

# 2º Trabalho Laboratorial



Mestrado Integrado em Engenharia Informática e  
Computação

Redes de Computadores

**Turma 3 - Grupo C**

Bernardo Barbosa - up201503477

Diogo Mota Pinto - up201404527

João Sá - up201506252

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

21 de Dezembro de 2017

## Sumário

Este relatório tem como objectivo descrever o segundo trabalho laboratorial da U.C. Redes de Computadores.

### Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Aplicação de Download</b>	<b>4</b>
2.1	Arquitetura . . . . .	4
2.1.1	Camada URL . . . . .	4
2.1.2	Camada FTP . . . . .	4
2.2	Funcionamento . . . . .	5
<b>3</b>	<b>Configuração de Rede e Estudo</b>	<b>7</b>
3.1	1ª Experiência - Configurar um IP de Rede . . . . .	7
3.2	2ª Experiência - Implementar duas LANS virtuais no switch .	7
3.3	3ª Experiência - Configurar um router em Linux . . . . .	7
3.4	4ª Experiência - Configurar um router comercial e implementar NAT . . . . .	7
3.5	5ª Experiência - DNS . . . . .	8
3.6	6ª Experiência - Ligações TCP . . . . .	8
<b>4</b>	<b>Conclusão</b>	<b>8</b>

## 1 Introdução

O trabalho realizado no âmbito da Unidade Curricular de Redes de Computadores teve como objetivos desenvolver uma aplicação de download de um ficheiro utilizando o protocolo FTP (*File Transfer Protocol*) e configurar uma rede de computadores.

Ambas as experiências foram realizadas separadamente sendo que a criação da rede de computadores foi feita no laboratório seguindo o guião, adaptando-o ao equipamento utilizado sendo que foi necessário configurar-se o endereço IP de rede, LANs virtuais no Switch, um router em Linux, um router comercial e o DNS.

Posteriormente, testou-se a aplicação de download na rede criada. A aplicação foi desenvolvida em C recorrendo à utilização de sockets para a comunicação com o servidor, enviando comandos e recebendo respostas.

Estes conceitos irão ser posteriormente abordados no relatório.

## 2 Aplicação de Download

Um dos objetivos do segundo projeto de RCOM era o desenvolvimento de uma aplicação de download na linguagem C.

Para compreender o funcionamento da mesma foi necessário estudar vários documentos RFCs, especialmente o RFC959 (FTP) e o RFC1738 (URL).

O grupo decidiu dividir a aplicação em duas camadas: A de processamento do URL e a do cliente FTP. Em cada camada existe uma estrutura que contém as propriedades necessárias às funções que estas desempenham.

### 2.1 Arquitetura

#### 2.1.1 Camada URL

---

```
typedef char url_content[256];

struct URL {
    url_content user; // string to user
    url_content password; // string to password
    url_content host; // string to host
    url_content ip; // string to IP
    url_content path; // string to path
    int port; // integer to port
};
```

---

Esta estrutura guarda em memória diversas strings que são preenchidas com os diferentes dados presentes no link: user, password, hostname, path. Após o processamento do URL, o atributo ip é preenchido. O atributo port é sempre 21 (número da porta de controlo do protocolo FTP).

---

```
void initURL(struct URL *url);
int parseURL(struct URL *url, const char *urlArg);

int parseNormalAuth(struct URL *url, const char *urlArg);
void initDefaultAuth(struct URL *url);
```

---

Descrição das principais funções que constituem esta camada:

- **initURL**, preenche com 0's as strings da instância da estrutura URL e iguala o atributo port a 21;
- **parseURL**, processa o link recebido como argumento pela aplicação e guarda a informação necessária nos atributos da estrutura URL;

#### 2.1.2 Camada FTP

---

```
struct FTP{
    int control_socket_fd; // file descriptor to control socket
```

```
int data_socket_fd; // file descriptor to data socket
};
```

---

Esta estrutura, contém apenas dois atributos: um descritor de ficheiro para o socket de controlo e um descritor de ficheiro para o socket de dados.

