Lab2 - Computer Networks

Mestrado Integrado em Engenharia Informática e Computação

Redes de Computadores

Turma 5:
Daniel Pereira Machado - 201506365
José Pedro Dias de Almeida Machado - 201504779
Sofia Catarina Bahamonde Alves - 201504570

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

22 de Dezembro de 2017

Conteúdo

1 Sumário

Este relatório tem como objetivo contextualizar o trabalho que tem vindo a ser desenvolvido na Unidade Curricular Redes de Computadores. Trata-se de implementar um cliente FTP, assim como configurar um rede de computadores. Deste modo foi desenvolvida uma *download application* e ao longo das aulas foram feitas todas as configurações necessárias, de modo a conseguir estabelecer uma rede como o que era esperado.

O projeto foi concluído com sucesso. Foi desenvolvida uma aplicação transferir um ficheiro através de um protocolo FTP. Esta aplicação tanto funciona em modo anónimo com utilizando um *username* e respetiva *password*.

2 Introdução

Este projeto encontra-se dividido em duas grandes partes. A primeira é implementar um cliente FTP (File Transfer Protocol) para fazer a transferência de um ficheiro, sendo que nesse sentido foi desenvolvida uma aplicação, *download application*. Numa primeira parte vamos explicar como é que essa aplicação foi desenvolvida assim como a sua arquitetura.

Numa segunda parte, o relatório vai-se debruçar sobre a configuração de uma rede de computadores. Ao longo das aulas práticas foram realizadas um conjunto de experiências no sentido de configurar essa mesma rede, nesta secção vamos explicar o que é suposto ser alcançado em cada experiência assim como os resultados que conseguimos obter. As experiências são as seguintes:

1. Configuração de um *IP* de rede;
2. Configuração de duas Redes *LAN* virtuais num *switch*;
3. Configuração de um *router* em *Linux*;
4. Configuração de um *router* comercial implementando *NAT*;
5. *DNS*;
6. Conexões *TCP*.

3 Download Application

A primeira parte deste trabalho consiste numa aplicação responsável pela transferência de um ficheiro utilizando o protocolo FTP. Esta aplicação permite que o download seja feito tanto em modo anónimo com através de um username e respectiva password, de seguida são apresentados alguns exemplos de utilização da aplicação.

- Modo User: `ftp://<user>:<password>@<host>/<url.path>`
- Modo Anónimo: `ftp://anonymous:1@<host>/<url.path>`
- Modo Anónimo (default): `ftp://<host>/<url.path>`

3.1 Arquitetura

A aplicação desenvolvida encontra-se devida em duas partes fundamentais, a primeira engloba o parser do url fornecido como argumento, que pode ser encontrada nos ficheiros url_parsing.c e url_parsing.h. A segunda parte, contida nos ficheiros TCP.c e TCP.h, é responsável por estabelecer e gerir as ligações TCP's necessárias para controlo e transferência de dados.

3.1.1 URL Parser

Por questões de organização e simplificação foi criada a estrutura url responsável por guardar informação fundamental para as ligação que serão criadas, informações estas que são retiradas do input do utilizador.

```
1  typedef struct{
2    char  user[256];
3    char  password[256];
4    char  host[256];
5    char  file_path[256];
6    char  file_name[256];
7    char  ip[256];
8  } url;
```

Struct Url

A primeira função a ser invocada será parse_url() que recebe como argumento o input do utilizador e uma instanciação da estrutura referida acima onde será armazenada a informação contida no primeiro argumento que será, antes disso, devidamente interpretada e validada. Esta função recorre a outras duas, nomeadamente userPassword() e getIP(), cuja funcionalidade pode ser facilmente deduzida.

3.1.2 TCP

Depois de interpretada e devidamente validada a informação introduzida pelo utilizador é chamada a função initConnection() responsável pela inicialização de uma socket de controlo. A ligação será por fim estabelecida através da função login que usará a socket inicializada posteriormente permitindo deta forma validar as credencias fornecidas pelo utilizador.

```
void login(ftp ftp, url url){
  char usr_cmd[MAX_SIZE];
  char pwd_cmd[MAX_SIZE];

  socketRead(ftp.ctrl_socket_fd, NULL);

  sprintf(usr_cmd, "USER %s\r\n", url.user);
  printf(">%s",usr_cmd);

  socketWrite(ftp.ctrl_socket_fd,usr_cmd);
  socketRead(ftp.ctrl_socket_fd,NULL);

  sprintf(pwd_cmd, "PASS %s\r\n", url.password);
  printf(">%s",pwd_cmd);

  socketWrite(ftp.ctrl_socket_fd, pwd_cmd);
  if(socketRead(ftp.ctrl_socket_fd,NULL) !=0){
    fprintf(stderr, "Wrong credentials. Exiting...\n");
    exit(1);
  }
}
```

Função Login

Seguidamente será enviado o comando PASV, que funcina como um pedido ao servidor FTP para transferir dados em modo passivo, ficando o cliente responsável pela abertura da ligação TCP para os dados. Será então criada um nova conexão recorrendo diretamente à função initSocket(), a fim de permitir a receção do ficheiro. A função retrieve() enviará o comando RETR necessário para inicializar a transferência, a receção dos dados e a respectiva escrita dos mesmo em disco será realizada através da função download().

```
int download(ftp ftp, url url){
  FILE* dest_file;
  if(!(dest_file = fopen(url.file_name, "w"))) {
        printf("Error opening file %s.\n",url.file_name);
        return 1;
    }

  char buf[1024];
  int bytes;
  while ((bytes = read(ftp.data_socket_fd, buf, sizeof(buf)))) {
    if (bytes < 0) {
      fprintf(stderr, "Error, nothing was received from data socket
          fd.\n");
      return 1;
    }

    if ((bytes = fwrite(buf, bytes, 1, dest_file)) < 0) {
      fprintf(stderr, "Error, cannot write data in file.\n");
      return 1;
    }
  }

  fclose(dest_file);

  printf("Finished downloading file\n");

  return 0;
}
```

Função Download

Terminada a receção de dados, a função endConnection() terá o papel fundamental de enviar o comando QUIT após o qual iraá fechar as sokets abertas anteriormente e libertar a memória alocada ao longo da execução a fim de terminar o programa com sucesso.

3.2 Resultados

A aplicação desenvolvida foi testada com diversos ficheiros, tanto em modo anónimo como em modo não anónimo, nunca sendo detetada uma anomalia na versão final, tendo o máximo ficheiro transferido um tamanho correspondente a cerca de 150MB.

Em caso de erro, para além de se proceder à terminação do processo será originada uma mensagem de erro contendo, dentro do possível, informação acerca do sucedido.

4 Configuração de uma Rede

4.1 Configuração de um *IP* de Rede

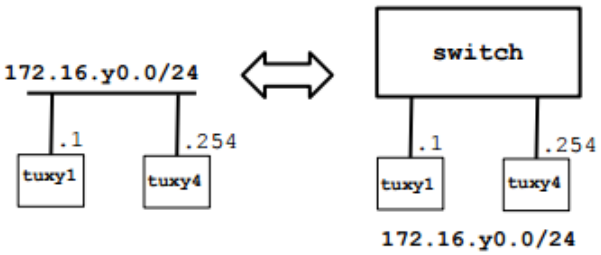


Figura 1: Experiência 1

A primeira experiência teve como objetivo a comunicação de 2 computadores na mesma rede para isso, foi preciso configurar os o **tux21** e o **tux24** para que eles assumissem os IP's pretendidos. Esta configuração foi feita usando **ifconfig** eth0 pois era a porta **eth0** de cada tux que estava ligada ao router cisco. O **tux21** foi configurado com o endereço **172.16.20.1** e o **tux24** com o endereço **172.16.20.254**. Analisando o log capturado no wireshark podemos verificar que o pacote ARP envia uma mensagem broadcast a todos os computadores com o objetivo de saber qual o endereço MAC que corresponde a um dado IP, depois disso todos os pacotes ICMP enviados recebem uma resposta deste mesmo.

4.1.1 Respostas

- Os pacotes ARP (Address Resolution Protocol) são pacotes enviados por um emissor que permite descobrir qual o endereço MAC de um destinatário cujo endereço IP já possui. Para isso, as máquinas ligadas na rede possuem uma tabela de endereços em cache que permite saber, dado um determinado pacote ARP, se o endereço IP nele contido corresponde ao seu endereço.
- Num pacote ARP o endereço MAC corresponde ao endereço físico do emissor do pacote original para que possa depois receber a resposta da mensagem, enquanto o endereço IP corresponde ao endereço IP da máquina cujo endereço MAC se pretende determinar.
- O comando "ping" gera pacotes ICMP com o objetivo de captar respostas (eco) de um destinatário, assim determinando se ele se encontra conectado e ativo.
- Os pacotes gerados por "ping" contêm o endereço MAC do emissor da mensagem e o endereço IP do destinatário pretendido.
- Para distinguir os pacotes ARP dos pacotes Ip temos um campo de 2 bytes ("type") que os distingue, entre os pacotes ICMP e IP existe um campo de 1 byte ("protocol") no IP header.
- Para ver o tamanho dos pacotes basta analisar no wireshark o campo frame length.

- A interface loopback é responsável por enviar de 10 em 10 segundos um pacote chamado do tipo LOOP que serve para verificar se a ligação se encontra ativa.

