



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Redes de Computadores 2021/2022

2º Trabalho laboratorial:

Realizado por:

- Angela Manuela Correia Antelo Costa Cruz up201806781
- Fabio Huang up201806829

Sumário

O presente trabalho foi realizado no âmbito da disciplina de Redes de Computadores (RC) do 3º ano do Licenciatura de Engenharia Informática e Computação (LEIC) da Universidade do Porto. Trata-se do segundo trabalho laboratorial da disciplina que está dividido em duas partes, uma sendo a implementação de uma aplicação de download usando o protocolo de FTP (File Transfer Protocol) e a segunda parte do trabalho trata-se de fazer uma configuração e a análise de uma rede.

Parte 1 - Aplicação de Download

A primeira parte do trabalho tem como objetivo a implementação de uma aplicação FTP (*File Transmission Protocol*) simples para receber um ficheiro usando o protocolo FTP. O programa deverá usar também ligações TCP (*Transmission Control Protocol*) e recorrer à utilização de sockets.

Execução do programa

Usage: ./download ftp://[<user>:<password>@]<host>/<url-path>

Example:./download ftp://[ftp.up.pt](ftp://ftp.up.pt)/pub/kodi/timestamp.txt

Estrutura da Aplicação utilizada

No início do programa, é feita a análise e verificação dos argumentos de entrada passados pela linha de comando mencionados em cima, estes valores são guardados num **struct <application_params_t>** nos parâmetros **<user>**, **<pass>**, **<host>**, **<url_path>** e **<filename>**. Depois é chamado a função **getip** onde retorna a **struct <hostent>** com informações extras sobre o **<host>** fornecido pelo utilizador (por exemplo endereço IP e nome do host), que são utilizadas para a conexão ao servidor FTP com um socket na porta 21. Depois de estabelecer a ligação com o servidor FTP, é passado ao servidor as credenciais (user e pass) para fazer login, no sucesso, é enviado o comando PASV para transferir os dados no modo passivo, onde o servidor responde com 6 bytes onde os últimos dois bytes são utilizados para calcular a porta para receber os dados de download (por exemplo 193,136,28,12,19,91 onde 19 e 91 são usados para calcular a porta, porta = 19x256 + 91 = 4955). De seguida é usada uma segunda socket mas na porta com o valor calculado, para fazer a conexão com o servidor FTP e fica aguardar para receber os dados do ficheiro a transferir. Estes dados só são transferidos pelo servidor, depois do primeiro socket enviar caminho completo do ficheiro a transferir (**<url_path>** + **<filename>**). Depois de receber e guardar os dados do ficheiro, é enviado o comando QUIT para desconectar com o servidor.

Download Com Sucesso

A figura abaixo representa um exemplo da execução da aplicação FTP com sucesso.

```

-zsh
220-Questions and comments can be sent to mirrors@porto.pt
220-
220-
220-
>>>Client:
user anonymous
>>>Server Response:
331 Please specify the password.
>>>Client:
pass anonymous
>>>Server Response:
230 Login successful.
>>>Client:
posv
227 Entering Passive Mode (193,137,29,15,216,127).
>>>Client:
retr pub/kodi/timestamp.txt
>>>Client 2 is Downloading the file:
>>>Server Response:
150 Opening BINARY mode data connection for pub/kodi/timestamp.txt (11 bytes).
226 Transfer complete.
>>>Client:
quit
>>>Server Response:
221 Goodbye.

fabio@fabioideMBP lab2 % cat timestamp.txt
1643404801
fabio@fabioideMBP lab2 %

```

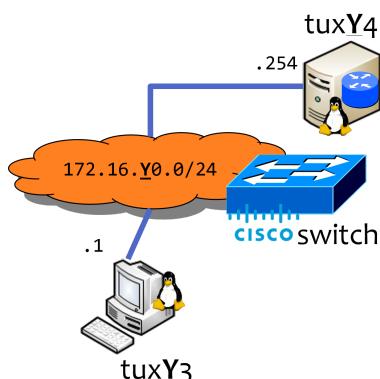
Parte 2- Configuração e Estudo de uma Rede

Experiência 1 - Configuração do IP

Objetivo

Nesta 1º experiência tínhamos como objetivo configurar os endereços IP de 2 computadores (tuxY3 e tuxY4, onde Y é o número da bancada, no nossa experiência foram usados os computadores da bancada 3 da sala I320) e criar uma ligação entre eles.

Arquitetura



Configurações:

- Tux33 Eth0 com IP 172.16.30.1 na porta 1
- Tux34 Eth0 com IP 172.16.30.254 na porta 2

Comandos principais de Configuração

tux33:

```

> ifconfig eth0 down
> ifconfig eth0 up
> ifconfig eth0 172.16.30.1/24
% Desativar eth0
% Ativar eth0
% Definir IP e máscara eth0

```

tux34:

```

> ifconfig eth0 down
% Desativar eth0

```

```
> ifconfig eth0 up                                % Ativar eth0
> ifconfig eth0 172.16.30.254/24                 % Definir IP e máscara eth0
```

Análise da experiência

ARP é tipicamente usado para associar endereços **IPs** com endereços **MACs**, quando um computador pretende fazer comunicação com outro computador pela primeira vez, é feita o broadcasting do pacote **ARP** para todos os computadores dentro da rede, este pacote contém as informações do **IP** do emissor e o **IP** do receptor, após o receptor receber este pacote responde com outro pacote **ARP** dizendo o endereço **MAC** associado ao seu **IP** onde este mapeamento é guardado na **ARP Table**.

Na nossa experiência quando é feito o **ping** de **tux33** para o **tux34**, podemos ver pelos resultados no wireshark, que foi feito um broadcasting de um pacote **ARP** com a informação '**Who has 172.16.30.254? Tell 172.16.30.1**' pelo **tux33**, e de seguida outro pacote **ARP** é respondida com a informação '**172.16.30.254 is at 00:21:5a:5a:74:3e**' pelo **tux34**, onde depois existe a comunicação entre os computadores criada pelo **ping**, que são pacotes do tipo **ICMP**, onde os pings de requests têm os endereços de **tux33** como o de origem (**IP: 172.16.30.1, MAC: 00:21:5a:5a:7d:b7**), e os endereços de **tux34** como o de destino (**IP: 172.16.30.254, MAC: 00:21:5a:5a:74:3e**).

Podemos determinar o tipo da trama de Ethernet a receber com o valor de **Type** da camada de **Ethernet II**, onde pode ter os valores **IPv4 (0x0800)** para tramas do tipo **ICMP**, **IPv4 (0x0806)** para as tramas de tipo **ARP**. Com o wireshark também podemos identificar o tipo de protocolo facilmente olhando na coluna de **Protocol**.

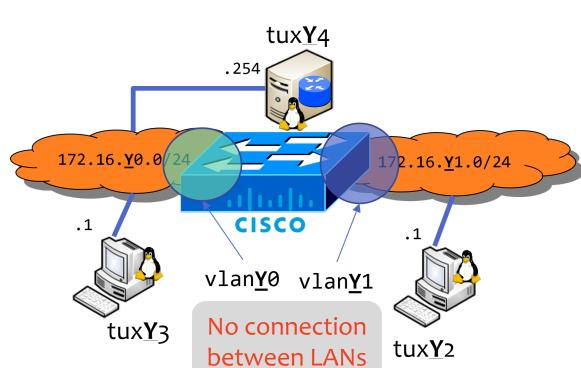
O tamanho da trama recebida pode ser encontrada diretamente na **Frame Length** dentro da primeira trama representada no Wireshark para qualquer tipo de trama ou para os pacotes de tipo **ICMP**, somando o valor de **Total Length** da camada de **Internet Protocol Version 4** com 14 bytes.

Experiência 2 - Virtual LANs

Objetivo

Nesta 2º experiência foi necessário usar as configurações dos **tux33** e **tux34** da experiência anterior. O objetivo desta experiência era configurar uma ligação através de VLAN e verificar o seu comportamento. Foi então necessário criar duas VLANs (referidas na imagem abaixo por **vlanY0** e **vlanY1**, no nosso caso **vlan30** e **vlan31**) no switch, no qual a **vlan30** está ligada ao **tux33** e **tux34** e a **vlan31** a um terceiro computador **tux32**.