---

```
int getIp(struct URL *url);
int connect_to (const char *address, const int port);
int disconnect_from (const struct FTP *connection, const struct URL
    *url);

int ftpLogin (const struct FTP *connection, const struct URL *url);
int ftpPasv (struct FTP *connection, char *pasvIP, int *pasvPort);
int ftpRetr (const struct FTP *connection, const struct URL *url);

int ftpDownload (const struct FTP *connection, const struct URL
    *url);

int ftpWrite(const struct FTP *connection, const char *frame);
int ftpRead(const struct FTP *connection, char *frame, size_t
    frame_length, char *exp_code);
```

---

Descrição das principais funções que constituem esta camada:

- **getIp**, obtém o IP a partir de um hostname.
- **connect\_to**, Criada uma ligação, entre o cliente FTP e o servidor FTP, através de uma socket TCP;
- **disconnect\_from**, É enviada uma mensagem ao servidor FTP a dizer que vamos fechar a ligação;
- **ftpLogin**, é feito o Login no servidor;
- **ftpPasv**, é feito um pedido ao servidor para entrar em modo passivo;
- **ftpRetr**, é feito um pedido ao servidor para este enviar o ficheiro para Download;
- **ftpDownload**, é feito o Download do ficheiro e guardado no mesmo diretório que a aplicação;

## 2.2 Funcionamento

A aplicação aceita um link como argumento, que deve ser especificado através da linha de comandos. O link pode conter um username e password, ou então nenhum caso se pretenda usar o modo anonymous. O primeiro passo é fazer o processamento do URL, para isso é chamada a função *parseURL()*, seguida da função *getIP()*.

Depois de o processamento do URL estar concluído, é necessário estabelecer a conexão, por isso é estabelecida a conexão entre o cliente FTP e o servidor FTP através de uma socket TCP, graças à função *connect\_to()*.

Seguindo o protocolo FTP, o grupo enviou uma série de comandos por uma ordem previamente determinada:

1. "**USER user**", o nome do utilizador é enviado.
2. "**PASS pass**", a password do utilizador é enviada.
3. "**PASV**", é pedido ao servidor para entrar em modo passivo, permitindo uma mútua comunicação entre o servidor e o cliente FTP. É também feita nova conexão do socket mas desta vez a uma porta processa com informação recebida do servidor.
4. "**RETR path**" é pedido ao servidor o envio do ficheiro para download.
5. "**QUIT**" depois de feito o download do ficheiro, sinalizando que vamos fechar a conexão.

## 3 Configuração de Rede e Estudo

### 3.1 1ª Experiência - Configurar um IP de Rede

A experiência teve como objetivo reconfigurar o IP de duas máquinas de modo a que estas sejam capazes de comunicar entre si. Começa-se por reconfigurar o endereço IP e o endereço MAC de duas máquinas denominadas tux1 e tux4 e criando uma rota entre as duas. Para isso, são usados os comandos *ifconfig* e *route*. Posteriormente, é testada a ligação entre as duas fazendo *ping*, verificando se estas comunicam entre si. Verifica-se se as rotas foram bem criadas com o comando *route -n* e as tabelas ARP usando o comando *arp -a*, apaga-se as tabelas da Máquina 1 *arp -d ipaddress*. Depois, começamos por capturar o pacotes no eth0 da máquina um usando o WireShark fazendo *ping* à máquina 4.

### 3.2 2ª Experiência - Implementar duas LANS virtuais no switch

Já a experiência 2, teve como meta a implementação de duas LANS virtuais no *switch*, uma para as Máquinas 1 e 4 já ligadas entre elas e outra LAN para a máquina 2 e 4. Começou-se por configurar o endereço IP da máquina 2, e assim que foi configurada criou-se uma LAN com as portas 1 e 3 do switch (Máquinas 1 e 4, respetivamente) e outra LAN com as portas 7 e 9 (Máquinas 2 e 4). Por fim, na Máquina 1 fez-se ping à Máquina 4 e à Máquina 2 e verificaram-se os seguintes resultados,

### 3.3 3ª Experiência - Configurar um router em Linux

A experiência teve como fim configurar a máquina 4 como um router que permitisse a comunicação entre as duas VLANS criadas na experiência anterior. Como tal, começou-se por configurar a porta eth1 da Máquina 4 utilizando um IP da mesma gama que a Máquina 2. Depois de criadas todas as rotas, foi possível pingar a Máquina 2 utilizando a Máquina 1. Verificou-se assim a Máquina 4 é intermediária entre estas duas, pois o pedido é reencaminhado para a Máquina 4 que está ligada a ambas as máquinas e por isso consegue aceder à Máquina 2, na resposta o percurso é idêntico sendo o inverso, fazendo com que o pacote vá da Máquina 2 até à Máquina 1.