4.2 Configuração de duas Redes *LAN* virtuais num *switch*

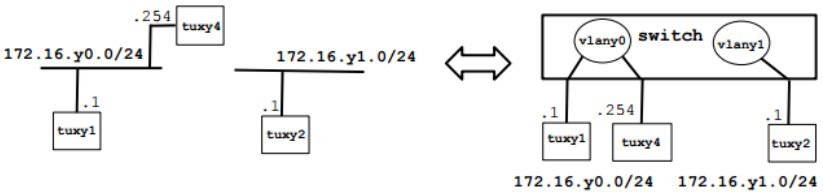


Figura 2: Experiência 2

Esta experiência consistia em criar duas LANs virtuais:

- a **vlan 20** com o **tux21** e o **tux24**
- a **vlan 21**com o **tux22**

Com esta configuração o **tux22** deixou de ter acesso aos outro 2 por se encontrar em sub-redes diferentes. Foi o utilizado o comando ping com os endereços no **tux22** e do **tux24** através do **tux21** e através da análise dos logs concluímos que por haverem duas sub-redes diferentes também existem duas broadcasts. Com isto ao utilizar o comando ping broadcast no **tux21** apenas vai receber os packets o **tux24** porque o **tux22** está num broadcast diferente. Assim conclui-se que existem 2 domínios de broadcast, **vlan 20** e **vlan 21**.

```
1     enable
2     configure terminal
3     vlan 20
4     exit
5     vlan 21
6     exit
7     interface fastethernet 0/1
8     switchport mode access
9     switchport mode access vlan 20
10    interface fastethernet 0/3
11    switchport mode access
12    switchport mode access vlan 20
13    interface fastethernet 0/5
14    switchport mode access
15    switchport mode access vlan 21
16    end
17    show vlan brief
```

Função Download

4.3 Configuração de um *router* em *Linux*

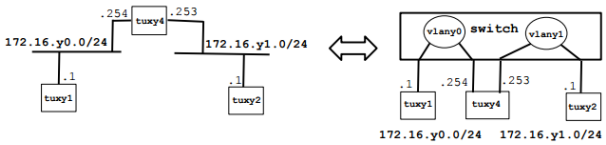


Figura 3: Experiência 1

Esta experiência consiste na criação de duas VLANs diferentes:

- **VLAN 20** - 172.16.20.0/24
- **VLAN 21** - 172.16.21.0/24

O computador tux2 pertence à Vlan 21 enquanto que os computadores tux1 e tux4 pertencem à Vlan 20. Antes demais deve-se ativar a porta eth1 do tux4 e ligar a mesma ao switch. Esta porta será a que se liga à Vlan 41. Configura-se esta porta com o endereço 172.16.21.253/24. Uma vez que se pretende que este computador sirva de router é necessário ativar o reencaminhamento de IPs através do comando:

De seguida adicionam-se as rotas necessárias no tux1 e no tux2 de forma a que estes, através do tux4, consigam aceder a uma rede que não pertencem. O primeiro endereço da rota identifica a gama de endereços para o qual se quer adicionar a rota, ou seja, os possíveis endereços de destino. Por sua vez, o segundo endereço identifica o IP para o qual se deve encaminhar o pacote. As rotas são adicionadas através do comando:

(adicionar mac adress, ARP messages, ICMP)

4.4 Configuração de um *router* comercial implementando *NAT*

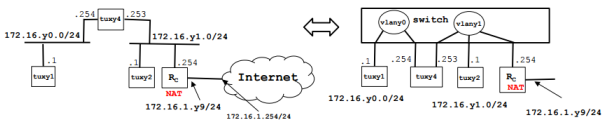


Figura 4: Experiência 1

(mostrar comandos) Esta experiência tem como objetivo a configuração do CISCO dentro da rede 21 de forma a que tanto os computadores na vlan20 como na vlan21 tenham acesso à Internet. Para configurar o router, depois de fazer login na linha de comandos corre-se o script do Anex. Os comandos referidos em anexo começam por configurar duas interfaces do router, atribuindo corretamente as configurações NAT. A configuração correta do NAT é essencial pois a falta desta poderia resultar em falta de acesso à Internet em qualquer um dos computadores pois, NAT tem a função de traduzir endereços, resultando neste caso na tradução do endereço de sub-rede de cada computador no endereço do router comercial.

O NAT tem como objetivo poupar o espaço de endereçamento público, recorrendo a IPs privados, uma vez que os IPs públicos são um recurso limitado

e atualmente escasso. Os endereços públicos são pagos e permitem identificar de forma unívoca uma máquina (PC, routers,etc.) na Internet, sendo que uma máquina configurada com um endereço IP privado terá para a Internet através do NAT. Essa tradução de um endereço privado num endereço público é conseguida através do NAT

4.5 DNS

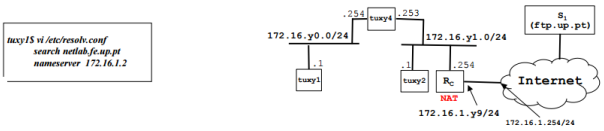


Figura 5: Experiência 5

Esta experiência tem como objetivo aceder a redes externas, conseguindo assim aceder à Internet através da rede interna criada. Assim sendo, é necessário configurar o DNS.

O DNS é um sistema de gestão de nomes hierárquico e distribuído para computadores, serviços ou qualquer recurso conectado à Internet ou mesmo numa rede privada. Funciona como um sistema de tradução de endereços IP em nomes de domínios. Para configurar o DNS, é necessário, em todos os hosts da rede criada, aceder e editar o ficheiro resolv.conf. Este ficheiro é lido cada vez que são invocadas rotinas que fornecem acesso à Internet. Neste caso, foi editado de modo a que fosse semelhante ao seguinte:

```
1 search netlab.fe.up.pt
2 nameserver 172.16.1.1
```

Ficheiro resolv.conf editado

Para testar esta experiência foi feito o teste de ping usando *www.google.com*. Nos logs, consequentemente, verificou-se que o DNS pergunta a informação contida num dado domain name, e este responde com o tempo de vido e o tamanho do pacote de dados.

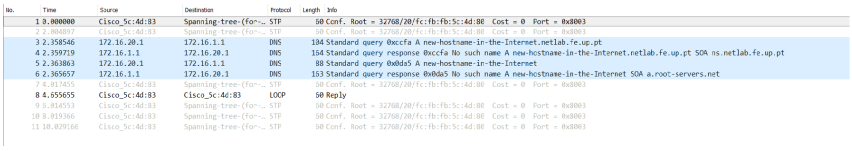


Figura 6: Dados DNS no Wireshark

4.6 Conexões *TCP*

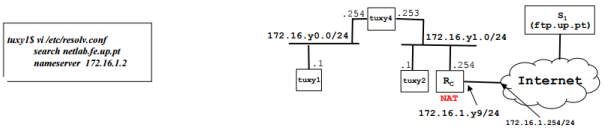


Figura 7: Experiência 1

Por fim, na experiência 6, compilou-se e executou-se a aplicação desenvolvida e descrita na primeira parte do relatório. O teste foi feito com recurso à transferência de um ficheiro através de um servidor FTP. A transferência foi bem-sucedida, mostrando que a configuração da rede foi feita sem erros.

O TCP (Transmission Control Protocol) utiliza o mecanismo ARQ (Automatic Repeat Request). Neste protocolo são definidas três fases: establecer a conexão, envio de informação e terminação. Este método consiste no controlo de erros na transmissão de dados. Para isso utiliza acknowledgment), de forma a garantir uma transmissão confiável através do serviço não confiável. Se não for recebido um acknowledgment antes do timeout, a trama é retransmitida até ser recebido um acknowledgment. Tal como podemos verificar nos logs do Whiteshark, cada vez que a porta 21 transmite temos um valor de ACK.

No nosso projeto são abertas duas conexões pela aplicação FTP. A conexão relativa à porta 21 é responsável pelos comandos, sendo a sua fonte o IP 172.16.20.1 e o destino o IP 90.130.70.73. A segunda conexão tem como fonte o IP 90.130.70.73 e destino o IP 172.16.20.1, sendo responsável pela transferência do ficheiro em si, é nesta que se encontra a informação de controlo do protocolo FTP.

5	6.014609	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60 Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
6	6.471985	172.16.20.1	172.16.1.1	DNS	79 Standard query 0x922f A speedtest.tele2.net
7	6.473445	172.16.1.1	172.16.20.1	DNS	233 Standard query response 0x922f A speedtest.tele2.net A 90.130.70.73 NS kista
8	6.473748	172.16.20.1	90.130.70.73	TCP	74 59971 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=36542728 T
9	6.525439	90.130.70.73	172.16.20.1	TCP	74 21 → 59971 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval
10	6.525484	172.16.20.1	90.130.70.73	TCP	66 59971 → 21 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=36542741 TSecr=1382265317
11	7.126635	90.130.70.73	172.16.20.1	FTP	86 Response: 220 (vsFTPd 2.3.5)
12	7.126692	172.16.20.1	90.130.70.73	TCP	66 59971 → 21 [ACK] Seq=1 Ack=21 Win=29312 Len=0 TSval=36542892 TSecr=13822654
13	7.126751	172.16.20.1	90.130.70.73	FTP	82 Request: USER anonymous
14	7.177734	90.130.70.73	172.16.20.1	TCP	66 21 → 59971 [ACK] Seq=21 Ack=17 Win=14592 Len=0 TSval=1382265475 TSecr=365428
15	7.177752	90.130.70.73	172.16.20.1	FTP	100 Response: 331 Please specify the password.
16	7.177826	172.16.20.1	90.130.70.73	FTP	74 Request: PASS 1

Figura 8: Dados FTP no Wireshark

Para fazer o controlo de congestão, o TCP mantém uma janela de congestão que consiste numa estimativa do número de octetos que a rede consegue encaminhar, não enviando mais octetos do que o mínimo da janela definida pelo recetor e pela janela de congestão. A transferência de dados em simultâneo pode levar a uma queda na taxa de transmissão, uma vez que a taxa de transferência é distribuída de igual forma para cada ligação.

