2º Trabalho Laboratorial

FEUP - RC - Turma 7, Grupo 7

2021/22

Duarte Guedes Sardão
up201905497

José Pedro Ferreira
up201904515

Lucas Calvet Santos
up201904517

Índice

Sumário

O objetivo do segundo trabalho laboratorial foi desenvolver uma aplicação de download, assim como configurar e estudar uma rede de computadores que nos permitisse utilizar a aplicação desenvolvida. A aplicação de download utiliza o protocolo FTP (File Transfer Protocol) para transferir um ficheiro de um servidor que implementa esse protocolo. A rede de computadores configurada baseou-se em quatro experiências seguidas durante as aulas práticas, que foram analisadas utilizando os *logs* em anexo.

O desenvolvimento desta aplicação permitiu um conhecimento detalhado do protocolo associado, assim como do sistema de RFCs para pesquisa de padrões estabelecidos. As experiências realizadas na rede de computadores também proporcionaram o nosso conhecimento de vários protocolos essenciais, como o TCP/IP (Transmission Control Protocol/Internet Protocol), o ARP (Address Resolution Protocol) e o ICMP (Internet Control Message Protocol).

Introdução

No contexto da unidade curricular de Redes de Computadores, foi proposto o desenvolvimento de um programa de download por FTP, protocolo que faz parte da camada de aplicação do TCP/IP, assim como a configuração e análise de uma rede. Neste relatório é descrito o desenvolvimento da aplicação, assim como as quatro experiências realizadas em contexto laboratorial.

Parte 1 - Aplicação de download

Arquitetura da aplicação de download

A aplicação de download desenvolvida reside apenas num ficheiro, `download.c`. Este programa verifica que o URL FTP passado é válido e conecta-se, através da utilização de *sockets*, ao servidor indicado. De seguida, envia os comandos FTP necessários para transferir o ficheiro pretendido. Inicialmente, regista-se com as credenciais passadas (`anonymous` com password vazia se nenhumas credenciais foram passadas), e subsequentemente pede o respetivo ficheiro e envia um comando para entrar em modo passivo (`PASV`). Assim, se o servidor responder com sucesso, indica o endereço e porta para receber os dados, ao qual o programa se conecta e começa a transferir os mesmos para um ficheiro com o mesmo nome do original, no diretório de trabalho. Aquando da terminação da transferência e do recebimento da resposta do servidor que indica que a transferência foi terminada, envia-se um comando `QUIT` e fecha-se a *socket* para terminar a conexão.

Estas funcionalidades são obtidas com a ajuda de duas funções utilitárias principais:

- a `connection`, que recebe um endereço IP e uma porta, e retorna uma socket com uma conexão inicializada.

```
/*
 * Establishes a connection to a certain IP and port, and
     returns the respective socket.
 *
 * server_address: the server's address
 * server_port: the server's port
 *
 * returns: On success, a file descriptor for the new socket is
     returned. On error, -1 is returned, and errno is set to
     indicate the error.
 */
int connection(char *server_address, int server_port) {
    struct sockaddr_in server_addr;
    int sockfd;
    /*server address handling*/
    bzero((char *)&server_addr, sizeof(server_addr));
    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = inet_addr(server_address);
    server_addr.sin_port = htons(server_port);
    /*open a TCP socket*/
    if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
    {
        return -1;
    }
    /*connect to the server*/
    if (connect(sockfd, (struct sockaddr *)&server_addr,
        sizeof(server_addr)) < 0)
    {
        return -1;
    }
    return sockfd;
}
```

- a `read_reply`, que recebe informações sobre o *buffer* onde guardar a resposta e o *file stream* da *socket*, assim como o código previsto da resposta do servidor e uma mensagem de erro caso o código recebido não seja o mesmo. Esta função trata de ignorar o texto mandado pelo servior até receber o código de resposta, e retorna a resposta recebida.

```
/*
 * Reads a reply from the FTP server
 *
 * reply_buffer: a pointer to store the address of the buffer
     containing the reply
 * reply_buffer_size: a pointer to store the allocated size of
     the buffer
 * stream: the socket's file stream
 * expected_code: the reply code that is expected
 * error_msg: the message to show in case the reply code is not
     the expected one
 *
 * returns: On success, the number of characters read,
 * including the delimiter character, but not including the
     terminating null byte.
 * On error, -1 is returned, and errno is set to indicate the
     error.
 */
int read_reply(char **reply_buffer, size_t *reply_buffer_size,
    FILE *stream, char *expected_code, char *error_msg)
{
    regex_t reply_regex;
    regcomp(&reply_regex, "^[0-9]{3} ", REG_EXTENDED |
        REG_NOSUB);
    ssize_t nread;
    nread = getline(reply_buffer, reply_buffer_size, stream);
    while (nread >= 0 && regexec(&reply_regex, *reply_buffer, 0,
        NULL, 0) != 0)
    {
        free(*reply_buffer);
        *reply_buffer = NULL;
        nread = getline(reply_buffer, reply_buffer_size, stream);
    }
    if (nread == -1)
    {
        error(1, errno, "error reading the server's reply");
    }
    if (strncmp(*reply_buffer, expected_code, 3) != 0)
    {
        error(1, 0, "%s. Response: %s", error_msg,
            *reply_buffer);
    }
    return nread;
}
```

Descrição de um download bem sucedido

Foram realizados vários testes de download de ficheiros, de vários servidores FTP, tendo estes sido bem sucedidos.