### 3.4 4ª Experiência - Configurar um router comercial e implementar NAT

Na experiência 4, pretendia-se configurar um router comercial com o NAT implementado. O NAT, Network Address Translation, é uma técnica que consiste em reescrever os endereços IP de origem de um pacote que passam por um router ou firewall de maneira que um computador de uma

rede interna tenha acesso ao exterior. Para tal, é gerado um número de 16 bits, utilizando uma tabela hash, escrevendo o número gerado no campo da porta de origem. Para isso foi necessário configurar o router com a Máquina 4, configurando assim a interface interna no processo de NAT. O router foi configurado definindo as rotas externas e internas através da consola. Depois desse processo, configurou-se o default gateway da Máquina 1 como Máquina 4 e o router como default gateway das Máquinas 2 e 4. Por fim verificou-se que todos os pacotes enviados pela Máquina 1 seguem para a Máquina 4, seguindo para o router ou para a Máquina 2.

### 3.5 5ª Experiência - DNS

Para conseguir aceder à rede externa através da rede interna criada foi necessário configurar o DNS, no ficheiro `resolv.conf` foram adicionadas duas linhas `search netlab.fe.up.pt` e `nameserver 172.16.1.1`. Foi testado o funcionamento desta fazendo ping a `www.google.com` e observando os pacotes no Wireshark.

### 3.6 6ª Experiência - Ligações TCP

Por fim, na experiência 6, compilou-se e executou-se a aplicação desenvolvida e descrita na primeira parte do relatório. Para testá-la, foi usado um servidor ftp da FEUP e efetuou-se o download de um ficheiro. O download efetuou-se correctamente, demonstrando que tanto a rede estava bem configurada, bem como a aplicação, acedendo sem erros ao servidor.

## 4 Conclusão

O objetivo de trabalho foi alcançado, pois foi possível de implementar um programa dividido em camadas independentes entre si. Através de um protocolo de ligação de dados capaz de enviar um ficheiro de uma Máquina para a outra, de acordo com as especificações pretendidas, preparado para detetar erros e corrigi-los eficientemente juntamente com uma rede implementada em laboratório, o grupo foi capaz de realizar o trabalho proposto com as funcionalidades pretendidas.

Concluindo, o trabalho realizado permitiu-nos consolidar e aplicar alguns dos conceitos estudados nas teóricas.



## Anexos

```
[jotapsa@Poseidon src]$ ./rcom.out ftp://ftp.up.pt/pub/CPAN/RECENT-1M.json
Username : anonymous
Password : mail@domain
Host : ftp.up.pt
Path : pub/CPAN/RECENT-1M.json
Host name : mirrors.up.pt
IP Address : 193.137.29.15
220-Welcome to the University of Porto's mirror archive (mirrors.up.pt)
220-----
220-
220-All connections and transfers are logged. The max number of connections is 200.
220-
220-For more information please visit our website: http://mirrors.up.pt/
220-Questions and comments can be sent to mirrors@uporto.pt
220-
220-
220
USER anonymous
331 Please specify the password.
PASS mail@domain
230 Login successful.
227 Entering Passive Mode (193,137,29,15,232,32).
150 Opening BINARY mode data connection for pub/CPAN/RECENT-1M.json (1101030 bytes).
226 Transfer complete.
221 Goodbye.
```

Figura 1: Modo anonimo

```
[jotapsa@Poseidon src]$ ./rcom.out ftp://[anonymous:domain@]ftp.up.pt/pub/CPAN/RECENT-1M.json
Username : anonymous
Password : domain
Host : ftp.up.pt
Path : pub/CPAN/RECENT-1M.json
Host name : mirrors.up.pt
IP Address : 193.137.29.15
220-Welcome to the University of Porto's mirror archive (mirrors.up.pt)
220-----
220-
220-All connections and transfers are logged. The max number of connections is 200.
220-
220-For more information please visit our website: http://mirrors.up.pt/
220-Questions and comments can be sent to mirrors@uporto.pt
220-
220-
220
USER anonymous
331 Please specify the password.
PASS domain
230 Login successful.
227 Entering Passive Mode (193,137,29,15,211,195).
150 Opening BINARY mode data connection for pub/CPAN/RECENT-1M.json (1101030 bytes).
226 Transfer complete.
221 Goodbye.
```