2183	8.336873	172.16.20.1	90.130.70.73	TCP	66 60465 → 22734 [ACK] Seq=1 Ack=3145730 Win=1969280 Len=0 TSval=36543194 TSecr=1382265
2184	8.338375	172.16.20.1	90.130.70.73	FTP	72 Request: QUIT
2185	8.338396	172.16.20.1	90.130.70.73	TCP	66 60465 → 22734 [FIN, ACK] Seq=1 Ack=3145730 Win=1969280 Len=0 TSval=36543195 TSecr=13
2186	8.338415	172.16.20.1	90.130.70.73	TCP	66 59971 → 21 [FIN, ACK] Seq=62 Ack=230 Win=29312 Len=0 TSval=36543195 TSecr=1382265583
2187	8.387708	90.130.70.73	172.16.20.1	FTP	90 Response: 226 Transfer complete.
2188	8.387746	172.16.20.1	90.130.70.73	TCP	54 59971 → 21 [RST] Seq=56 Win=0 Len=0
2189	8.389319	90.130.70.73	172.16.20.1	FTP	80 Response: 221 Goodbye.
2190	8.389333	172.16.20.1	90.130.70.73	TCP	54 59971 → 21 [RST] Seq=63 Win=0 Len=0
2191	8.389340	90.130.70.73	172.16.20.1	TCP	66 21 → 59971 [FIN, ACK] Seq=268 Ack=63 Win=14592 Len=0 TSval=1382265778 TSecr=3654319
2192	8.389347	172.16.20.1	90.130.70.73	TCP	54 59971 → 21 [RST] Seq=63 Win=0 Len=0
2193	8.391459	90.130.70.73	172.16.20.1	TCP	66 22734 → 60465 [ACK] Seq=3145730 Ack=2 Win=14592 Len=0 TSval=1382265778 TSecr=3654319
2194	10.024133	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60 Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
2195	11.064402	Cisco_5c:4d:83	Cisco_5c:4d:83	LOOP	60 Reply

Figura 9: Dados FTP no Wireshark - fim do processo com 1 download

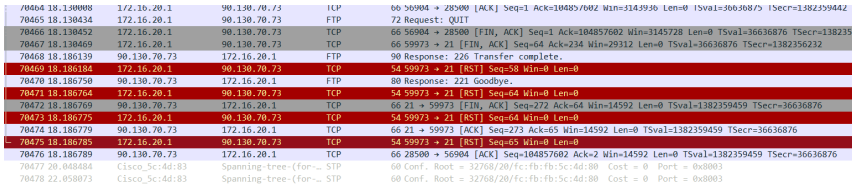


Figura 10: Dados FTP no Wireshark - fim do processo com 2 downloads em simultâneo - tux21 e tux22

5 Conclusão

De modo geral, foram atingidos os principais objetivos deste projeto. Conseguimos interiorizar os conceitos abordados ao longo trabalho e elaborar código eficiente, resultando numa implementação estável e coerente com o que era pedido no guião.

A implementação do cliente FTP (File Transfer Protocol) foi concluída com sucesso, o cliente é capaz de fazer download de diferentes tipos de ficheiros e tamanhos diferentes. Através desta implementação, o grupo entendeu melhor este protocolo e percebeu, também como funciona um protocolo oficial comparando ao protocolo desenvolvido no primeiro projeto.

Em relação à configuração de rede, também foi concluída com sucesso. Os elementos do grupo ficaram assim a perceber como funciona, a um nível aprofundado, a configuração de uma rede.

A Código fonte

A.1 TCP.h

```
1  #ifndef TCP_H
2  #define TCP_H
3  #include "url_parsing.h"
4
5
6  #define MAX_SIZE 516
7
8  typedef struct
9  {
10     int ctrl_socket_fd; // file descriptor to control socket
11     int data_socket_fd; // file descriptor to data socket
12 } ftp;
13
14 int initConnection(ftp* ftp,char* address, int port);
15 int endConnection(ftp ftp);
16
17 int initSocket(char* ip_address,int port);
18 int socketRead(int socketfd, char* repply);
19 int socketWrite(int socketfd, char* cmd);
20
21 void login(ftp ftp, url url);
22 void passiveMode(ftp ftp, char* ip_adress, int* port);
23 void retrieve(ftp ftp, url url);
24 int download(ftp ftp, url url);
25
26
27 #endif
```

TCP.h

A.2 TCP.c

```
1  #include <stdio.h>
2  #include <arpa/inet.h>
3  #include <stdlib.h>
4  #include <unistd.h>
5  #include <netdb.h>
6  #include <string.h>
7
8  #include "TCP.h"
9
10 #define READ 1
11 #define NO_READ 0
12
13
14 int initConnection(ftp* ftp,char* ip_address, int port){
15   int socketfd;
16
17   if ((socketfd = initSocket(ip_address, port)) < 0) {
18         printf("ERROR: Cannot connect socket.\n");
19         return 1;
20     }
21
22     ftp->ctrl_socket_fd = socketfd;
23   ftp->data_socket_fd = 0;
24
25   return 0;
26
27 }
28
29 int initSocket(char* ip_address, int port){
30
31   int   socketfd;
32   struct    sockaddr_in server_addr;
33
34   /*server address handling*/
35   bzero((char*)&server_addr,sizeof(server_addr));
36   server_addr.sin_family = AF_INET;
37   server_addr.sin_addr.s_addr = inet_addr(ip_address);  /*32 bit
         Internet address network byte ordered*/
38   server_addr.sin_port = htons(port);       /*server TCP port must
         be network byte ordered */
39
40   /*open an TCP socket*/
41   if ((socketfd = socket(AF_INET,SOCK_STREAM,0)) < 0) {
42         perror("socket()");
43           return -1;
44       }
45   /*connect to the server*/
46       if(connect(socketfd,
47              (struct sockaddr *)&server_addr,
48        sizeof(server_addr)) < 0){
49           perror("connect()");
50     return -1;
51   }
52   return socketfd;
53
54 }
55
56 void login(ftp ftp, url url){
57   char usr_cmd[MAX_SIZE];
58   char pwd_cmd[MAX_SIZE];
59
60   socketRead(ftp.ctrl_socket_fd, NULL);
61
```

```
62    sprintf(usr_cmd, "USER %s\r\n", url.user);
63    printf(">%s",usr_cmd);
64
65    socketWrite(ftp.ctrl_socket_fd,usr_cmd);
66    socketRead(ftp.ctrl_socket_fd,NULL);
67
68    sprintf(pwd_cmd, "PASS %s\r\n", url.password);
69    printf(">%s",pwd_cmd);
70
71    socketWrite(ftp.ctrl_socket_fd, pwd_cmd);
72    if(socketRead(ftp.ctrl_socket_fd,NULL) !=0){
73      fprintf(stderr, "Wrong credentials. Exiting...\n");
74      exit(1);
75    }
76  }
77
78  void passiveMode(ftp ftp, char* ip_adress, int* port){
79    char repply[MAX_SIZE];
80
81    socketWrite(ftp.ctrl_socket_fd, "PASV\r\n");
82    if(socketRead(ftp.ctrl_socket_fd,repply) !=0){
83      fprintf(stderr, "Error entering passive mode. Exiting...\n");
84      exit(1);
85    }
86
87    int values[6];
88    char* data = strchr(repply, '(');
89    sscanf(data, "(%d, %d, %d, %d, %d, %d)", &values[0],&values[1],&
          values[2],&values[3],&values[4],&values[5]);
90    sprintf(ip_adress, "%d.%d.%d.%d", values[0],values[1],values[2],
          values[3]);
91    *port = values[4]*256+values[5];
92  }
93
94  void retrieve(ftp ftp, url url){
95    char cmd[MAX_SIZE];
96
97    sprintf(cmd, "RETR %s%s\r\n", url.file_path, url.file_name);
98    printf(">%s",cmd);
99    socketWrite(ftp.ctrl_socket_fd, cmd);
100
101   if(socketRead(ftp.ctrl_socket_fd,NULL) != 0){
102     fprintf(stderr, "Error retrieving file. Exiting...\n");
103     exit(1);
104   }
105 }
106
107 int socketRead(int socketfd, char* repply){
108   FILE* fp = fdopen(socketfd, "r");
109   int allocated = 0;
110
111   if(repply == NULL){
112     repply = (char*) malloc(sizeof(char) * MAX_SIZE);
113     allocated = 1;
114   }
115
116   do {
117     memset(repply, 0, MAX_SIZE);
118     repply = fgets(repply, MAX_SIZE, fp);
119     printf("<%s", repply);
120   } while (!('1' <= repply[0] && repply[0] <= '5') || repply[3] !=
        ' ');
121
122   char r0= repply[0];
123
```

```
124   if(allocated)
125     free(repply);
126
127   return (r0>'4');
128 }
129
130 int socketWrite(int socketfd, char* cmd){
131
132     int ret = write(socketfd, cmd, strlen(cmd));
133     return ret;
134 }
135
136 int download(ftp ftp, url url){
137   FILE* dest_file;
138   if(!(dest_file = fopen(url.file_name, "w"))) {
139         printf("Error opening file %s.\n",url.file_name);
140         return 1;
141     }
142
143   char buf[1024];
144   int bytes;
145   while ((bytes = read(ftp.data_socket_fd, buf, sizeof(buf)))) {
146     if (bytes < 0) {
147       fprintf(stderr, "Error, nothing was received from data socket
              fd.\n");
148       return 1;
149     }
150
151     if ((bytes = fwrite(buf, bytes, 1, dest_file)) < 0) {
152       fprintf(stderr, "Error, cannot write data in file.\n");
153       return 1;
154     }
155   }
156
157   fclose(dest_file);
158
159   printf("Finished downloading file\n");
160
161   return 0;
162 }
163
164
165 int endConnection(ftp ftp){
166
167   printf("Closing connection\n");
168   socketWrite(ftp.ctrl_socket_fd,"QUIT\r\n");
169
170   if(socketRead(ftp.ctrl_socket_fd,NULL) != 0){
171     fprintf(stderr, "Error closing connection. Closing...\n");
172     close(ftp.data_socket_fd);
173     close(ftp.ctrl_socket_fd);
174     exit(1);
175   }
176
177   close(ftp.data_socket_fd);
178   close(ftp.ctrl_socket_fd);
179
180   printf("Ending conection! Come back soon\n");
181
182   return 0;
183 }
```