Arquitetura



Configurações:

VLAN 30:

- Tux33 Eth0 com IP 172.16.30.1 na porta 1
- Tux34 Eth0 com IP 172.16.30.254 na porta 2

VLAN 31:

- Tux32 Eth0 com IP 172.16.31.1 na porta 4

Comandos principais de Configuração

... todos os comandos da experiência 1

```
tux32:  
> ifconfig eth0 down                                % Desativar eth0  
> ifconfig eth0 up                                  % Ativar eth0  
> ifconfig eth0 172.16.31.1/24                      % Definir IP e máscara eth0  
  
GTKterm (Switch):  
> configure terminal                                 % Criar vlan 30  
> vlan 30  
> end  
  
> configure terminal                                 % Criar vlan 31  
> vlan 31  
> end  
  
> configure terminal                                 % Adicionar interface da  
> interface fastethernet 0/1                         % porta 1 (tux33) para  
> switchport mode access                           % a vlan 30  
> switchport access vlan 30  
> end  
  
> configure terminal                                 % Adicionar interface da  
> interface fastethernet 0/2                         % porta 2 (tux34) para  
> switchport mode access                           % a vlan 30  
> switchport access vlan 30  
> end  
  
> configure terminal                                 % Adicionar interface da  
> interface fastethernet 0/4                         % porta 4 (tux32) para  
> switchport mode access                           % a vlan 30  
> switchport access vlan 31  
> end
```

Análise da experiência

Para configurar a vlan, usamos os seguintes comandos no switch:

```
> configure terminal                                 % Criar vlan Y0  
> vlan Y0  
> end  
  
> configure terminal                                 % Adicionar interface da  
> interface fastethernet 0/X                       % porta X para  
> switchport mode access                           % a vlan Y0  
> switchport access vlan Y0  
> end
```

Na nossa experiência foram usados estes comando para criar **vlan 30 e 31**, de seguida adicionou-se as portas dos computadores às **vlans** respectivas, ou seja configuramos o **tux33 e tux34** na **vlan 30**, e **tux32** na **vlan 31**.

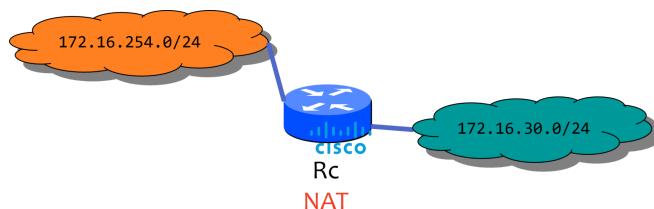
Durante a experiência conseguimos fazer **ping** de **tux33** para **tux34** mas não de **tux33** para **tux32**. No entanto, quando fazemos ping de broadcast de tux33, este consegue chegar ao **tux33** e **tux34**, mas não para **tux32**, o mesmo acontece no lado de **tux32**, quando fazemos ping de broadcast de **tux32**, só o **tux32** consegue-o receber, com isso em conta, podemos concluir que **tux33** e **tux34** estão num mesmo **vlan**, enquanto o **tux32** está num outro **vlan**, ou seja, temos dentro da nossa rede 2 **domínios broadcast** diferentes (172.16.31.255 para **vlan 31** e 172.16.30.255 para **vlan30**).

Experiência 3 - Configuração de Router

Objetivo

Objetivo desta experiência foi aprender a analisar o ficheiro de configuração do Router Cisco, testar as entradas para DNS e configurar as rotas na máquina local (nota: esta experiência foi feito em remote)

Arquitetura



Análise da experiência

(Cisco) How to configure a static route in a commercial router?

Podemos fazer isso com os comandos seguintes na consola do router:

```
> configure terminal  
> ip route <address> <netmask> <next_hop>  
> exit
```

(Cisco) How to configure NAT in a commercial router?

1. Definir as interfaces inside/outside
2. Definir a NAT pool com um intervalo definido de endereços IP (exemplo abaixo só é usado um único IP)
3. Associar uma access list com os endereços das interfaces inside permitidas

Exemplo:

```
> interface FastEthernet0/0  
> ip address 172.16.31.254 255.255.255.0  
> ip nat inside  
  
> interface FastEthernet0/1  
> ip address 172.16.2.39 255.255.255.0
```

```
> ip nat outside  
  
> ip nat pool ovrlid 172.16.2.39 172.16.2.39 prefix-length 24  
> ip nat inside source list 1 pool ovrlid overload  
  
> access-list 1 permit 172.16.30.0 0.0.0.7  
> access-list 1 permit 172.16.31.0 0.0.0.7
```

(Cisco) What does NAT do?

Como o nome indica (Network Address Translation), NAT converte entre IPs públicas e IPs privados, com o objetivo de poupar o espaço limitado de endereçamento público (IPv4), os endereços públicos são geridos por uma entidade reguladora e que permitem identificar um computador na Internet unicamente, enquanto, os endereços privados só são conhecidos numa rede local, que quando sair para a internet, o seu endereço privado tem de ser convertida pela NAT para um endereço público.

(DNS) How to configure the DNS service at a host?

Para configurar o DNS, podemos modificar o ficheiro **/etc/resolv.conf**, especificando o endereço IP ou domain name do DNS.

Exemplo:

```
> sudo nano /etc/resolv.conf  
depois adicionar no ficheiro nameserver <endereço IP>
```

(DNS) What packets are exchanged by DNS and what information is transported

São pacotes do tipo UDP enviados para o servidor DNS para receber o endereço IP de um domain name desejado. Estes pedidos contém um query com domain name a desejar, com um Transaction ID para identificar a resposta a esse pedido, onde se pode encontrar no campo das Answers, o/s endereço/s IPs associados a esse domain name.

(Linux) What ICMP packets are observed and why?

Os pacotes de ICMP observados são pacotes de Echo Reply e Echo Request, que permite identificar no caso de mensagens enviadas com falha.

(Linux) What are the IP and MAC addresses associated to ICMP packets and why?

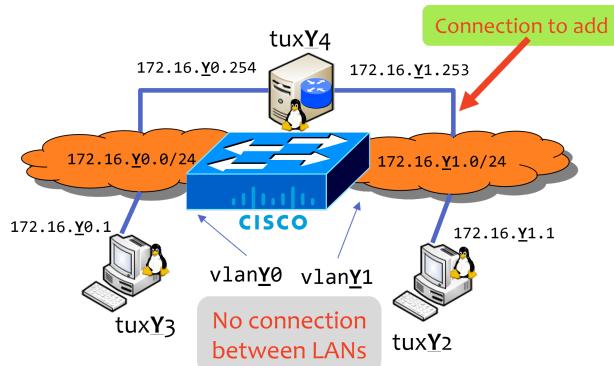
Num pacote ICMP de Echo Request, os endereços IP e MAC de origem correspondem à da máquina de origem, e os endereços IP e MAC de destino são correspondem à da máquina de destino e vice versa para ICMP de Reply.

Experiência 4 -Router Configuration (Lab)

Objetivo

Configurar um router em linux e configurar um router em cisco.

Arquitetura



Comandos principais de Configuração

... todos os comandos da experiência 1 e 2

tux34:

```
> ifconfig eth1 down                                % Desativar eth1  
> ifconfig eth1 up                                 % Ativar eth1  
> ifconfig eth1 172.16.31.253/24                  % Definir IP e máscara eth1
```

GTKterm (Switch):

```
> configure terminal  
> interface fastethernet 0/3                         % Adicionar interface da  
> switchport mode access                            % porta 3 (tux34) para  
> switchport access vlan 31                          % a vlan 31  
> end
```

tux34:

```
> echo 1 > /proc/sys/net/ipv4/ip_forward           % Enabling forwarding in tuxes  
  
> echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts  
% Enabling echo reply to broadcast request
```

tux32:

```
> ip route add 172.16.30.0/24 via 172.16.31.253  
% Add route to VLAN30
```

tux33:

```
> ip route add 172.16.31.0/24 via 172.16.30.254  
% Add route to VLAN31
```

Análise da experiência

(Linux) What routes are there in the tuxes? What are their meaning?