Figura 2: Modo normal

```

gnu61:~# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 172.16.60.254 0.0.0.0 UG 0 0 0 eth0
172.16.60.0 172.16.60.254 255.255.255.0 UG 0 0 0 eth0
172.16.60.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0

```

Figura 3: Rotas - Máquina 1

```

gnu62:~# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 172.16.61.254 0.0.0.0 UG 0 0 0 eth1
0.0.0.0 172.16.2.254 0.0.0.0 UG 0 0 0 eth0
172.16.2.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
172.16.60.0 172.16.61.253 255.255.255.0 UG 0 0 0 eth1
172.16.61.0 0.0.0.0 255.255.255.0 U 0 0 0 eth1

```

Figura 4: Rotas - Máquina 2

```

gnu64:~# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 172.16.61.254 0.0.0.0 UG 0 0 0 eth1
172.16.60.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
172.16.61.0 0.0.0.0 255.255.255.0 U 0 0 0 eth1

```

Figura 5: Rotas - Máquina 4

## Registros

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Cisco_7b:ce:81	Cisco_7b:ce:81	LOOP	60	Reply
2	1.359610	Cisco_7b:ce:81	Cisco_7b:ce:81	Spanning-tree-(f_ STP	60	Conf. Root = 32768/1/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001
3	3.355292	Cisco_7b:ce:81	Cisco_7b:ce:81	Spanning-tree-(f_ STP	60	Conf. Root = 32768/1/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001
4	5.360167	Cisco_7b:ce:81	Cisco_7b:ce:81	Spanning-tree-(f_ STP	60	Conf. Root = 32768/1/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001
5	7.364988	Cisco_7b:ce:81	Cisco_7b:ce:81	Spanning-tree-(f_ STP	60	Conf. Root = 32768/1/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001
6	8.324940	172.16.60.1	172.16.60.254	ICMP	98	Echo (ping) request id=0x0893, seq=1/256, ttl=64 (reply in 7)
7	8.325086	172.16.60.254	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0893, seq=1/256, ttl=64 (request in 6)
8	9.323940	172.16.60.1	172.16.60.254	ICMP	98	Echo (ping) request id=0x0893, seq=2/512, ttl=64 (reply in 9)
9	9.324082	172.16.60.254	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0893, seq=2/512, ttl=64 (request in 8)
10	9.374317	Cisco_7b:ce:81	Cisco_7b:ce:81	Spanning-tree-(f_ STP	60	Conf. Root = 32768/1/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001
11	10.007479	Cisco_7b:ce:81	Cisco_7b:ce:81	LOOP	60	Reply
12	10.322941	172.16.60.1	172.16.60.254	ICMP	98	Echo (ping) request id=0x0893, seq=3/768, ttl=64 (reply in 13)
13	10.323059	172.16.60.254	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0893, seq=3/768, ttl=64 (request in 12)
14	11.322437	172.16.60.1	172.16.60.254	ICMP	98	Echo (ping) request id=0x0893, seq=4/1024, ttl=64 (reply in 15)
15	11.322559	172.16.60.254	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0893, seq=4/1024, ttl=64 (request in 14)
16	11.374664	Cisco_7b:ce:81	Cisco_7b:ce:81	Spanning-tree-(f_ STP	60	Conf. Root = 32768/1/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001
17	12.322440	172.16.60.1	172.16.60.254	ICMP	98	Echo (ping) request id=0x0893, seq=5/1280, ttl=64 (reply in 18)
18	12.322558	172.16.60.254	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0893, seq=5/1280, ttl=64 (request in 17)
19	13.322436	172.16.60.1	172.16.60.254	ICMP	98	Echo (ping) request id=0x0893, seq=6/1536, ttl=64 (reply in 20)
20	13.322573	172.16.60.254	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0893, seq=6/1536, ttl=64 (request in 19)
21	13.334428	HewlettP_5a:79:97	Netronix_71:71:2e	ARP	60	Who has 172.16.60.1? Tell 172.16.60.254
22	13.334438	Netronix_71:71:2e	HewlettP_5a:79:97	ARP	42	172.16.60.1 is at 00:08:54:71:71:2e
23	13.379575	Cisco_7b:ce:81	Cisco_7b:ce:81	Spanning-tree-(f_ STP	60	Conf. Root = 32768/1/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001