TCP.c

A.3 url_parsing.h

```
#ifndef URL_PARSING_H
#define URL_PARSING_H
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdlib.h>
#include <unistd.h>
#include <signal.h>
#include <netdb.h>
#include <string.h>
#include <strings.h>
#include <termios.h>
#include <fcntl.h>



typedef struct{
  char  user[256];
  char  password[256];
  char  host[256];
  char  file_path[256];
  char  file_name[256];
  char  ip[256];
} url;


int userPassword(url * info_struct, char * complete_url);
int parse_url(char complete_url[],url * info_struct);
int getIp(url* info_struct);
#endif
```

url_parsing.h

A.4 url_parsing.c

```
1  #include "url_parsing.h"
2
3
4
5  int userPassword(url * info_struct, char * complete_url){
6    char * at = strrchr(complete_url, '@');
7    char* first_slash = strchr(complete_url, '/'); //slash is never
         null
8    first_slash += 2; // first_slash * its increased by two to point
         to the beggining of the user name "//[username]"
9    char* password = strchr(first_slash, ':');
10   if(password == NULL){
11     fprintf(stderr, "Your link must contain a ':' separating the
           username and password!'\n");
12     return 1;
13   }
14   memcpy(info_struct->user, first_slash, password - first_slash);
         //password - slash it's the size of username in bytes
15   info_struct->user[password-first_slash]=0;
16   password++; //the password pointer was poiting to ":" and it has
         to point to the first character of the userPassword
17   memcpy(info_struct->password,password,at - password);
18   info_struct->password[at-password] = 0; //string end character
19   return 0;
20 }
21
22 int parse_url(char complete_url[],url * info_struct){
23   if(strncmp(complete_url, "ftp://", strlen("ftp://")) != 0){
24     fprintf(stderr, "The link does not begin with 'ftp://'\n");
25     return 1;
26 }
27
28   char* slash_after_host;
29
30   if(!strchr(complete_url, '@')){
31     memcpy(info_struct->user, "anonymous", strlen("anonymous") + 1)
           ;
32     memcpy(info_struct->password, "anonymous", strlen("anonymous")
           + 1);
33
34     char * s1 = strchr(complete_url,'/');
35     s1++;
36     s1++;
37
38     slash_after_host = strchr(s1, '/');
39     memcpy(info_struct->host, s1, slash_after_host-s1);
40     info_struct->host[slash_after_host-s1] = 0;
41   }
42   else{
43     if(userPassword(info_struct,complete_url)!=0)
44       return 1;
45
46       char * at = strrchr(complete_url, '@');
47       at++;
48
49       slash_after_host = strchr(at, '/');
50       memcpy(info_struct->host, at, slash_after_host-at);
51       info_struct->host[slash_after_host-at] = 0;
52
53   }
54
55
56   char* last_slash = strrchr(complete_url, '/');
```

```
57    last_slash++; //to point to the element after the slash
58    memcpy(info_struct->file_path, slash_after_host, last_slash-
          slash_after_host);
59    info_struct->file_path[last_slash-slash_after_host] = 0;
60
61    memcpy(info_struct->file_name, last_slash, strlen(last_slash) +
          1);
62
63    getIp(info_struct);
64
65    return 0;
66  }
67
68
69  int getIp(url* info_struct) {
70      struct hostent* h;
71
72      if ((h = gethostbyname(info_struct->host)) == NULL) {
73          herror("gethostbyname");
74          return 1;
75      }
76
77
78      char* ip = inet_ntoa(*((struct in_addr *) h->h_addr));
79      strcpy(info_struct->ip, ip);
80
81      return 0;
82  }
```

url_parsing.c

A.5 makefile

```
all: main.c url_parsing.c url_parsing.h TCP.c TCP.h
    gcc -Wall main.c url_parsing.c TCP.c -o clientDownload
```

makefile

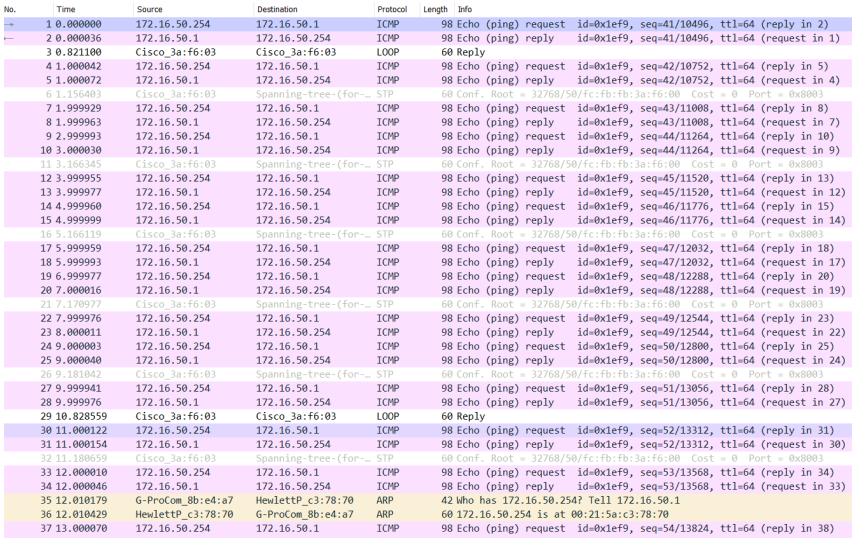
A.6 main.c

```
#include <stdio.h>
#include <arpa/inet.h>
#include <stdlib.h>
#include <unistd.h>
#include <netdb.h>
#include <string.h>

#include "url_parsing.h"
#include "TCP.h"

#define PORT 21

int main(int argc, char** argv){

  if(argc != 2){
    fprintf(stderr, "Usage: %s <address>\n", argv[0]);
    exit(1);
  }

  url url;
  ftp ftp;

  if(parse_url(argv[1], &url) != 0){
    fprintf(stderr, "Invalid URL\n");
    exit(1);
  }

  printf("user:%s\n", url.user);
  printf("pass:%s\n", url.password);
  printf("ip:%s\n", url.ip);
  printf("path:%s\n", url.file_path);
  printf("file_name:%s\n", url.file_name);
  printf("host:%s\n", url.host);
  printf("\n\n");

  if(initConnection(&ftp,url.ip,PORT) !=0){
    fprintf(stderr, "Error opening control connection\n");
    exit(1);
  }

  login(ftp,url);

  char ip_address[MAX_SIZE];
  int port;

  passiveMode(ftp, ip_address, &port);

  if ((ftp.data_socket_fd = initSocket(ip_address,port))<0){
    fprintf(stderr, "Error opening data connection\n");
    exit(1);
  }

  retrieve(ftp,url);
  download(ftp,url);
```

```
55      endConnection(ftp);
56
57      return 0;
58
59      }
```