Foram geradas rotas quando se ligam os computadores através de VLANs. O tux 33 tem rota para a VLAN 30, o tux 32 para a VLAN 31 e o tux 34 tem rota para ambas vlans. A gateway é 0.0.0.0 visto que a ligação não tem nenhum intermediário pois está diretamente conectada com o destino.

(Linux) What information does an entry of the forwarding table contain?

Foram geradas forwarding tables para todos os computadores. A informação que contém é:

- **Destination**- Endereço IP da rede de destino
- **Gateway** - Endereço da gateway
- **Genmask** - Máscara subnet
- **Flags**- Informações sobre rota (**U**-Up and **G**-Gateway)
- **Metric**- Custo da rota
- **Ref**- Número de referência da rota
- **Use**- Contador de pesquisa da rota
- **Iface**-Placa de rede usada para enviar mensagem (eg.eth0)

Tabela para tux 32:

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
172.16.30.0	172.16.31.253	255.255.255.0	UG	0	0	0	eth0
172.16.31.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0

Tabela para tux 33:

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
172.16.30.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
172.16.31.0	172.16.30.254	255.255.255.0	UG	0	0	0	eth0

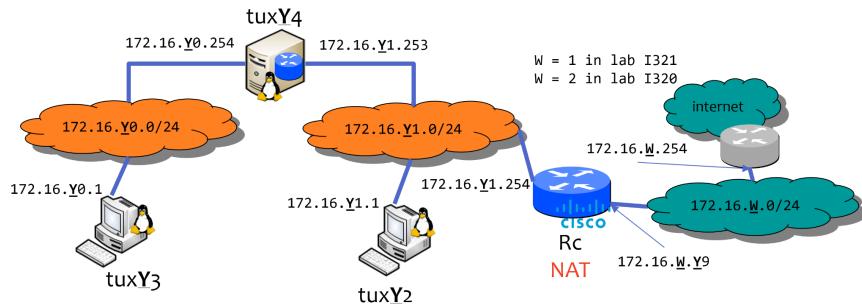
Tabela para tux 34:

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
172.16.30.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
172.16.31.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1

- (Linux) What are the IP and MAC addresses associated to ICMP packets and why?

Os pacotes ICMP são pedidos e respostas ping. Os endereços IP e MAC de origem e destino associados com os pacotes ICMP são os endereços MAC e IP referentes aos computadores que recebem/enviam os pacotes. No caso de o tux 33 enviar um pacote ICMP para o tux 34 conectados por eth0 o endereços de IP e MAC de origem será a do tux 33 (*IP: 172.16.30.1, MAC: 00:21:5a:5a:7d:b7*) e os de destino serão os endereços do tux 34 (*IP: 172.16.30.254, MAC: 00:21:5a:5a:74:3e*). Caso os computadores não estarem a usar a mesma placa de rede não será possível enviar o pacote ICMP sem os alterar.

Configuração do Cisco Router



GTKterm (Switch):

```
> configure terminal
> interface fastethernet 0/5
> switchport mode access
> switchport access vlan 31
> end
% Adicionar interface da
% porta 5 (Cisco Router)
% para a vlan 31
```

GTKterm (Cisco Router):

```
> configure terminal
> % Colar o ficheiro de configuração do Cisco Router
> ...
> exit

tux32:
> ip route add default via 172.16.31.254
% Configurar default gateway para Router Cisco

tux34:
> ip route add default via 172.16.31.254
% Configurar default gateway para Router Cisco
tux33:
> ip route add default via 172.16.30.254
% Configurar default gateway para tux34
```

Análise da experiência

(Cisco) What are the paths followed by the packets in the experiments carried out and why?

No caso da rota existir, os pacotes seguem essa mesma rota. Caso contrário, os pacotes são dirigidos ao router pois este foi definido em cima como rota default, daí caso não exista uma rota definida todos os pedidos serão redirecionados para o router.

Conclusão

Este trabalho teve como objetivo o desenvolvimento de uma aplicação de download de ficheiros através de conexões utilizando os protocolos FTP e a segunda parte para fazer uma configuração e a análise de uma rede , onde tivemos que saber funcionar com dispositivos como o switch , o router e multiplas tecnicas nomeadamente ICMP,ARP ,NAT,DNS.

Em suma consideramos que o trabalho foi concluído com sucesso visto que todos os requisitos foram cumpridos e reconhecemos que o desenvolvimento do mesmo ajudou a aprofundar os conhecimentos dado na unidade curricular.

Anexos

application.h

```
#ifndef APPLICATION_H

#define APPLICATION_H

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <signal.h>

#include <netdb.h>

#include <netinet/in.h>

#include<arpa/inet.h>

#include <unistd.h>

#include <libgen.h>

#include "macros.h"

#include "log.h"

typedef struct application_params_t{

    char *user;

    char *pass;

    char *host;

    char *url_path;

    char *filename;

}application_params_t;

application_params_t application_params;

int parseArgs(int argc, char **argv);

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

void segmentation_fault_handler();

int getServerResponse(int sockfd, int linesN);

int quit(int sockfd);
```

```
int passiveModeRequest(int sockfd) ;  
  
int download(int sockfd2, int sockfd) ;  
  
void sendRetr(int sockfd) ;  
  
#endif
```

application.c

```
#include "application.h"  
  
int parseArgs(int argc, char **argv)  
{  
  
    if (argc != 2)  
    {  
  
        return -1;  
    }  
  
    signal(SIGSEGV, segmentation_fault_handler);  
  
    char protocol[7];  
  
    strncpy(protocol, argv[1], 6);  
  
    protocol[6]='\0';  
  
    if(strcmp(protocol,"ftp://") !=0){  
  
        logError("protocol is not FTP.");  
  
        exit(-1);  
    }  
  
    char *ret = strstr(argv[1], "@");  
  
    if (ret == NULL)  
    {  
  
        // No user and pass provided  
  
        // ftp://<host>/<url-path>  
  
        // removedHeader will contain <host>/<url-path>  
  
        char *removedHeader = (char *)malloc(strlen(argv[1]) - 6);  
  
        memcpy(removedHeader, &argv[1][6], strlen(argv[1])-6);
```

```
application_params.host = strtok(removedHeader, "/");
application_params.url_path = strtok(NULL, ";");
application_params.user = "anonymous";
application_params.pass = "anonymous";
}

else
{
    // header will contain ftp://[<user>:<password>@]

    char *header = strtok(argv[1], "@");

    application_params.host = strtok(NULL, "/");
    application_params.url_path = strtok(NULL, ";");

    // user_pass will contain <user>:<passw ord>

    char *user_pass = (char *)malloc(strlen(header) - 6);
    memcpy(user_pass, &header[6], strlen(header)-6);

    application_params.user = strtok(user_pass, ":");

    application_params.pass = strtok(NULL, ":");

}

if (application_params.host == NULL || application_params.url_path == NULL)
{
    return -1;
}

application_params.filename = basename(application_params.url_path);

char msg[MAX_SIZE];

sprintf(msg, "\n\tuser = %s\n\tpass = %s\n\thost = %s\n\turl path = %s\n\tfile
name = %s\n",
        application_params.user,
```

```

        application_params.pass,
        application_params.host,
        application_params.url_path,
        application_params.filename);

    logInfo(msg);

    return 0;
}

struct hostent *getip(char *host)
{
    struct hostent *h;

    /**
     * The struct hostent (host entry) with its terms documented
     */

    struct hostent {

        char *h_name;      // Official name of the host.

        char **h_aliases;   // A NULL-terminated array of alternate names for
the host.

        int h_addrtype;    // The type of address being returned; usually
AF_INET.

        int h_length;      // The length of the address in bytes.

        char **h_addr_list; // A zero-terminated array of network addresses
for the host.

        // Host addresses are in Network Byte Order.
    };
}

#define h_addr h_addr_list[0]    The first address in h_addr_list.

*/
if ((h = gethostbyname(host)) == NULL)
{