▶ Frame 6: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0  
 ▶ Ethernet II, Src: Netronix\_71:71:2e (00:08:54:71:71:2e), Dst: HewlettP\_5a:79:97 (00:21:5a:5a:79:97)  
 ▶ Internet Protocol Version 4, Src: 172.16.60.1, Dst: 172.16.60.254  
 ▶ Internet Control Message Protocol

Figura 6: Experiência 1 - Máquina 1 —> Máquina 4

No.	Time	Source	Destination	Protocol	Length	Info
13	11.324161	172.16.60.254	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0d39, seq=3/768, ttl=64 (request in 12)
14	12.029067	Cisco_7b:ce:81	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001	
15	12.323747	172.16.60.1	172.16.60.254	ICMP	98	Echo (ping) request id=0x0d39, seq=4/1024, ttl=64 (reply in 16)
16	12.323863	172.16.60.254	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0d39, seq=4/1024, ttl=64 (request in 15)
17	13.323745	172.16.60.1	172.16.60.254	ICMP	98	Echo (ping) request id=0x0d39, seq=5/1280, ttl=64 (reply in 18)
18	13.323871	172.16.60.254	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0d39, seq=5/1280, ttl=64 (request in 17)
19	14.037978	Cisco_7b:ce:81	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001	
20	14.338780	HewlettP_5a:79:97	Netronix_71:71:2e	ARP	60	Who has 172.16.60.1? Tell 172.16.60.254
21	14.338797	Netronix_71:71:2e	HewlettP_5a:79:97	ARP	42	172.16.60.1 is at 00:08:54:71:71:2e
22	16.038710	Cisco_7b:ce:81	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001	
23	17.523618	Cisco_7b:ce:81	Cisco_7b:ce:81	LOOP	60	Reply
24	18.043563	Cisco_7b:ce:81	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001	
25	19.461969	172.16.60.1	172.16.61.253	ICMP	98	Echo (ping) request id=0x0d40, seq=1/256, ttl=64 (reply in 26)
26	19.462112	172.16.61.253	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0d40, seq=1/256, ttl=64 (request in 25)
27	20.048429	Cisco_7b:ce:81	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001	
28	20.460972	172.16.60.1	172.16.61.253	ICMP	98	Echo (ping) request id=0x0d40, seq=2/512, ttl=64 (reply in 29)
29	20.461107	172.16.61.253	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0d40, seq=2/512, ttl=64 (request in 28)
30	21.459975	172.16.60.1	172.16.61.253	ICMP	98	Echo (ping) request id=0x0d40, seq=3/768, ttl=64 (reply in 31)
31	21.460093	172.16.61.253	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0d40, seq=3/768, ttl=64 (request in 30)
32	22.053251	Cisco_7b:ce:81	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001	
33	22.459742	172.16.60.1	172.16.61.253	ICMP	98	Echo (ping) request id=0x0d40, seq=4/1024, ttl=64 (reply in 34)
34	22.459855	172.16.61.253	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0d40, seq=4/1024, ttl=64 (request in 33)
35	23.459755	172.16.60.1	172.16.61.253	ICMP	98	Echo (ping) request id=0x0d40, seq=5/1280, ttl=64 (reply in 36)
36	23.459870	172.16.61.253	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0d40, seq=5/1280, ttl=64 (request in 35)
37	24.062307	Cisco_7b:ce:81	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001	
38	26.063005	Cisco_7b:ce:81	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001	
39	27.531121	Cisco_7b:ce:81	Cisco_7b:ce:81	LOOP	60	Reply
40	28.067799	Cisco_7b:ce:81	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001	
41	28.510048	172.16.60.1	172.16.61.1	ICMP	98	Echo (ping) request id=0x0d48, seq=1/256, ttl=64 (reply in 42)
42	28.510304	172.16.61.1	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0d48, seq=1/256, ttl=63 (request in 41)
43	29.509050	172.16.60.1	172.16.61.1	ICMP	98	Echo (ping) request id=0x0d48, seq=2/512, ttl=64 (reply in 44)
44	29.509273	172.16.61.1	172.16.60.1	ICMP	98	Echo (ping) replv id=0x0d48, seq=2/512, ttl=63 (request in 43)