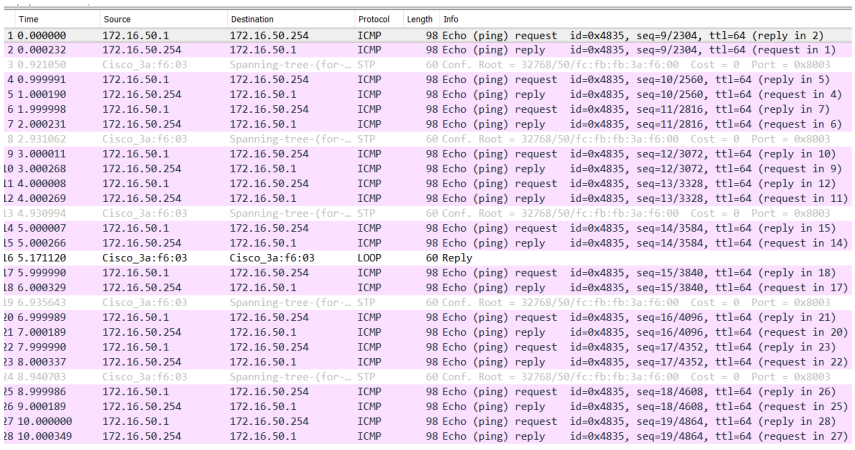
Main

A.7 Wireshark Logs



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) request id=0x1ef9, seq=41/10496, ttl=64 (reply in 2)
2	0.000036	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) reply id=0x1ef9, seq=41/10496, ttl=64 (request in 1)
3	0.821100	Cisco_3a:f6:03	Cisco_3a:f6:03	LOOP	60	Reply
4	1.000042	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) request id=0x1ef9, seq=42/10752, ttl=64 (reply in 5)
5	1.000072	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) reply id=0x1ef9, seq=42/10752, ttl=64 (request in 4)
6	1.156403	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
7	1.999929	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) request id=0x1ef9, seq=43/11008, ttl=64 (reply in 8)
8	1.999963	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) reply id=0x1ef9, seq=43/11008, ttl=64 (request in 7)
9	2.999993	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) request id=0x1ef9, seq=44/11264, ttl=64 (reply in 10)
10	3.000030	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) reply id=0x1ef9, seq=44/11264, ttl=64 (request in 9)
11	3.166345	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
12	3.999955	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) request id=0x1ef9, seq=45/11520, ttl=64 (reply in 13)
13	3.999977	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) reply id=0x1ef9, seq=45/11520, ttl=64 (request in 12)
14	4.999960	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) request id=0x1ef9, seq=46/11776, ttl=64 (reply in 15)
15	4.999999	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) reply id=0x1ef9, seq=46/11776, ttl=64 (request in 14)
16	5.166119	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
17	5.999959	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) request id=0x1ef9, seq=47/12032, ttl=64 (reply in 18)
18	5.999993	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) reply id=0x1ef9, seq=47/12032, ttl=64 (request in 17)
19	6.999977	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) request id=0x1ef9, seq=48/12288, ttl=64 (reply in 20)
20	7.000016	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) reply id=0x1ef9, seq=48/12288, ttl=64 (request in 19)
21	7.170977	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
22	7.999976	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) request id=0x1ef9, seq=49/12544, ttl=64 (reply in 23)
23	8.000011	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) reply id=0x1ef9, seq=49/12544, ttl=64 (request in 22)
24	9.000003	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) request id=0x1ef9, seq=50/12800, ttl=64 (reply in 25)
25	9.000040	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) reply id=0x1ef9, seq=50/12800, ttl=64 (request in 24)
26	9.181042	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
27	9.999941	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) request id=0x1ef9, seq=51/13056, ttl=64 (reply in 28)
28	9.999976	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) reply id=0x1ef9, seq=51/13056, ttl=64 (request in 27)
29	10.828559	Cisco_3a:f6:03	Cisco_3a:f6:03	LOOP	60	Reply
30	11.000122	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) request id=0x1ef9, seq=52/13312, ttl=64 (reply in 31)
31	11.000154	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) reply id=0x1ef9, seq=52/13312, ttl=64 (request in 30)
32	11.180659	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
33	12.000010	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) request id=0x1ef9, seq=53/13568, ttl=64 (reply in 34)
34	12.000046	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) reply id=0x1ef9, seq=53/13568, ttl=64 (request in 33)
35	12.010179	G-ProCom_8b:e4:a7	HewlettP_c3:78:70	ARP	42	Who has 172.16.50.254? Tell 172.16.50.1
36	12.010429	HewlettP_c3:78:70	G-ProCom_8b:e4:a7	ARP	60	172.16.50.254 is at 00:21:5a:c3:78:70
37	13.000070	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) request id=0x1ef9, seq=54/13824, ttl=64 (reply in 38)

Figura 11: Experiência 1



	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x4835, seq=9/2304, ttl=64 (reply in 2)
2	0.000232	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x4835, seq=9/2304, ttl=64 (request in 1)
3	0.921050	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
4	0.999991	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x4835, seq=10/2560, ttl=64 (reply in 5)
5	1.000190	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x4835, seq=10/2560, ttl=64 (request in 4)
6	1.999998	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x4835, seq=11/2816, ttl=64 (reply in 7)
7	2.000231	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x4835, seq=11/2816, ttl=64 (request in 6)
8	2.931062	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
9	3.000011	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x4835, seq=12/3072, ttl=64 (reply in 10)
10	3.000268	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x4835, seq=12/3072, ttl=64 (request in 9)
11	4.000008	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x4835, seq=13/3328, ttl=64 (reply in 12)
12	4.000269	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x4835, seq=13/3328, ttl=64 (request in 11)
13	4.930994	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
14	5.000007	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x4835, seq=14/3584, ttl=64 (reply in 15)
15	5.000266	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x4835, seq=14/3584, ttl=64 (request in 14)
16	5.171120	Cisco_3a:f6:03	Cisco_3a:f6:03	LOOP	60	Reply
17	5.999990	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x4835, seq=15/3840, ttl=64 (reply in 18)
18	6.000329	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x4835, seq=15/3840, ttl=64 (request in 17)
19	6.935643	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
20	6.999989	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x4835, seq=16/4096, ttl=64 (reply in 21)
21	7.000189	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x4835, seq=16/4096, ttl=64 (request in 20)
22	7.999990	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x4835, seq=17/4352, ttl=64 (reply in 23)
23	8.000337	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x4835, seq=17/4352, ttl=64 (request in 22)
24	8.940703	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
25	8.999986	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x4835, seq=18/4608, ttl=64 (reply in 26)
26	9.000189	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x4835, seq=18/4608, ttl=64 (request in 25)
27	10.000000	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x4835, seq=19/4864, ttl=64 (reply in 28)
28	10.000349	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x4835, seq=19/4864, ttl=64 (request in 27)

Figura 12: Experiência 2 - ponto 6

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
2	2.004902	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
3	4.009698	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
4	6.014593	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
5	6.501272	Cisco_3a:f6:03	Cisco_3a:f6:03	LOOP	60	Reply
6	8.019351	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
7	10.024276	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
8	12.029200	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
9	14.033977	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
10	16.066067	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
11	16.500440	Cisco_3a:f6:03	Cisco_3a:f6:03	LOOP	60	Reply
12	18.068805	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
13	20.073674	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
14	22.078556	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
15	24.083682	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
16	26.088319	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
17	26.507903	Cisco_3a:f6:03	Cisco_3a:f6:03	LOOP	60	Reply
18	28.093177	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
19	30.097974	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
20	32.102805	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
21	34.107663	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
22	36.112597	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
23	36.506940	Cisco_3a:f6:03	Cisco_3a:f6:03	LOOP	60	Reply
24	38.117425	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
25	40.122277	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
26	42.127101	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
27	44.131960	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
28	46.136941	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
29	46.519445	Cisco_3a:f6:03	Cisco_3a:f6:03	LOOP	60	Reply
30	48.141624	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003
31	50.146543	Cisco_3a:f6:03	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8003

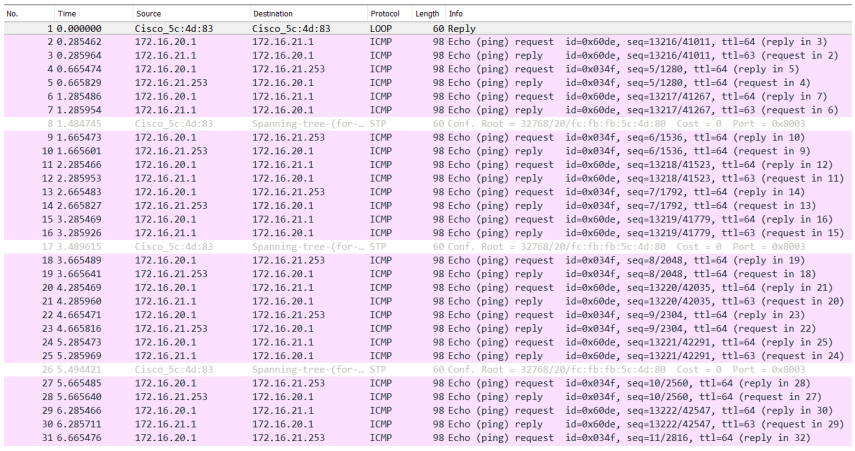
Figura 13: Experiência 2 - ponto 10 - tux1

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
2	0.486631	Cisco_3a:f6:05	Cisco_3a:f6:05	LOOP	60	Reply
3	2.004799	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
4	4.009656	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
5	6.015204	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
6	8.019319	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
7	10.052366	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
8	10.485723	Cisco_3a:f6:05	Cisco_3a:f6:05	LOOP	60	Reply
9	12.054172	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
10	14.059092	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
11	16.063968	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
12	18.069001	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
13	20.073653	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
14	20.493103	Cisco_3a:f6:05	Cisco_3a:f6:05	LOOP	60	Reply
15	22.078463	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
16	24.083394	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
17	26.088062	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
18	28.092902	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
19	30.097913	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
20	30.492211	Cisco_3a:f6:05	Cisco_3a:f6:05	LOOP	60	Reply
21	32.102636	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
22	34.107595	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
23	36.112285	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
24	38.117125	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
25	40.122365	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
26	40.504692	Cisco_3a:f6:05	Cisco_3a:f6:05	LOOP	60	Reply
27	42.127078	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
28	42.643731	Cisco_3a:f6:05	CDP/VTP/DTP/PAgP/UD…	CDP	435	Device ID: tux-sw5 Port ID: FastEthernet0/3
29	44.131747	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
30	46.136668	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
31	48.141330	Cisco_3a:f6:05	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005