```

```
    logError("gethostbyname()") ;

    exit(-1) ;

}

char msg[MAX_SIZE] ;

sprintf(msg, "\n\tHost name : %s\n\tIP Address : %s\n", h->h_name,
inet_ntoa(*((struct in_addr *)h->h_addr))) ;

logInfo(msg) ;

return h ;

}

void segmentation_fault_handler()

{

    logUsage() ;

    exit(-1) ;

}

// type = 0 for user, type = 1 for pass

void sendCredentials(int sockfd, int type)

{

    char buff[MAX_SIZE] ;

    if (type == 0)

    {

        sprintf(buff, "user %s", application_params.user) ;

    }

    else

    {

        sprintf(buff, "pass %s", application_params.pass) ;

    }

}
```

```
printf(">>>Client:\n%s\n", buff) ;

write(sockfd, buff, strlen(buff)) ;

write(sockfd, "\n", 1) ;

}

void sendRetr(int sockfd)

{

    char buff[MAX_SIZE] ;

    sprintf(buff, "retr %s", application_params.url_path) ;

    printf(">>>Client:\n%s\n", buff) ;

    write(sockfd, buff, strlen(buff)) ;

    write(sockfd, "\n", 1) ;

}

int download(int sockfd2, int sockfd)

{

    char buff[2] ;

    int res ;

    FILE * fileFp = fopen(application_params.filename, "wb") ;



    printf(">>>Client 2 is Downloading the file:\n") ;

    res = read(sockfd2, buff, 1) ;

    fwrite(buff, 1, 1, fileFp) ;

    while (res > 0)

    {

        res = read(sockfd2, buff, 1) ;

        fwrite(buff, 1, 1, fileFp) ;



    }

    fclose(fileFp) ;
```

```
        return getServerResponse(sockfd, 2);

}

int passiveModeRequest(int sockfd)
{
    char ip_port[MAX_SIZE];
    int res;
    char buff[2];
    printf(">>>Client:\npasv\n");
    write(sockfd, "pasv", 4);
    write(sockfd, "\n", 1);
    res = read(sockfd, buff, 1);
    printf("%c", buff[0]);
    while (res > 0)
    {
        res = read(sockfd, buff, 1);
        printf("%c", buff[0]);
        if (buff[0] == '\n')
        {
            break;
        }
        if (buff[0] == '(')
        {
            int index = 0;
            while (res > 0)
            {
                res = read(sockfd, buff, 1);
                printf("%c", buff[0]);
                if (buff[0] == ')')

```

```
        {

            break;

        }

        ip_port[index] = buff[0];

        index++;

    }

}

// Parse the and calculate port number

int a, b;

char *temp = strtok(ip_port, ",,");

for (int i = 0; i < 5; i++)

{

    temp = strtok(NULL, ",,");

    if (i == 3)

    {

        a = atoi(temp);

    }

    if (i == 4)

    {

        b = atoi(temp);

    }

}

return a * 256 + b;

}

int quit(int sockfd){
```

```
    write(sockfd, "quit", 4);

    write(sockfd, "\n", 1);

    printf(">>>Client:\nquit\n");

    int status = getServerResponse(sockfd, 2);

    return status;

}

int getServerResponse(int sockfd, int linesN)

{

    char buff[2];

    int status = 0;

    int res;

    printf(">>>Server Response:\n");

    for (int i = 0; i < 3; i++)

    {

        res = read(sockfd, buff, 1);

        printf("%c", buff[0]);

        status = status * 10 + atoi(&buff[0]);

    }

    // printf("Status code: %d\n", status);

    int count = 0;

    while (res > 0)

    {

        res = read(sockfd, buff, 1);

        printf("%c", buff[0]);

        if (buff[0] == '\n')

        {

            count++;

        }

    }

}
```

```
    if (count == linesN)

    {

        break;

    }

}

return status;

}

// int getServer


int main(int argc, char **argv)

{

    if (parseArgs(argc, argv) < 0)

    {

        logUsage();

        exit(-1);

    }

    struct hostent *h = getip(application_params.host);

    int sockfd;

    struct sockaddr_in server_addr;

    /*server address handling*/

    bzero((char *)&server_addr, sizeof(server_addr));

    server_addr.sin_family = AF_INET;

    server_addr.sin_addr.s_addr = inet_addr(inet_ntoa(*((struct in_addr

*)h->h_addr))); /*32 bit Internet address network byte ordered*/

    server_addr.sin_port = htons(FTP_PORT);

/*server TCP port must be network byte ordered */
```

```
/*open a TCP socket*/

if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) < 0)

{

    logError("socket()");

    exit(-1);

}

/*connect to the server*/

if (connect(sockfd,

            (struct sockaddr *)&server_addr,

            sizeof(server_addr)) < 0)

{

    logError("connect()");

    exit(-1);

}

// Server response after connection

if (getServerResponse(sockfd, 10) != 220)

{

    logError("getServerResponse() Failed! Unable to connect to server\n");

    exit(-1);

}

// Send User

sendCredentials(sockfd, USER_TYPE);

// Server response after sending user

if (getServerResponse(sockfd, 1) != 331)

{

    logError("getServerResponse() Failed! Wrong user provided\n");

}
```

```
    exit(-1);

}

// Send Pass

sendCredentials(sockfd, PASS_TYPE);

if (getServerResponse(sockfd, 1) != 230)

{

    logError("getServerResponse() Failed! Wrong password provided\n");

    exit(-1);

}

int portNum = passiveModeRequest(sockfd);

// printf("portNum = %d\n", portNum);

int sockfd2;

struct sockaddr_in server_addr2;

/*For the second client to receive the file data*/

bzero((char *)&server_addr2, sizeof(server_addr2));

server_addr2.sin_family = AF_INET;

server_addr2.sin_addr.s_addr = inet_addr(inet_ntoa(*((struct in_addr *)
*)h->h_addr))); /*32 bit Internet address network byte ordered*/

server_addr2.sin_port = htons(portNum);

/*server TCP port must be network byte ordered */

/*open a TCP socket*/

if ((sockfd2 = socket(AF_INET, SOCK_STREAM, 0)) < 0)

{

    logError("socket() ");

    exit(-1);
```

```
}

/*connect to the server*/

if (connect(sockfd2,
            (struct sockaddr *)&server_addr2,
            sizeof(server_addr2)) < 0)

{

    logError("connect()");

    exit(-1);

}

sendRetr(sockfd);

if(download(sockfd2, sockfd) != 150){

    logError("download() Failed! Unable to download\n");

    exit(-1);

}

if(quit(sockfd) != 221){

    logError("quit() Failed! Unable to disconnect\n");

    exit(-1);

}

if (close(sockfd)<0) {

    perror("close()");

    exit(-1);

}

if (close(sockfd2)<0) {

    perror("close()");

    exit(-1);

}
```

```
    return 0;  
}  
}
```

log.h

```
#ifndef LOG_H  
  
#define LOG_H  
  
#include <stdio.h>  
  
#include <stdlib.h>  
  
#include <string.h>  
  
#include <unistd.h>  
  
#include "macros.h"  
  
  
void logError(char *msg);  
  
/**  
* @brief Log Sucess.  
*  
* Prints Sucess message in the right format  
*  
* @param msg      array to put the message  
* @return void  
*/  
  
void logSuccess(char *msg);  
  
/**  
* @brief Log Info.  
*  
* Prints message with information in the right format  
*  
* @param msg      array to put the message
```

```

* @return void
*/
void logInfo(char *msg);

/***
* @brief Log Warning.

*
* Prints Warning message in the right format
*
* @param msg      array to put the message
* @return void
*/
void logWarning(char *msg);

/***
* @brief Log Usage.

*
* Prints the correct usage of the application
*
* @return void
*/
void logUsage();

void logServer(char *msg);

#endif

```

Log.c

```

#include "log.h"

void logError(char *msg) {
    char buf[MAX_SIZE];
    sprintf(buf, "\033[0;31m>>>ERROR:\t%s\n\033[0m", msg);
    write(STDOUT_FILENO, buf, strlen(buf));
}

void logSuccess(char *msg) {

