 194.210.238.295 A 194.210.238.296 A 194.210.238.297 A 194.210.238.298 A 194.210.238.299 A 194.210.238.300 A 194.210.238.301 A 194.210.238.302 A 194.210.238.303 A 194.210.238.304 A 194.210.238.305 A 194.210.238.306 A 194.210.238.307 A 194.210.238.308 A 194.210.238.309 A 194.210.238.310 A 194.210.238.311 A 194.210.238.312 A 194.210.238.313 A 194.210.238.314 A 194.210.238.315 A 194.210.238.316 A 194.210.238.317 A 194.210.238.318 A 194.210.238.319 A 194.210.238.320 A 194.210.238.321 A 194.210.238.322 A 194.210.238.323 A 194.210.238.324 A 194.210.238.325 A 194.210.238.326 A 194.210.238.327 A 194.210.238.328 A 194.210.238.329 A 194.210.238.330 A 194.210.238.331 A 194.210.238.332 A 194.210.238.333 A 194.210.238.334 A 194.210.238.335 A 194.210.238.336 A 194.210.238.337 A 194.210.238.338 A 194.210.238.339 A 194.210.238.340 A 194.210.238.341 A 194.210.238.342 A 194.210.238.343 A 194.210.238.344 A 194.210.238.345 A 194.210.238.346 A 194.210.238.347 A 194.210.238.348 A 194.210.238.349 A 194.210.238.350 A 194.210.238.351 A 194.210.238.352 A 194.210.238.353 A 194.210.238.354 A 194.210.238.355 A 194.210.238.356 A 194.210.238.357 A 194.210.238.358 A 194.210.238.359 A 194.210.238.360 A 194.210.238.361 A 194.210.238.362 A 194.210.238.363 A 194.210.238.364 A 194.210.238.365 A 194.210.238.366 A 194.210.238.367 A 194.210.238.368 A 194.210.238.369 A 194.210.238.370 A 194.210.238.371 A 194.210.238.372 A 194.210.238.373 A 194.210.238.374 A 194.210.238.375 A 194.210.238.376 A 194.210.238.377 A 194.210.238.378 A 194.210.238.379 A 194.210.238.380 A 194.210.238.381 A 194.210.238.382 A 194.210.238.383 A 194.210.238.384 A 194.210.238.385 A 194.210.238.386 A 194.210.238.387 A 194.210.238.388 A 194.210.238.389 A 194.210.238.390 A 194.210.238.391 A 194.210.238.392 A 194.210.238.393 A 194.210.238.394 A 194.210.238.395 A 194.210.238.396 A 194.210.238.397 A 194.210.238.398 A 194.210.238.399 A 194.210.238.400 A 194.210.238.401 A 194.210.238.402 A 194.210.238.403 A 194.210.238.404 A 194.210.238.405 A 194.210.238.406 A 194.210.238.407 A 194.210.238.408 A 194.210.238.409 A 194.210.238.410 A 194.210.238.411 A 194.210.238.412 A 194.210.238.413 A 194.210.238.414 A 194.210.238.415 A 194.210.238.416 A 194.210.238.417 A 194.210.238.418 A 194.210.238.419 A 194.210.238.420 A 194.210.238.421 A 194.210.238.422 A 194.210.238.423 A 194.210.238.424 A 194.210.238.425 A 194.210.238.426 A 194.210.238.427 A 194.210.238.428 A 194.210.238.429 A 194.210.238.430 A 194.210.238.431 A 194.210.238.432 A 194.210.238.433 A 194.210.238.434 A 194.210.238.435 A 194.210.238.436 A 194.210.238.437 A 194.210.238.438 A 194.210.238.439 A 194.210.238.440 A 194.210.238.441 A 194.210.238.442 A 194.210.238.443 A 194.210.238.444 A 194.210.238.445 A 194.210.238.446 A 194.210.238.447 A 194.210.238.448 A 194.210.238.449 A 194.210.238.450 A 194.210.238.451 A 194.210.238.452 A 194.210.238.453 A 194.210.238.454 A 194.210.238.455 A 194.210.238.456 A 194.210.238.457 A 194.210.238.458 A 194.210.238.459 A 194.210.238.460 A 194.210.238.461 A 194.210.238.462 A 194.210.238.463 A 194.210.238.464 A 194.210.238.465 A 194.210.238.466 A 194.210.238.467 A 194.210.238.468 A 194.210.238.469 A 194.210.238.470 A 194.210.238.471 A 194.210.238.472 A 194.210.238.473 A 194.210.238.474 A 194.210.238.475 A 194.210.238.476 A 194.210.238.477 A 194.210.238.478 A 194.210.238.479 A 194.210.238.480 A 194.210.238.481 A 194.210.238.482 A 194.210.238.483 A 194.210.238.484 A 194.210.238.485 A 194.210.238.486 A 194.210.238.487 A 194.210.238.488 A 194.210.238.489 A