Figura 14: Experiência 2 - ponto 10 - tux1

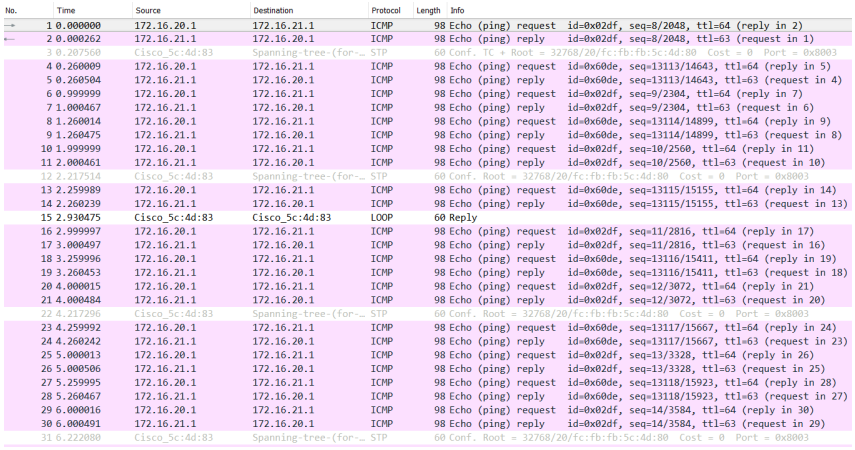
No.	Time	Source	Destination	Protocol	Length	Info
→ 1	0.000000	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13183/32563, ttl=64 (reply in 2)
← 2	0.000499	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13183/32563, ttl=63 (request in 1)
3	0.117144	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
4	0.227997	172.16.20.1	172.16.20.254	ICMP	98	Echo (ping) request id=0x0326, seq=6/1536, ttl=64 (reply in 5)
5	0.228197	172.16.20.254	172.16.20.1	ICMP	98	Echo (ping) reply id=0x0326, seq=6/1536, ttl=64 (request in 4)
6	1.000004	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13184/32819, ttl=64 (reply in 7)
7	1.000493	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13184/32819, ttl=63 (request in 6)
8	1.227987	172.16.20.1	172.16.20.254	ICMP	98	Echo (ping) request id=0x0326, seq=7/1792, ttl=64 (reply in 9)
9	1.228141	172.16.20.254	172.16.20.1	ICMP	98	Echo (ping) reply id=0x0326, seq=7/1792, ttl=64 (request in 8)
10	2.000003	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13185/33075, ttl=64 (reply in 11)
11	2.000474	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13185/33075, ttl=63 (request in 10)
12	2.121815	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
13	2.228002	172.16.20.1	172.16.20.254	ICMP	98	Echo (ping) request id=0x0326, seq=8/2048, ttl=64 (reply in 14)
14	2.228200	172.16.20.254	172.16.20.1	ICMP	98	Echo (ping) reply id=0x0326, seq=8/2048, ttl=64 (request in 13)
15	2.700701	Cisco_5c:4d:83	Cisco_5c:4d:83	LOOP	60	Reply
16	2.999993	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13186/33331, ttl=64 (reply in 17)
17	3.000495	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13186/33331, ttl=63 (request in 16)
18	3.005457	HewlettP_a6:a4:f1	G-ProCom_8c:af:9d	ARP	60	Who has 172.16.20.1? Tell 172.16.20.254
19	3.005469	G-ProCom_8c:af:9d	HewlettP_a6:a4:f1	ARP	42	172.16.20.1 is at 00:0f:fe:8c:af:9d
20	3.228042	172.16.20.1	172.16.20.254	ICMP	98	Echo (ping) request id=0x0326, seq=9/2304, ttl=64 (reply in 21)
21	3.228399	172.16.20.254	172.16.20.1	ICMP	98	Echo (ping) reply id=0x0326, seq=9/2304, ttl=64 (request in 20)
22	4.000004	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13187/33587, ttl=64 (reply in 23)
23	4.000483	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13187/33587, ttl=63 (request in 22)
24	4.126703	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
25	4.228090	172.16.20.1	172.16.20.254	ICMP	98	Echo (ping) request id=0x0326, seq=10/2560, ttl=64 (reply in 26)
26	4.228290	172.16.20.254	172.16.20.1	ICMP	98	Echo (ping) reply id=0x0326, seq=10/2560, ttl=64 (request in 25)
27	4.467947	Cisco_5c:4d:83	CDP/VTP/DTP/PAgP/UD…	CDP	435	Device ID: tux-sw2 Port ID: FastEthernet0/1
28	4.999996	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13188/33843, ttl=64 (reply in 29)
29	5.000489	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13188/33843, ttl=63 (request in 28)
30	5.228014	172.16.20.1	172.16.20.254	ICMP	98	Echo (ping) request id=0x0326, seq=11/2816, ttl=64 (reply in 31)
31	5.228184	172.16.20.254	172.16.20.1	ICMP	98	Echo (ping) reply id=0x0326, seq=11/2816, ttl=64 (request in 30)
32	5.999990	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13189/34099, ttl=64 (reply in 33)
33	6.000478	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13189/34099, ttl=63 (request in 32)

Figura 15: Experiência 3 - tux1



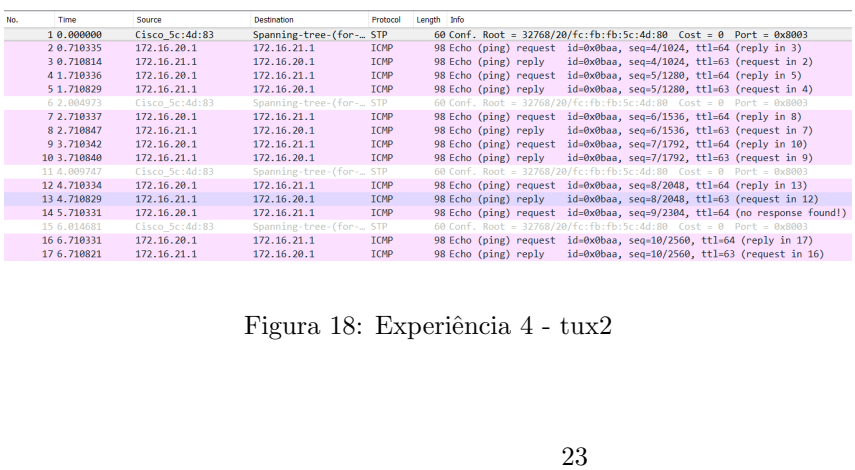
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Cisco_5c:4d:83	Cisco_5c:4d:83	LOOP	60	Reply
2	0.285462	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13216/41011, ttl=64 (reply in 3)
3	0.285964	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13216/41011, ttl=63 (request in 2)
4	0.665474	172.16.20.1	172.16.21.253	ICMP	98	Echo (ping) request id=0x034f, seq=5/1280, ttl=64 (reply in 5)
5	0.665829	172.16.21.253	172.16.20.1	ICMP	98	Echo (ping) reply id=0x034f, seq=5/1280, ttl=64 (request in 4)
6	1.285486	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13217/41267, ttl=64 (reply in 7)
7	1.285954	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13217/41267, ttl=63 (request in 6)
8	1.484745	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
9	1.665473	172.16.20.1	172.16.21.253	ICMP	98	Echo (ping) request id=0x034f, seq=6/1536, ttl=64 (reply in 10)
10	1.665601	172.16.21.253	172.16.20.1	ICMP	98	Echo (ping) reply id=0x034f, seq=6/1536, ttl=64 (request in 9)
11	2.285466	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13218/41523, ttl=64 (reply in 12)
12	2.285953	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13218/41523, ttl=63 (request in 11)
13	2.665483	172.16.20.1	172.16.21.253	ICMP	98	Echo (ping) request id=0x034f, seq=7/1792, ttl=64 (reply in 14)
14	2.665827	172.16.21.253	172.16.20.1	ICMP	98	Echo (ping) reply id=0x034f, seq=7/1792, ttl=64 (request in 13)
15	3.285469	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13219/41779, ttl=64 (reply in 16)
16	3.285926	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13219/41779, ttl=63 (request in 15)
17	3.489615	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
18	3.665489	172.16.20.1	172.16.21.253	ICMP	98	Echo (ping) request id=0x034f, seq=8/2048, ttl=64 (reply in 19)
19	3.665641	172.16.21.253	172.16.20.1	ICMP	98	Echo (ping) reply id=0x034f, seq=8/2048, ttl=64 (request in 18)
20	4.285469	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13220/42035, ttl=64 (reply in 21)
21	4.285960	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13220/42035, ttl=63 (request in 20)
22	4.665471	172.16.20.1	172.16.21.253	ICMP	98	Echo (ping) request id=0x034f, seq=9/2304, ttl=64 (reply in 23)
23	4.665816	172.16.21.253	172.16.20.1	ICMP	98	Echo (ping) reply id=0x034f, seq=9/2304, ttl=64 (request in 22)
24	5.285473	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13221/42291, ttl=64 (reply in 25)
25	5.285969	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13221/42291, ttl=63 (request in 24)
26	5.494421	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
27	5.665485	172.16.20.1	172.16.21.253	ICMP	98	Echo (ping) request id=0x034f, seq=10/2560, ttl=64 (reply in 28)
28	5.665640	172.16.21.253	172.16.20.1	ICMP	98	Echo (ping) reply id=0x034f, seq=10/2560, ttl=64 (request in 27)
29	6.285466	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13222/42547, ttl=64 (reply in 30)
30	6.285711	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13222/42547, ttl=63 (request in 29)
31	6.665476	172.16.20.1	172.16.21.253	ICMP	98	Echo (ping) request id=0x034f, seq=11/2816, ttl=64 (reply in 32)