```

```

    char buf[MAX_SIZE];
    sprintf(buf, "\033[0;32m>>>SUCCESS:\t%s\n\033[0m", msg);
    write(STDOUT_FILENO, buf, strlen(buf));
}

void logWarning(char *msg) {
    char buf[MAX_SIZE];
    sprintf(buf, "\033[0;33m>>>WARNING:\t%s\n\033[0m", msg);
    write(STDOUT_FILENO, buf, strlen(buf));
}

void logInfo(char *msg) {
    char buf[MAX_SIZE];
    sprintf(buf, ">>>Info:\t%s\n", msg);
    write(STDOUT_FILENO, buf, strlen(buf));
}

void logUsage() {
    char buf[MAX_SIZE];
    sprintf(buf, "Usage: ./download ftp://[<user>:<password>@]<host>/<url-path>\n");
    write(STDOUT_FILENO, buf, strlen(buf));
}

void logServer(char *msg) {
    char buf[MAX_SIZE];
    sprintf(buf, ">>>Server:\n%s\n", msg);
    write(STDOUT_FILENO, buf, strlen(buf));
}

```

macros.h

```

#define MAX_SIZE 255
#define FTP_PORT 21

#define USER_TYPE 0
#define PASS_TYPE 1

```

Logs das Experiências

Log Experiência 1

8	6.383830232	Cisco_b6:8c:01	CDP/VTP/DTP/PAgP/U...	DTP	90	Dynamic Trunk Protocol
9	8.019575064	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/1/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001
10	10.024512778	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/1/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001
11	12.029353205	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/1/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001
12	13.286305642	Hewlett_P_5a:7d:b7	Broadcast	ARP	42	Who has 172.16.30.254? Tell 172.16.30.1
13	13.286440854	Hewlett_P_5a:7d:3e	Hewlett_P_5a:7d:b7	ARP	60	172.16.30.254 is at 00:21:5a:5a:74:3e
14	13.286449305	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x1961, seq=1/256, ttl=64 (reply in 15)
15	13.286585355	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x1961, seq=1/256, ttl=64 (request in 14)
16	14.034250203	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/1/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001
17	14.302960124	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x1961, seq=2/512, ttl=64 (reply in 18)
18	14.303120688	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x1961, seq=2/512, ttl=64 (request in 17)
19	14.449473153	Cisco_b6:8c:01	Cisco_b6:8c:01	LOOP	60	Reply
20	15.326967051	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x1961, seq=3/768, ttl=64 (reply in 21)
21	15.327105546	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x1961, seq=3/768, ttl=64 (request in 20)
22	16.043240725	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/1/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001
23	16.350965528	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x1961, seq=4/1024, ttl=64 (reply in 24)
24	16.351101098	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x1961, seq=4/1024, ttl=64 (request in 23)
25	17.374961212	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x1961, seq=5/1280, ttl=64 (reply in 26)
26	17.375097751	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x1961, seq=5/1280, ttl=64 (request in 25)
27	18.044110550	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/1/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001
28	18.371036542	Hewlett_P_5a:7d:3e	Hewlett_P_5a:7d:b7	ARP	60	Who has 172.16.30.1? Tell 172.16.30.254
29	18.371057145	Hewlett_P_5a:7d:b7	Hewlett_P_5a:7d:3e	ARP	42	172.16.30.1 is at 00:21:5a:5a:7d:b7
30	18.398955289	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x1961, seq=6/1536, ttl=64 (reply in 31)
31	18.399081073	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x1961, seq=6/1536, ttl=64 (request in 30)
32	20.048901810	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/1/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001

Log Experiência 2 - step 6 (tux33)

1	0.000000000	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001
2	0.034019504	Cisco_b6:8c:01	CDP/VTP/DTP/PAgP/U...	CDP	601	Device ID: gnu-sw3 Port ID: FastEthernet0/1
3	1.434743759	Cisco_b6:8c:01	Cisco_b6:8c:01	LOOP	60	Reply
4	2.0040905242	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001
5	4.014276888	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001
6	6.014795972	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001
7	8.019567050	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001
8	8.024439956	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001
9	11.442339172	Cisco_b6:8c:01	Cisco_b6:8c:01	LOOP	60	Reply
10	12.029337655	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001
11	12.784289884	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x1a1f, seq=1/256, ttl=64 (reply in 12)
12	12.784449820	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x1a1f, seq=1/256, ttl=64 (request in 11)
13	13.801443836	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x1a1f, seq=2/512, ttl=64 (reply in 14)
14	13.801578350	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x1a1f, seq=2/512, ttl=64 (request in 13)
15	14.038844584	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001
16	14.825443677	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x1a1f, seq=3/768, ttl=64 (reply in 17)
17	14.825576794	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x1a1f, seq=3/768, ttl=64 (request in 16)
18	15.849445892	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x1a1f, seq=4/1024, ttl=64 (reply in 19)
19	15.849607644	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x1a1f, seq=4/1024, ttl=64 (request in 18)
20	16.039220426	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001
21	16.873445175	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x1a1f, seq=5/1280, ttl=64 (reply in 22)
22	16.873580178	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x1a1f, seq=5/1280, ttl=64 (request in 21)
23	17.833409647	Hewlett_P_5a:7d:b7	Hewlett_P_5a:7d:3e	ARP	42	Who has 172.16.30.254? Tell 172.16.30.1
24	17.833526421	Hewlett_P_5a:7d:3e	Hewlett_P_5a:7d:b7	ARP	60	Who has 172.16.30.254 is at 00:21:5a:5a:7d:b7
25	17.937849458	Hewlett_P_5a:7d:b7	Hewlett_P_5a:7d:3e	ARP	60	Who has 172.16.30.17 is at 00:21:5a:5a:7d:b7
27	18.044169529	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001

Log Experiência 2 - step 9 (tux33)

19	26.067941740	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001	
20	27.137417887	Cisco_b6:8c:01	Cisco_b6:8c:01	LOOP	60	Reply	
21	27.819200450	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x1c4f, seq=1/256, ttl=64 (no response found!)	
22	28.068360547	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001	
23	28.824590228	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x1c4f, seq=2/512, ttl=64 (no response found!)	
24	29.848490708	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x1c4f, seq=3/768, ttl=64 (no response found!)	
25	30.073234445	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001	
26	30.872484759	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x1c4f, seq=4/1024, ttl=64 (no response found!)	
27	31.8992481312	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x1c4f, seq=5/1280, ttl=64 (no response found!)	
28	32.078131608	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001	
29	32.922087738	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x1c4f, seq=6/1536, ttl=64 (no response found!)	
30	33.944483650	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x1c4f, seq=7/1792, ttl=64 (no response found!)	
31	34.083082602	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001	
32	34.968484431	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x1c4f, seq=8/2048, ttl=64 (no response found!)	
33	35.9922483790	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x1c4f, seq=9/2304, ttl=64 (no response found!)	
34	36.8923560746	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001	
35	37.016517023	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x1c4f, seq=10/2560, ttl=64 (no response found!)	
36	37.145973366	Cisco_b6:8c:01	Cisco_b6:8c:01	LOOP	60	Reply	
37	38.040487180	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x1c4f, seq=11/2816, ttl=64 (no response found!)	
38	38.092878030	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001	
39	39.064512948	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x1c4f, seq=12/3072, ttl=64 (no response found!)	
40	40.088514542	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x1c4f, seq=13/3328, ttl=64 (no response found!)	
41	41.097687744	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001	
42	41.112511178	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x1c4f, seq=14/3584, ttl=64 (no response found!)	
43	42.102695598	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001	
44	42.136509350	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x1c4f, seq=15/3840, ttl=64 (no response found!)	
45	43.160483567	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x1c4f, seq=16/4096, ttl=64 (no response found!)	
46	44.167495744	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001	
47	44.184506323	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x1c4f, seq=17/4352, ttl=64 (no response found!)	
48	46.116822422	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001	
49	47.152707547	Cisco_b6:8c:01	Cisco_b6:8c:01	LOOP	60	Reply	
50	50.040469906	172.172419393	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8001
51	50.040469906	fe80::221:5aff:fe5.. ff02::2	ICMPv6	70	Router Solicitation from 00:21:5a:5a:7d:b7		
52	50.127124482	cisco_h6:8c:01	Spanning-tree-(for...	STP	60	Conf. Root = 32768/30/00:1e:14:h6:8c:00 Cost = 0 Port = 0x8001	

Log Experiência 2 - step 9 (tux32)