Figura 7: Experiência 3 - Máquina 1 —> Máquina 4/ Máquina 2

No.	Time	Source	Destination	Protocol	Length	Info
2	0.754580	Cisco_7b:ce:83	Cisco_7b:ce:83	LOOP	60	Reply
3	2.004726	Cisco_7b:ce:83	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8003	
4	4.009639	Cisco_7b:ce:83	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8003	
5	6.014554	Cisco_7b:ce:83	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8003	
6	8.023789	Cisco_7b:ce:83	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8003	
7	10.024373	Cisco_7b:ce:83	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8003	
8	10.753959	Cisco_7b:ce:83	Cisco_7b:ce:83	LOOP	60	Reply
9	11.976287	Netronix_71:71:2e	Broadcast	ARP	60	Who has 172.16.60.254? Tell 172.16.60.1
10	11.976311	HewlettP_5a:79:97	Netronix_71:71:2e	ARP	42	172.16.60.254 is at 00:21:5a:5a:79:97
11	11.976409	172.16.60.1	172.16.61.1	ICMP	98	Echo (ping) request id=0x0f4a, seq=1/256, ttl=64 (reply in 12)
12	11.976642	172.16.61.1	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0f4a, seq=1/256, ttl=63 (request in 11)
13	12.029277	Cisco_7b:ce:83	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8003	
14	12.976420	172.16.60.1	172.16.61.1	ICMP	98	Echo (ping) request id=0x0f4a, seq=2/512, ttl=64 (reply in 15)
15	12.976563	172.16.61.1	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0f4a, seq=2/512, ttl=63 (request in 14)
16	13.976439	172.16.60.1	172.16.61.1	ICMP	98	Echo (ping) request id=0x0f4a, seq=3/768, ttl=64 (reply in 17)
17	13.976571	172.16.61.1	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0f4a, seq=3/768, ttl=63 (request in 16)
18	14.034322	Cisco_7b:ce:83	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8003	
19	14.976474	172.16.60.1	172.16.61.1	ICMP	98	Echo (ping) request id=0x0f4a, seq=4/1024, ttl=64 (reply in 20)
20	14.976618	172.16.61.1	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0f4a, seq=4/1024, ttl=63 (request in 19)
21	15.976507	172.16.60.1	172.16.61.1	ICMP	98	Echo (ping) request id=0x0f4a, seq=5/1280, ttl=64 (reply in 22)
22	15.976639	172.16.61.1	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0f4a, seq=5/1280, ttl=63 (request in 21)
23	16.039280	Cisco_7b:ce:83	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8003	
24	16.990416	HewlettP_5a:79:97	Netronix_71:71:2e	ARP	42	Who has 172.16.60.1? Tell 172.16.60.254
25	16.990508	Netronix_71:71:2e	HewlettP_5a:79:97	ARP	60	172.16.60.1 is at 00:08:54:71:71:2e
26	18.048768	Cisco_7b:ce:83	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8003	
27	20.049170	Cisco_7b:ce:83	Spanning-tree-(f_ STP	60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8003	
28	20.761741	Cisco_7b:ce:83	Cisco_7b:ce:83	LOOP	60	Reply