Figura 16: Experiência 3 - tux2



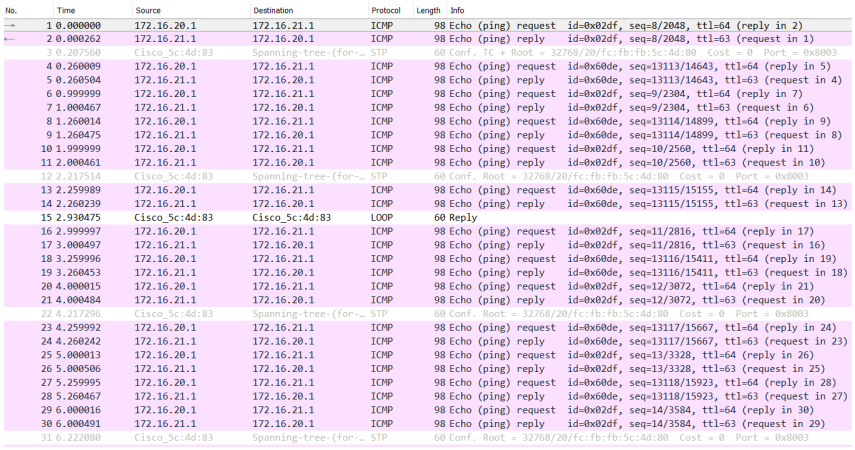
No.	Time	Source	Destination	Protocol	Length	Info
→ 1	0.000000	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x02df, seq=8/2048, ttl=64 (reply in 2)
← 2	0.000262	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x02df, seq=8/2048, ttl=63 (request in 1)
3	0.207560	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. TC + Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
4	0.260009	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13113/14643, ttl=64 (reply in 5)
5	0.260504	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13113/14643, ttl=63 (request in 4)
6	0.999999	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x02df, seq=9/2304, ttl=64 (reply in 7)
7	1.000467	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x02df, seq=9/2304, ttl=63 (request in 6)
8	1.260014	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13114/14899, ttl=64 (reply in 9)
9	1.260475	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13114/14899, ttl=63 (request in 8)
10	1.999999	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x02df, seq=10/2560, ttl=64 (reply in 11)
11	2.000461	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x02df, seq=10/2560, ttl=63 (request in 10)
12	2.217514	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
13	2.259989	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13115/15155, ttl=64 (reply in 14)
14	2.260239	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13115/15155, ttl=63 (request in 13)
15	2.930475	Cisco_5c:4d:83	Cisco_5c:4d:83	LOOP	60	Reply
16	2.999997	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x02df, seq=11/2816, ttl=64 (reply in 17)
17	3.000497	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x02df, seq=11/2816, ttl=63 (request in 16)
18	3.259996	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13116/15411, ttl=64 (reply in 19)
19	3.260453	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13116/15411, ttl=63 (request in 18)
20	4.000015	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x02df, seq=12/3072, ttl=64 (reply in 21)
21	4.000484	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x02df, seq=12/3072, ttl=63 (request in 20)
22	4.217296	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
23	4.259992	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13117/15667, ttl=64 (reply in 24)
24	4.260242	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13117/15667, ttl=63 (request in 23)
25	5.000013	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x02df, seq=13/3328, ttl=64 (reply in 26)
26	5.000506	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x02df, seq=13/3328, ttl=63 (request in 25)
27	5.259995	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13118/15923, ttl=64 (reply in 28)
28	5.260467	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13118/15923, ttl=63 (request in 27)
29	6.000016	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x02df, seq=14/3584, ttl=64 (reply in 30)
30	6.000491	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x02df, seq=14/3584, ttl=63 (request in 29)
31	6.222080	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003

Figura 17: Experiência 3 - tux3



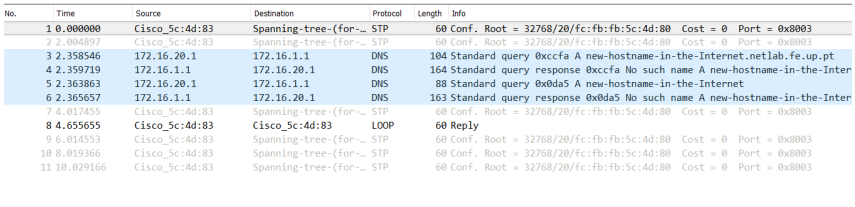
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
2	0.710335	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x0baa, seq=4/1024, ttl=64 (reply in 3)
3	0.710814	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x0baa, seq=4/1024, ttl=63 (request in 2)
4	1.710336	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x0baa, seq=5/1280, ttl=64 (reply in 5)
5	1.710829	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x0baa, seq=5/1280, ttl=63 (request in 4)
6	2.004973	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
7	2.710337	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x0baa, seq=6/1536, ttl=64 (reply in 8)
8	2.710847	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x0baa, seq=6/1536, ttl=63 (request in 7)
9	3.710342	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x0baa, seq=7/1792, ttl=64 (reply in 10)
10	3.710840	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x0baa, seq=7/1792, ttl=63 (request in 9)
11	4.009747	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
12	4.710334	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x0baa, seq=8/2048, ttl=64 (reply in 13)
13	4.710829	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x0baa, seq=8/2048, ttl=63 (request in 12)
14	5.710331	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x0baa, seq=9/2304, ttl=64 (no response found!)
15	6.014681	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
16	6.710331	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x0baa, seq=10/2560, ttl=64 (reply in 17)
17	6.710821	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x0baa, seq=10/2560, ttl=63 (request in 16)

Figura 18: Experiência 4 - tux2



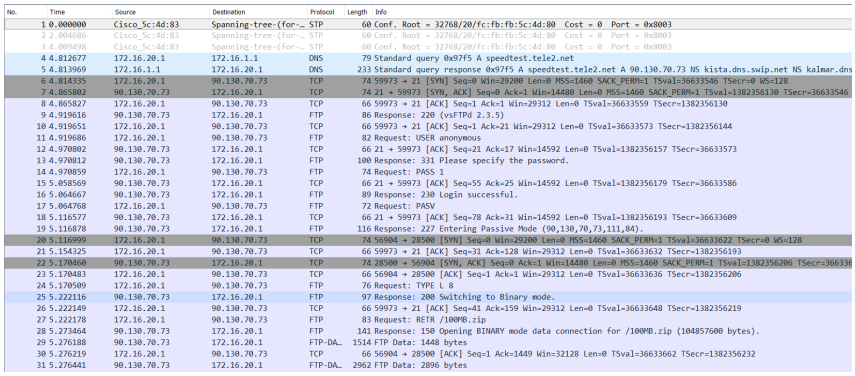
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x02df, seq=8/2048, ttl=64 (reply in 2)
2	0.000262	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x02df, seq=8/2048, ttl=63 (request in 1)
3	0.207560	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. TC + Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
4	0.260009	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13113/14643, ttl=64 (reply in 5)
5	0.260504	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13113/14643, ttl=63 (request in 4)
6	0.999999	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x02df, seq=9/2304, ttl=64 (reply in 7)
7	1.000467	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x02df, seq=9/2304, ttl=63 (request in 6)
8	1.260014	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13114/14899, ttl=64 (reply in 9)
9	1.260475	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13114/14899, ttl=63 (request in 8)
10	1.999999	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x02df, seq=10/2560, ttl=64 (reply in 11)
11	2.000461	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x02df, seq=10/2560, ttl=63 (request in 10)
12	2.217514	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
13	2.259989	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13115/15155, ttl=64 (reply in 14)
14	2.260239	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13115/15155, ttl=63 (request in 13)
15	2.930475	Cisco_5c:4d:83	Cisco_5c:4d:83	LOOP	60	Reply
16	2.999997	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x02df, seq=11/2816, ttl=64 (reply in 17)
17	3.000497	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x02df, seq=11/2816, ttl=63 (request in 16)
18	3.259996	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13116/15411, ttl=64 (reply in 19)
19	3.260453	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13116/15411, ttl=63 (request in 18)
20	4.000015	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x02df, seq=12/3072, ttl=64 (reply in 21)
21	4.000484	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x02df, seq=12/3072, ttl=63 (request in 20)
22	4.217296	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
23	4.259992	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13117/15667, ttl=64 (reply in 24)
24	4.260242	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13117/15667, ttl=63 (request in 23)
25	5.000013	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x02df, seq=13/3328, ttl=64 (reply in 26)
26	5.000506	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x02df, seq=13/3328, ttl=63 (request in 25)
27	5.259995	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x60de, seq=13118/15923, ttl=64 (reply in 28)
28	5.260467	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x60de, seq=13118/15923, ttl=63 (request in 27)
29	6.000016	172.16.20.1	172.16.21.1	ICMP	98	Echo (ping) request id=0x02df, seq=14/3584, ttl=64 (reply in 30)
30	6.000491	172.16.21.1	172.16.20.1	ICMP	98	Echo (ping) reply id=0x02df, seq=14/3584, ttl=63 (request in 29)
31	6.222080	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003