1	0.000000000	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
2	2.004888362	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
3	4.013951238	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
4	6.014602649	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
5	8.019479766	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
6	8.250533577	Cisco_b6:8c:04	Cisco_b6:8c:04	LOOP	60 Reply
7	10.024353113	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
8	12.029254116	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
9	14.038232694	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
10	16.039079798	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
11	18.043919900	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
12	18.245244582	Cisco_b6:8c:04	Cisco_b6:8c:04	LOOP	60 Reply
13	20.048754206	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
14	22.053588441	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
15	24.062536918	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
16	26.063361743	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
17	26.790285305	Cisco_b6:8c:04	CDP/VTP/DTP/Pag/...	CDP	601 Device ID: gnu-sw3 Port ID: FastEthernet0/4
18	28.068304441	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
19	28.257300088	Cisco_b6:8c:04	Cisco_b6:8c:04	LOOP	60 Reply
20	30.07051306	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
21	32.077933841	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
22	34.066897993	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
23	36.087675466	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
24	38.092533168	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
25	38.260443580	Cisco_b6:8c:04	Cisco_b6:8c:04	LOOP	60 Reply
26	40.097442063	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
27	42.102287473	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
28	44.111214648	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
29	46.112018102	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
30	48.116894801	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
31	48.268010009	Cisco_b6:8c:04	Cisco_b6:8c:04	LOOP	60 Reply
32	50.121752363	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
33	52.126704769	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
34	54.135805359	Cisco_b6:8c:04	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004

Log Experiência 2 - step 9 (tux34)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
2	2.007948515	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
3	4.013884127	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
4	5.082852533	Cisco_b6:8c:02	Cisco_b6:8c:02	LOOP	60 Reply	
5	5.764577002	172.16.30.1	172.16.30.255	ICMP	98 Echo (ping) request id=0x1c4f, seq=1/256, ttl=64 (no response found!)	
6	6.013748279	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
7	6.769874965	172.16.30.1	172.16.30.255	ICMP	98 Echo (ping) request id=0x1c4f, seq=2/512, ttl=64 (no response found!)	
8	7.793845751	172.16.30.1	172.16.30.255	ICMP	98 Echo (ping) request id=0x1c4f, seq=3/768, ttl=64 (no response found!)	
9	8.019757854	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
10	8.817829667	172.16.30.1	172.16.30.255	ICMP	98 Echo (ping) request id=0x1c4f, seq=4/1024, ttl=64 (no response found!)	
11	9.841852974	172.16.30.1	172.16.30.255	ICMP	98 Echo (ping) request id=0x1c4f, seq=5/1280, ttl=64 (no response found!)	
12	10.023925010	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
13	10.865815588	172.16.30.1	172.16.30.255	ICMP	98 Echo (ping) request id=0x1c4f, seq=6/1536, ttl=64 (no response found!)	
14	11.889800552	172.16.30.1	172.16.30.255	ICMP	98 Echo (ping) request id=0x1c4f, seq=7/1792, ttl=64 (no response found!)	
15	12.029871169	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
16	12.913793199	172.16.30.1	172.16.30.255	ICMP	98 Echo (ping) request id=0x1c4f, seq=8/2048, ttl=64 (no response found!)	
17	13.937782213	172.16.30.1	172.16.30.255	ICMP	98 Echo (ping) request id=0x1c4f, seq=9/2304, ttl=64 (no response found!)	
18	14.038011817	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
19	14.96181876	172.16.30.1	172.16.30.255	ICMP	98 Echo (ping) request id=0x1c4f, seq=11/2816, ttl=64 (no response found!)	
20	15.098415007	Cisco_b6:8c:02	Cisco_b6:8c:02	LOOP	60 Reply	
21	15.985769113	172.16.30.1	172.16.30.255	ICMP	98 Echo (ping) request id=0x1c4f, seq=13/3328, ttl=64 (no response found!)	
22	16.039867150	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
23	17.009786632	172.16.30.1	172.16.30.255	ICMP	98 Echo (ping) request id=0x1c4f, seq=12/3072, ttl=64 (no response found!)	
24	18.033780177	172.16.30.1	172.16.30.255	ICMP	98 Echo (ping) request id=0x1c4f, seq=13/3328, ttl=64 (no response found!)	
25	18.043928147	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
26	19.05776935	172.16.30.1	172.16.30.255	ICMP	98 Echo (ping) request id=0x1c4f, seq=14/3584, ttl=64 (no response found!)	
27	20.049839175	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
28	20.081748917	172.16.30.1	172.16.30.255	ICMP	98 Echo (ping) request id=0x1c4f, seq=15/3840, ttl=64 (no response found!)	
29	21.105716910	172.16.30.1	172.16.30.255	ICMP	98 Echo (ping) request id=0x1c4f, seq=16/4096, ttl=64 (no response found!)	
30	22.053877194	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
31	22.129731626	172.16.30.1	172.16.30.255	ICMP	98 Echo (ping) request id=0x1c4f, seq=17/4352, ttl=64 (no response found!)	
32	24.062057443	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
33	25.097962674	Cisco_b6:8c:02	Cisco_b6:8c:02	LOOP	60 Reply	
34	26.063830851	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
35	27.985639935	fe80::221:5aff:fe... ff02::2		ICMPv6	70 Router Solicitation from 00:21:5a:5a:7d:b7	
36	28.067762572	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
37	30.073630159	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
38	32.077326653	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
39	34.087223049	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002	
40	35.097217990	Cisco_b6:8c:02	Cisco_b6:8c:02	LOOP	60 Reply	

Log Experiência 2 - step 10 (tux33)

1	0.000000000	Cisco_b6:8c:01	Cisco_b6:8c:01	LOOP	60	Reply
2	1.216233421	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
3	3.221174793	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
4	5.230497574	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
5	7.230831882	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
6	9.235729464	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
7	10.011767831	Cisco_b6:8c:01	Cisco_b6:8c:01	LOOP	60	Reply
8	11.240628932	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
9	13.245549912	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
10	15.254863055	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
11	17.255276144	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
12	18.442479436	Cisco_b6:8c:01	CDP/VTP/DTP/PAgP/U..	CDP	601	Device ID: gnu-sw3 Port ID: FastEthernet0/1
13	19.260198101	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
14	20.006987248	Cisco_b6:8c:01	Cisco_b6:8c:01	LOOP	60	Reply
15	21.265060415	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
16	23.270014848	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
17	25.279388265	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
18	27.279734796	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
19	29.284628258	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
20	30.018656328	Cisco_b6:8c:01	Cisco_b6:8c:01	LOOP	60	Reply
21	31.289500140	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
22	33.294405686	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
23	35.303770233	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
24	37.304211189	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
25	39.309058418	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
26	40.022287265	Cisco_b6:8c:01	Cisco_b6:8c:01	LOOP	60	Reply
27	41.313998116	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
28	43.319259851	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
29	45.328184260	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
30	47.328604125	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
31	49.333551716	Cisco_b6:8c:01	Spanning-tree-(for..	STP	60	Conf. Root = 32768/30/00:1e:14:b6:8c:
32	50.029923728	Cisco_b6:8c:01	Cisco_b6:8c:01	LOOP	60	Reply

Log Experiência 2 - step 10 (tux32)