Figura 8: Experiência 3 - Máquina 1 —> Máquina 2 | Interface Eth0 na Máquina 4

No.	Time	Source	Destination	Protocol	Length	Info
2	2.000577	Cisco_7b:ce:89	Spanning-tree-(f_ STP		60	Conf. Root = 32768/61/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8009
3	4.009543	Cisco_7b:ce:89	Spanning-tree-(f_ STP		60	Conf. Root = 32768/61/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8009
4	5.439662	Cisco_7b:ce:89	Cisco_7b:ce:89	LOOP		60 Reply
5	9.010304	Cisco_7b:ce:89	Spanning-tree-(f_ STP		60	Conf. Root = 32768/61/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8009
6	6.636967	172.16.60.1	172.16.61.1	ICMP	98	Echo (ping) request id=0x0f4a, seq=1/256, ttl=63 (reply in 9)
7	6.637073	Kye_25:21:9e	Broadcast	ARP	60	Who has 172.16.61.253? Tell 172.16.61.1
8	6.637086	Netronix_71:73:da	Kye_25:21:9e	ARP	42	172.16.61.253 is at 00:08:54:71:73:da
9	6.637172	172.16.61.1	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0f4a, seq=1/256, ttl=64 (request in 6)
10	7.636978	172.16.60.1	172.16.61.1	ICMP	98	Echo (ping) request id=0x0f4a, seq=2/512, ttl=63 (reply in 11)
11	7.637083	172.16.61.1	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0f4a, seq=2/512, ttl=64 (request in 10)
12	7.837456	Kye_25:21:9e	Broadcast	ARP	60	Who has 172.16.61.254? Tell 172.16.2.62
13	8.015370	Cisco_7b:ce:89	Spanning-tree-(f_ STP		60	Conf. Root = 32768/61/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8009
14	8.636997	172.16.60.1	172.16.61.1	ICMP	98	Echo (ping) request id=0x0f4a, seq=3/768, ttl=63 (reply in 15)
15	8.637093	172.16.61.1	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0f4a, seq=3/768, ttl=64 (request in 14)
16	8.834243	Kye_25:21:9e	Broadcast	ARP	60	Who has 172.16.61.254? Tell 172.16.2.62
17	9.637024	172.16.60.1	172.16.61.1	ICMP	98	Echo (ping) request id=0x0f4a, seq=4/1024, ttl=63 (reply in 18)
18	9.637138	172.16.61.1	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0f4a, seq=4/1024, ttl=64 (request in 17)
19	9.834262	Kye_25:21:9e	Broadcast	ARP	60	Who has 172.16.61.254? Tell 172.16.2.62
20	10.024206	Cisco_7b:ce:89	Spanning-tree-(f_ STP		60	Conf. Root = 32768/61/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8009
21	10.637063	172.16.60.1	172.16.61.1	ICMP	98	Echo (ping) request id=0x0f4a, seq=5/1280, ttl=63 (reply in 22)
22	10.637161	172.16.61.1	172.16.60.1	ICMP	98	Echo (ping) reply id=0x0f4a, seq=5/1280, ttl=64 (request in 21)
23	12.025348	Cisco_7b:ce:89	Spanning-tree-(f_ STP		60	Conf. Root = 32768/61/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8009
24	12.642961	Netronix_71:73:da	Kye_25:21:9e	ARP	42	Who has 172.16.61.1? Tell 172.16.61.253
25	12.643052	Kye_25:21:9e	Netronix_71:73:da	ARP	60	172.16.61.1 is at 00:c0:df:25:21:9e
26	14.029927	Cisco_7b:ce:89	Spanning-tree-(f_ STP		60	Conf. Root = 32768/61/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8009
27	15.447451	Cisco_7b:ce:89	Cisco_7b:ce:89	LOOP		60 Reply
28	16.034921	Cisco_7b:ce:89	Spanning-tree-(f_ STP		60	Conf. Root = 32768/61/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8009