Figura 19: Experiência 4 - tux3



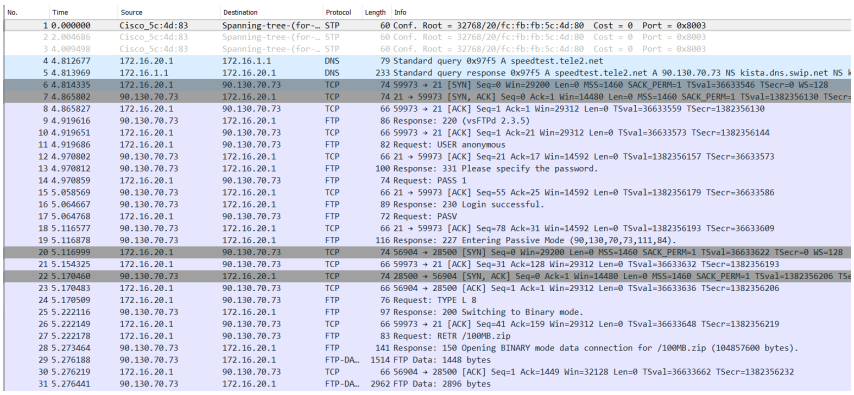
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
2	2.004897	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
3	2.358546	172.16.20.1	172.16.1.1	DNS	104	Standard query 0xccfa A new-hostname-in-the-Internet.netlab.fe.up.pt
4	2.359719	172.16.1.1	172.16.20.1	DNS	164	Standard query response 0xccfa No such name A new-hostname-in-the-Inter
5	2.363863	172.16.20.1	172.16.1.1	DNS	88	Standard query 0x0da5 A new-hostname-in-the-Internet
6	2.365657	172.16.1.1	172.16.20.1	DNS	163	Standard query response 0x0da5 No such name A new-hostname-in-the-Inter
7	4.017455	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
8	4.655655	Cisco_5c:4d:83	Cisco_5c:4d:83	LOOP	60	Reply
9	6.014553	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
10	8.019366	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
11	10.029166	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003

Figura 20: Experiência 5



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
2	2.004686	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
3	4.009498	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
4	4.812677	172.16.20.1	172.16.1.1	DNS	79	Standard query 0x97f5 A speedtest.tele2.net
5	4.813969	172.16.1.1	172.16.20.1	DNS	233	Standard query response 0x97f5 A speedtest.tele2.net A 90.130.70.73 NS kista.dns.swip.net NS kalmar.dns
6	4.814335	172.16.20.1	90.130.70.73	TCP	74	59973 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=36633546 TSecr=0 WS=128
7	4.865802	90.130.70.73	172.16.20.1	TCP	74	21 → 59973 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=1382356130 TSecr=36633546
8	4.865827	172.16.20.1	90.130.70.73	TCP	66	59973 → 21 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=36633559 TSecr=1382356130
9	4.919616	90.130.70.73	172.16.20.1	FTP	86	Response: 220 (vsFTPd 2.3.5)
10	4.919651	172.16.20.1	90.130.70.73	TCP	66	59973 → 21 [ACK] Seq=1 Ack=21 Win=29312 Len=0 TSval=36633573 TSecr=1382356144
11	4.919686	172.16.20.1	90.130.70.73	FTP	82	Request: USER anonymous
12	4.970802	90.130.70.73	172.16.20.1	TCP	66	21 → 59973 [ACK] Seq=21 Ack=17 Win=14592 Len=0 TSval=1382356157 TSecr=36633573
13	4.970812	90.130.70.73	172.16.20.1	FTP	100	Response: 331 Please specify the password.
14	4.970859	172.16.20.1	90.130.70.73	FTP	74	Request: PASS 1
15	5.058569	90.130.70.73	172.16.20.1	TCP	66	21 → 59973 [ACK] Seq=55 Ack=25 Win=14592 Len=0 TSval=1382356179 TSecr=36633586
16	5.064667	90.130.70.73	172.16.20.1	FTP	89	Response: 230 Login successful.
17	5.064768	172.16.20.1	90.130.70.73	FTP	72	Request: PASV
18	5.116577	90.130.70.73	172.16.20.1	TCP	66	21 → 59973 [ACK] Seq=78 Ack=31 Win=14592 Len=0 TSval=1382356193 TSecr=36633609
19	5.116878	90.130.70.73	172.16.20.1	FTP	116	Response: 227 Entering Passive Mode (90,130,70,73,111,84).
20	5.116999	172.16.20.1	90.130.70.73	TCP	74	56904 → 28500 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=36633622 TSecr=0 WS=128
21	5.154325	172.16.20.1	90.130.70.73	TCP	66	59973 → 21 [ACK] Seq=31 Ack=128 Win=29312 Len=0 TSval=36633632 TSecr=1382356193
22	5.170460	90.130.70.73	172.16.20.1	TCP	74	28500 → 56904 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=1382356206 TSecr=366336
23	5.170483	172.16.20.1	90.130.70.73	TCP	66	56904 → 28500 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=36633636 TSecr=1382356206
24	5.170509	172.16.20.1	90.130.70.73	FTP	76	Request: TYPE L 8
25	5.222116	90.130.70.73	172.16.20.1	FTP	97	Response: 200 Switching to Binary mode.
26	5.222149	172.16.20.1	90.130.70.73	TCP	66	59973 → 21 [ACK] Seq=41 Ack=159 Win=29312 Len=0 TSval=36633648 TSecr=1382356219
27	5.222178	172.16.20.1	90.130.70.73	FTP	83	Request: RETR /100MB.zip
28	5.273464	90.130.70.73	172.16.20.1	FTP	141	Response: 150 Opening BINARY mode data connection for /100MB.zip (104857600 bytes).
29	5.276188	90.130.70.73	172.16.20.1	FTP-DA…	1514	FTP Data: 1448 bytes
30	5.276219	172.16.20.1	90.130.70.73	TCP	66	56904 → 28500 [ACK] Seq=1 Ack=1449 Win=32128 Len=0 TSval=36633662 TSecr=1382356232
31	5.276441	90.130.70.73	172.16.20.1	FTP-DA…	2962	FTP Data: 2896 bytes

Figura 21: Experiência 6



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
2	2.004686	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
3	4.009498	Cisco_5c:4d:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/20/fc:fb:fb:5c:4d:80 Cost = 0 Port = 0x8003
4	4.812677	172.16.20.1	172.16.1.1	DNS	79	Standard query 0x97f5 A speedtest.tele2.net
5	4.813969	172.16.1.1	172.16.20.1	DNS	233	Standard query response 0x97f5 A speedtest.tele2.net A 90.130.70.73 NS kista.dns.swip.net NS k
6	4.814335	172.16.20.1	90.130.70.73	TCP	74	59973 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=36633546 TSecr=0 WS=128
7	4.865002	90.130.70.73	172.16.20.1	TCP	74	21 → 59973 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=1382356130 TSecr=
8	4.865827	172.16.20.1	90.130.70.73	TCP	66	59973 → 21 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=36633559 TSecr=1382356130
9	4.919616	90.130.70.73	172.16.20.1	FTP	86	Response: 220 (vsFTPd 2.3.5)
10	4.919651	172.16.20.1	90.130.70.73	TCP	66	59973 → 21 [ACK] Seq=1 Ack=21 Win=29312 Len=0 TSval=36633573 TSecr=1382356144
11	4.919686	172.16.20.1	90.130.70.73	FTP	82	Request: USER anonymous
12	4.970802	90.130.70.73	172.16.20.1	TCP	66	21 → 59973 [ACK] Seq=21 Ack=17 Win=14592 Len=0 TSval=1382356157 TSecr=36633573
13	4.970812	90.130.70.73	172.16.20.1	FTP	100	Response: 331 Please specify the password.
14	4.970859	172.16.20.1	90.130.70.73	FTP	74	Request: PASS 1
15	5.058569	90.130.70.73	172.16.20.1	TCP	66	21 → 59973 [ACK] Seq=55 Ack=25 Win=14592 Len=0 TSval=1382356179 TSecr=36633586
16	5.064667	90.130.70.73	172.16.20.1	FTP	89	Response: 230 Login successful.
17	5.064768	172.16.20.1	90.130.70.73	FTP	72	Request: PASV
18	5.116577	90.130.70.73	172.16.20.1	TCP	66	21 → 59973 [ACK] Seq=78 Ack=31 Win=14592 Len=0 TSval=1382356193 TSecr=36633609
19	5.116878	90.130.70.73	172.16.20.1	FTP	116	Response: 227 Entering Passive Mode (90,130,70,73,111,84).
20	5.116999	172.16.20.1	90.130.70.73	TCP	74	56904 → 28500 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=36633622 TSecr=0 WS=128
21	5.154125	172.16.20.1	90.130.70.73	TCP	66	59973 → 21 [ACK] Seq=31 Ack=128 Win=29312 Len=0 TSval=36633632 TSecr=1382356193
22	5.170460	90.130.70.73	172.16.20.1	TCP	74	28500 → 56904 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=1382356206 TS
23	5.170483	172.16.20.1	90.130.70.73	TCP	66	56904 → 28500 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=36633636 TSecr=1382356206
24	5.170509	172.16.20.1	90.130.70.73	FTP	76	Request: TYPE L 8
25	5.222116	90.130.70.73	172.16.20.1	FTP	97	Response: 200 Switching to Binary mode.
26	5.222149	172.16.20.1	90.130.70.73	TCP	66	59973 → 21 [ACK] Seq=41 Ack=159 Win=29312 Len=0 TSval=36633648 TSecr=1382356219
27	5.222178	172.16.20.1	90.130.70.73	FTP	83	Request: RETR /100MB.zip
28	5.273464	90.130.70.73	172.16.20.1	FTP	141	Response: 150 Opening BINARY mode data connection for /100MB.zip (104857600 bytes).
29	5.276188	90.130.70.73	172.16.20.1	FTP-DA…	1514	FTP Data: 1448 bytes
30	5.276219	172.16.20.1	90.130.70.73	TCP	66	56904 → 28500 [ACK] Seq=1 Ack=1449 Win=32128 Len=0 TSval=36633662 TSecr=1382356232
31	5.276441	90.130.70.73	172.16.20.1	FTP-DA…	2962	FTP Data: 2896 bytes

Figura 22: Experiência 6 - 2 downloads em simultâneo