9	14.034212040	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
10	15.787317125	Cisco_b6:8c:04	Cisco_b6:8c:04	LOOP	60	Reply
11	16.038892480	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
12	18.047949414	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
13	20.048669776	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
14	22.053517763	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
15	24.058365059	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
16	24.214879157	Cisco_b6:8c:04	CDP/VTP/DTP/PAgP/U..	CDP	601	Device ID: gnu-sw3 Port ID: FastEthernet0/4
17	25.794970785	Cisco_b6:8c:04	Cisco_b6:8c:04	LOOP	60	Reply
18	26.063278424	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
19	28.072263777	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
20	30.073091329	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
21	32.077847339	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
22	34.082705460	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
23	35.106168562	172.16.31.1	172.16.31.255	ICMP	98	Echo (ping) request id=0x1898, seq=1/256, ttl=64 (no response found!)
24	35.794155152	Cisco_b6:8c:04	Cisco_b6:8c:04	LOOP	60	Reply
25	36.087577759	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
26	36.114822657	172.16.31.1	172.16.31.255	ICMP	98	Echo (ping) request id=0x1898, seq=2/512, ttl=64 (no response found!)
27	37.130820877	172.16.31.1	172.16.31.255	ICMP	98	Echo (ping) request id=0x1898, seq=3/768, ttl=64 (no response found!)
28	38.096549849	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
29	38.162820145	172.16.31.1	172.16.31.255	ICMP	98	Echo (ping) request id=0x1898, seq=4/1024, ttl=64 (no response found!)
30	39.186821298	172.16.31.1	172.16.31.255	ICMP	98	Echo (ping) request id=0x1898, seq=5/1280, ttl=64 (no response found!)
31	40.097322147	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
32	40.210821055	172.16.31.1	172.16.31.255	ICMP	98	Echo (ping) request id=0x1898, seq=6/1536, ttl=64 (no response found!)
33	41.234818577	172.16.31.1	172.16.31.255	ICMP	98	Echo (ping) request id=0x1898, seq=7/1792, ttl=64 (no response found!)
34	42.102184389	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
35	42.258823013	172.16.31.1	172.16.31.255	ICMP	98	Echo (ping) request id=0x1898, seq=8/2048, ttl=64 (no response found!)
36	44.107053125	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
37	45.801679049	Cisco_b6:8c:04	Cisco_b6:8c:04	LOOP	60	Reply
38	46.111938205	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
39	48.120886961	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
40	50.121706408	Cisco_b6:8c:04	Spanning-tree-(for..	STP	60	Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8004
41	50.746819465	fe80::221:5aff:fe6..	ff02::fb	MDNS	180	Standard query 0x0000 PTR _ftp._tcp.local, "QM" question PTR _nfs._tcp.local, "QM" question PTR _a
42	50.746903134	172.16.31.1	224.0.0.251	MDNS	160	Standard query 0x0000 PTR _ftp._tcp.local, "QM" question PTR _nfs._tcp.local, "QM" question PTR _a

Log Experiência 2 - step 10 (tux34)

1	0.00000000	Cisco_b6:8c:02	Cisco_b6:8c:02	LOOP	60 Reply	
2	1.230003764	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
3	3.233733643	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
4	5.243726487	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
5	7.243474213	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
6	8.430873546	Cisco_b6:8c:02	CDP/VT/DP/Pag/U...	CDP	601 Device ID: gnu-sw3 Port ID: FastEthernet0/2	
7	9.249441255	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
8	9.995191729	Cisco_b6:8c:02	Cisco_b6:8c:02	LOOP	60 Reply	
9	11.25357330	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
10	13.259164016	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
11	15.269007609	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
12	17.268692897	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
13	19.272816821	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
14	20.006702971	Cisco_b6:8c:02	Cisco_b6:8c:02	LOOP	60 Reply	
15	21.278828282	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
16	23.282800440	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
17	25.292835120	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
18	27.292505182	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
19	29.298364039	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
20	30.010239361	Cisco_b6:8c:02	Cisco_b6:8c:02	LOOP	60 Reply	
21	31.302170045	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
22	33.308133315	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
23	35.316309722	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
24	37.318080057	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
25	39.322056546	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
26	40.017788844	Cisco_b6:8c:02	Cisco_b6:8c:02	LOOP	60 Reply	
27	41.327953816	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
28	43.331781333	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
29	45.341684152	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
30	47.341409948	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
31	49.347305192	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
32	50.016946592	Cisco_b6:8c:02	Cisco_b6:8c:02	LOOP	60 Reply	
33	51.351232931	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
34	53.356953362	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
35	55.364765197	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
36	57.366232420	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
37	59.370091226	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8002	
38	60.024560399	Cisco_b6:8c:02	Cisco_b6:8c:02	LOOP	60 Rplv	

Log Experiência 4 - step 10 (tux33)

14	19.856073484	172.16.30.1	172.16.30.254	ICMP	98 Echo (ping) request id=0x1f5c, seq=1/256, ttl=64 (reply in 15)	
15	19.856239426	172.16.30.254	172.16.30.1	ICMP	98 Echo (ping) reply id=0x1f5c, seq=1/256, ttl=64 (request in 14)	
16	20.048654538	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8001	
17	20.861543747	172.16.30.1	172.16.30.254	ICMP	98 Echo (ping) request id=0x1f5c, seq=2/512, ttl=64 (reply in 18)	
18	20.861677981	172.16.30.254	172.16.30.1	ICMP	98 Echo (ping) reply id=0x1f5c, seq=2/512, ttl=64 (request in 17)	
19	21.885546681	172.16.30.1	172.16.30.254	ICMP	98 Echo (ping) request id=0x1f5c, seq=3/768, ttl=64 (reply in 20)	
20	21.885678681	172.16.30.254	172.16.30.1	ICMP	98 Echo (ping) reply id=0x1f5c, seq=3/768, ttl=64 (request in 19)	
21	22.053681738	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8001	
22	24.063001463	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8001	
23	24.942655245	HewlettP_5a:7d:b7	HewlettP_5a:7d:b7	ARP	60 Who has 172.16.30.1? Tell 172.16.30.254	
24	24.942675219	HewlettP_5a:7d:b7	HewlettP_5a:7d:b7	ARP	42 172.16.30.1 is at 00:21:5a:5a:7d:b7	
25	25.085506586	HewlettP_5a:7d:b7	HewlettP_5a:7d:b7	ARP	42 Who has 172.16.30.254? Tell 172.16.30.1	
26	25.085616725	HewlettP_5a:7d:b7	HewlettP_5a:7d:b7	ARP	60 172.16.30.254 is at 00:21:5a:5a:7d:3e	
27	25.627406900	Cisco_b6:8c:01	Cisco_b6:8c:01	LOOP	60 Reply	
28	26.063455149	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8001	
29	28.068435088	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8001	
30	30.073241755	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8001	
31	31.352005621	172.16.30.1	172.16.31.253	ICMP	98 Echo (ping) request id=0x1f63, seq=1/256, ttl=64 (reply in 32)	
32	31.352164161	172.16.31.253	172.16.30.1	ICMP	98 Echo (ping) reply id=0x1f63, seq=1/256, ttl=64 (request in 31)	
33	32.078237970	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8001	
34	32.381549535	172.16.30.1	172.16.31.253	ICMP	98 Echo (ping) request id=0x1f63, seq=2/512, ttl=64 (reply in 35)	
35	32.381678391	172.16.31.253	172.16.30.1	ICMP	98 Echo (ping) reply id=0x1f63, seq=2/512, ttl=64 (request in 34)	
36	33.405547021	172.16.30.1	172.16.31.253	ICMP	98 Echo (ping) request id=0x1f63, seq=3/768, ttl=64 (reply in 37)	
37	33.405680767	172.16.31.253	172.16.30.1	ICMP	98 Echo (ping) reply id=0x1f63, seq=3/768, ttl=64 (request in 36)	
38	34.091264927	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8001	
39	35.4639311985	Cisco_b6:8c:01	Cisco_b6:8c:01	LOOP	60 Reply	
40	36.100702544	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8001	
41	38.101185840	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8001	
42	40.106160125	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8001	
43	42.110966024	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8001	
44	44.115846722	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8001	
45	45.487944488	172.16.30.1	172.16.31.1	ICMP	98 Echo (ping) request id=0x1f6d, seq=1/256, ttl=64 (reply in 46)	
46	45.488369610	172.16.31.1	172.16.30.1	ICMP	98 Echo (ping) reply id=0x1f6d, seq=1/256, ttl=64 (request in 45)	
47	45.634414642	Cisco_b6:8c:01	Cisco_b6:8c:01	LOOP	60 Reply	
48	46.125037521	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8001	
49	46.497545360	172.16.30.1	172.16.31.1	ICMP	98 Echo (ping) request id=0x1f6d, seq=2/512, ttl=64 (reply in 50)	
50	46.497783308	172.16.31.1	172.16.30.1	ICMP	98 Echo (ping) reply id=0x1f6d, seq=2/512, ttl=64 (request in 49)	
51	47.517549496	172.16.30.1	172.16.31.1	ICMP	98 Echo (ping) request id=0x1f6d, seq=3/768, ttl=64 (reply in 52)	
52	47.517792662	172.16.31.1	172.16.30.1	ICMP	98 Echo (ping) reply id=0x1f6d, seq=3/768, ttl=64 (request in 51)	
53	48.125612099	Cisco_b6:8c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:le:14:b6:8c:00 Cost = 0 Port = 0x8001	
54	48.541544378	172.16.30.1	172.16.31.1	ICMP	98 Echo (ping) request id=0x1f6d, seq=4/1024, ttl=64 (reply in 55)	