Figura 9: Experiência 3 - Máquina 1 —> Máquina 2 | Interface Eth1 na Máquina 4

No.	Time	Source	Destination	Protocol	Length	Info
16	17.342167	172.16.60.1	172.16.61.253	ICMP	98	Echo (ping) request id=0x1e10, seq=3/768, ttl=64 (reply in 17)
17	17.342285	172.16.61.253	172.16.60.1	ICMP	98	Echo (ping) reply id=0x1e10, seq=3/768, ttl=64 (request in 16)
18	18.039488	Cisco_7b:ce:81	Spanning-tree-(f_ STP		60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001
19	18.341164	172.16.60.1	172.16.61.253	ICMP	98	Echo (ping) request id=0x1e10, seq=4/1024, ttl=64 (reply in 20)
20	18.341291	172.16.61.253	172.16.60.1	ICMP	98	Echo (ping) reply id=0x1e10, seq=4/1024, ttl=64 (request in 19)
21	19.340920	172.16.60.1	172.16.61.253	ICMP	98	Echo (ping) request id=0x1e10, seq=5/1280, ttl=64 (reply in 22)
22	19.341038	172.16.61.253	172.16.60.1	ICMP	98	Echo (ping) reply id=0x1e10, seq=5/1280, ttl=64 (request in 21)
23	20.040492	Cisco_7b:ce:81	Spanning-tree-(f_ STP		60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001
24	20.344930	HewlettP_5a:79:97	Netronix_71:71:2e	ARP	60	Who has 172.16.60.1? Tell 172.16.60.254
25	20.344943	Netronix_71:71:2e	HewlettP_5a:79:97	ARP	42	172.16.60.1 is at 00:08:54:71:71:2e
26	22.049186	Cisco_7b:ce:81	Spanning-tree-(f_ STP		60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001
27	23.088032	172.16.60.1	172.16.61.1	ICMP	98	Echo (ping) request id=0x1e14, seq=1/256, ttl=64 (reply in 28)
28	23.088283	172.16.61.1	172.16.60.1	ICMP	98	Echo (ping) reply id=0x1e14, seq=1/256, ttl=63 (request in 27)
29	24.054055	Cisco_7b:ce:81	Spanning-tree-(f_ STP		60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001
30	24.087032	172.16.60.1	172.16.61.1	ICMP	98	Echo (ping) request id=0x1e14, seq=2/512, ttl=64 (reply in 31)
31	24.087262	172.16.61.1	172.16.60.1	ICMP	98	Echo (ping) reply id=0x1e14, seq=2/512, ttl=63 (request in 30)
32	25.090039	172.16.60.1	172.16.61.1	ICMP	98	Echo (ping) request id=0x1e14, seq=3/768, ttl=64 (reply in 33)
33	25.090267	172.16.61.1	172.16.60.1	ICMP	98	Echo (ping) reply id=0x1e14, seq=3/768, ttl=63 (request in 32)
34	26.058854	Cisco_7b:ce:81	Spanning-tree-(f_ STP		60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001
35	26.085034	172.16.60.1	172.16.61.1	ICMP	98	Echo (ping) request id=0x1e14, seq=4/1024, ttl=64 (reply in 36)
36	26.085271	172.16.61.1	172.16.60.1	ICMP	98	Echo (ping) reply id=0x1e14, seq=4/1024, ttl=63 (request in 35)
37	26.956579	Cisco_7b:ce:81	Cisco_7b:ce:81	LOOP		60 Reply
38	27.084921	172.16.60.1	172.16.61.1	ICMP	98	Echo (ping) request id=0x1e14, seq=5/1280, ttl=64 (reply in 39)
39	27.085132	172.16.61.1	172.16.60.1	ICMP	98	Echo (ping) reply id=0x1e14, seq=5/1280, ttl=63 (request in 38)
40	28.063704	Cisco_7b:ce:81	Spanning-tree-(f_ STP		60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001
41	30.072690	Cisco_7b:ce:81	Spanning-tree-(f_ STP		60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001
42	30.784216	172.16.60.1	172.16.61.254	ICMP	98	Echo (ping) request id=0x1e1b, seq=1/256, ttl=64 (reply in 43)
43	30.785135	172.16.61.254	172.16.60.1	ICMP	98	Echo (ping) reply id=0x1e1b, seq=1/256, ttl=254 (request in 42)
44	31.784922	172.16.60.1	172.16.61.254	ICMP	98	Echo (ping) request id=0x1e1b, seq=2/512, ttl=64 (reply in 45)
45	31.785818	172.16.61.254	172.16.60.1	ICMP	98	Echo (ping) reply id=0x1e1b, seq=2/512, ttl=254 (request in 44)
46	32.073435	Cisco_7b:ce:81	Spanning-tree-(f_ STP		60	Conf. Root = 32768/60/00:1e:14:7b:ce:80 Cost = 0 Port = 0x8001
47	32.784922	172.16.60.1	172.16.61.254	ICMP	98	Echo (ping) request id=0x1e1b, seq=3/768, ttl=64 (reply in 48)

Figura 10: Experiência 4 - Máquina 1 —> Máquina 4/ Máquina 2/ Router