Log Experiência 4 - step 14 (tux34)

186	146.3559228...	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002
187	147.3741385...	Cisco_b6:8c:03	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8003
188	147.7306209...	HewlettP_5a:7d:b7	Broadcast	ARP	60 Who has 172.16.30.254? Tell 172.16.30.1
189	147.7306475...	HewlettP_5a:7d:b7	HewlettP_5a:7d:b7	ARP	42 172.16.30.254 is at 00:21:5a:5a:74:3e
190	147.7307660...	172.16.30.1	172.16.31.1	ICMP	98 Echo (ping) request id=0x2064, seq=1/256, ttl=64 (reply in 191)
191	147.7310436...	172.16.31.1	172.16.30.1	ICMP	98 Echo (ping) reply id=0x2064, seq=1/256, ttl=63 (request in 190)
192	147.7307796...	EncoreNe_b4:b8:94	Broadcast	ARP	42 Who has 172.16.31.1? Tell 172.16.31.253
193	147.7308984...	HewlettP_61:24:01	EncoreNe_b4:b8:94	ARP	60 172.16.31.1 is at 00:21:5a:61:24:01
194	147.730912...	172.16.30.1	172.16.31.1	ICMP	98 Echo (ping) request id=0x2064, seq=1/256, ttl=63 (reply in 195)
195	147.7310308...	172.16.31.1	172.16.30.1	ICMP	98 Echo (ping) reply id=0x2064, seq=1/256, ttl=64 (request in 194)
196	148.3558772...	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002
197	148.5591519...	fe80::221:5aff:fe5...	ff02::2	ICMPv6	70 Router Solicitation from 00:21:5a:5a:74:3e
198	148.7320696...	172.16.30.1	172.16.31.1	ICMP	98 Echo (ping) request id=0x2064, seq=2/512, ttl=64 (reply in 199)
199	148.7322146...	172.16.31.1	172.16.30.1	ICMP	98 Echo (ping) reply id=0x2064, seq=2/512, ttl=63 (request in 198)
200	148.7320831...	172.16.30.1	172.16.31.1	ICMP	98 Echo (ping) request id=0x2064, seq=2/512, ttl=63 (reply in 201)
201	148.7321938...	172.16.31.1	172.16.30.1	ICMP	98 Echo (ping) reply id=0x2064, seq=2/512, ttl=64 (request in 200)
202	149.3790516...	Cisco_b6:8c:03	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8003
203	149.4799990...	Cisco_b6:8c:02	Cisco_b6:8c:02	LOOP	60 Reply
204	149.4880067...	Cisco_b6:8c:03	Cisco_b6:8c:03	LOOP	60 Reply
205	149.756052...	172.16.30.1	172.16.31.1	ICMP	98 Echo (ping) request id=0x2064, seq=3/768, ttl=64 (reply in 206)
206	149.7561949...	172.16.31.1	172.16.30.1	ICMP	98 Echo (ping) reply id=0x2064, seq=3/768, ttl=63 (request in 205)
207	149.7560680...	172.16.30.1	172.16.31.1	ICMP	98 Echo (ping) request id=0x2064, seq=3/768, ttl=63 (reply in 208)
208	149.7561784...	172.16.31.1	172.16.30.1	ICMP	98 Echo (ping) reply id=0x2064, seq=3/768, ttl=64 (request in 207)
209	150.3618279...	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002
210	150.7800454...	172.16.30.1	172.16.31.1	ICMP	98 Echo (ping) request id=0x2064, seq=4/1024, ttl=64 (reply in 211)
211	150.7801963...	172.16.31.1	172.16.30.1	ICMP	98 Echo (ping) reply id=0x2064, seq=4/1024, ttl=63 (request in 210)
212	150.7800653...	172.16.30.1	172.16.31.1	ICMP	98 Echo (ping) request id=0x2064, seq=4/1024, ttl=63 (reply in 213)
213	150.7801756...	172.16.31.1	172.16.30.1	ICMP	98 Echo (ping) reply id=0x2064, seq=4/1024, ttl=64 (request in 212)
214	151.3879854...	Cisco_b6:8c:03	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8003
215	151.8040334...	172.16.30.1	172.16.31.1	ICMP	98 Echo (ping) request id=0x2064, seq=5/1280, ttl=64 (reply in 216)
216	151.8041861...	172.16.31.1	172.16.30.1	ICMP	98 Echo (ping) reply id=0x2064, seq=5/1280, ttl=63 (request in 215)
217	151.8040529...	172.16.30.1	172.16.31.1	ICMP	98 Echo (ping) request id=0x2064, seq=5/1280, ttl=63 (reply in 218)
218	151.8041666...	172.16.31.1	172.16.30.1	ICMP	98 Echo (ping) reply id=0x2064, seq=5/1280, ttl=64 (request in 217)
219	152.3654450...	Cisco_b6:8c:02	Spanning-tree-(for...	STP	60 Conf. Root = 32768/30/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8002
220	152.911152...	HewlettP_5a:7d:b7	HewlettP_5a:7d:b7	ARP	42 Who has 172.16.30.1? Tell 172.16.30.254
221	152.9112855...	HewlettP_5a:7d:b7	HewlettP_5a:7d:b7	ARP	60 172.16.30.1 is at 00:21:5a:5a:7d:b7
222	152.9856548...	HewlettP_61:24:01	EncoreNe_b4:b8:94	ARP	60 Who has 172.16.31.253? Tell 172.16.31.1
223	152.9856746...	EncoreNe_b4:b8:94	HewlettP_61:24:01	ARP	42 172.16.31.253 is at 00:e0:7d:b4:b8:94
224	153.3887546...	Cisco_b6:8c:03	Spanning-tree-(for...	STP	60 Conf. Root = 32768/31/00:1e:14:b6:8c:00 Cost = 0 Port = 0x8003

Script para configuração do Router

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname gnu-rtr3
!
boot-start-marker
boot-end-marker
!
! card type command needed for slot/vwic-slot 0/0
logging message-counter syslog
logging buffered 51200 warnings
enable secret 5 $1$u53Q$vBawpP8.1YpCT6ypap1zX.
!
no aaa new-model
```

```
dot11 syslog
ip source-route
!
!
!
!
ip cef
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
!
!
!
!
!
!
username root privilege 15 secret 5 $1$8AFR$bNAYevxPFjXFExpnZI2fj.
username cisco password 7 02050D480809
archive
log config
hidekeys
!
!
!
!
!
!
!
!
!
!
interface FastEthernet0/0
description $ETH-LAN$$ETH-SW-LAUNCH$$INTF-INFO-FE 0$
ip address 172.16.31.254 255.255.255.0
ip nat inside
ip virtual-reassembly
duplex auto
speed auto
!
interface FastEthernet0/1
ip address 172.16.2.39 255.255.255.0
ip nat outside
```

```
ip virtual-reassembly
duplex auto
speed auto
!
ip forward-protocol nd
ip route 0.0.0.0 0.0.0.0 172.16.2.254
ip route 172.16.30.0 255.255.255.0 172.16.31.253
ip http server
ip http access-class 23
ip http authentication local
ip http secure-server
ip http timeout-policy idle 60 life 86400 requests 10000
!
!
ip nat pool ovrld 172.16.2.39 172.16.2.39 prefix-length 24
ip nat inside source list 1 pool ovrld overload
!
access-list 1 permit 172.16.30.0 0.0.0.7
access-list 1 permit 172.16.31.0 0.0.0.7
!
!
!
!
!
!
control-plane
!
!
!
line con 0
login local
line aux 0
line vty 0 4
access-class 23 in
privilege level 15
login local
transport input telnet ssh
line vty 5 15
access-class 23 in
privilege level 15
```

```
login local
transport input telnet ssh
!
scheduler allocate 20000 1000
end
```