Para compilar o programa basta utilizar o comando `make`, ou `make debug` para o compilar com a flag `DEBUG` e obter mais informação sobre a conexão com o servidor.

O programa deve ser executado com o seguinte formato:

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

- `user` e `password` - credenciais do utilizador, opcionais
- `host` - endereço do servidor FTP
- `url-path` - caminho do ficheiro pretendido

Exemplo de downloads bem sucedidos:

- ./download ftp://ftp.up.pt/pub/kodi/screenshots/kodi-addons.jpg
- ./download ftp://rcom:rcom@netlab1.fe.up.pt/files/crab.mp4

Parte 2 - Configuração e análise de uma rede

Experiência 1 - IP Config

O objetivo desta experiência era ligar os computadores gnu53 e gnu54 de uma subrede.

Configuração Inicial

Com os comandos `ifconfig` foi possível definir endereços IP e *netmasks* para cada computador e estabelecer rotas.

Verificou-se a conectividade através do comando `ping`, que confirmou o funcionamento dos anteriores comandos.

De seguida apagou-se a tabela ARP e fez-se ping outra vez: como as tabelas tinham sido apagadas, os protocolos ARP foram executados: estes protocolos mapeiam endereços ipv4 (camada de rede) a um endereço MAC (físico).

ARP e Tramas

Com o Wireshark analisou-se o funcionamento dos pacotes ARP: O gnu emissor envia em broadcast um protocolo ARP - os campos de IP e do próprio endereço MAC são preenchidos enquanto que o endereço MAC de destino, ainda desconhecido, é nulo.

Quando o computador com o IP de destino certo receber o protocolo, responderá com um protocolo ARP de retorno, desta vez com o endereço MAC definido. Este endereço será então registado pelo emissor original.

As tramas recebidas podem ser ARP, IP ou ICMP; a distinção verifica-se ao inspecionar o cabeçalho do pacote, onde o valor type é:

- `0x0806` para ARP
- `0x0800` para IP
- `0x0800` com 1 no campo tipo de serviço para ICMP

Os tamanhos das tramas podem ser analisados com o Wireshark. Também com o Wireshark verifica-se o ocasional envio de tramas loopback, que permitem um computador confirmar a correta configuração de rede ao receber respostas de si próprio.

Experiência 2 - Virtual LANs

A experiência 2 envolveu a configuração de duas VLAN (redes virtuais locais) no switch, a vlan50 e a vlan51. Os computadores gnu53 e gnu54 fazem parte da rede vlan50 e o gnu52 da rede vlan51.

Configuração Inicial

A configuração de cada vlan é feita no GTKTERM ligado ao switch, iniciada pelo comando `configure terminal`. Para criar uma rede basta escrever o comando `vlan`, procedimento que realizamos para criar ambas as redes. Para adicionar uma porta a uma das redes a sequência de comandos é a seguinte :

- `interface fa 0/[numero da porta]`
- `switchport mode access`
- `switchport access vlan [numero da rede]`

Nesta experiência, adicionamos a porta 1 (conectada ao gnu53) e a porta 2 (conectada ao gnu54) à vlan50 e a porta 13 (conectada ao gnu52) à vlan51. Por último para sair do modo de configuração do switch é necessário sair com o comando `end`.

Arquitetura de Rede

Como configuramos duas redes virtuais não comunicáveis entre si, existem dois domínios de broadcast. Assim, por exemplo, um ping em broadcast por parte do gnu53, apenas receberá uma resposta do gnu54 (ICMP reply). O gnu52 não envia resposta a este ping, pois não faz parte da mesma rede que o gnu53 e gnu54, o que é notório na ausência de log no Wireshark.

Experiência 3 - Router Configuration (Online)

A experiência decorreu à distância e englobou os seguintes tópicos:

- Análise de um ficheiro de configuração de um Router Cisco.
- Teste de registos DNS
- Configuração de rotas numa máquina Linux

Configuração Router Cisco

O objetivo desta parte da experiência foi analisar o ficheiro de configuração de um Router Cisco.

A partir do ficheiro de configuração do Router Cisco que nos foi fornecido, inicialmente identificamos: - O nome do Router: `gnu-rtr1` - As portas ethernet: duas do tipo `fast-ethernet` - Os endereços IP configurados e respetivas netmasks:

`172.16.30.1 255.255.255.0` e `172.16.254.45 255.255.255.0` - As rotas configuradas: `0.0.0.0 0.0.0.0 172.16.254.1` e `172.16.40.0 255.255.255.0 172.16.30.2`

De seguida identificamos que interface é que estava conectada com a internet, neste caso a `FastEthernet0/1` com address `172.16.254.45`, devido ao comando `ip nat outside`. Reparamos que apenas um endereço está disponível para NATing, como fica evidente analisando o comando `ip nat pool ovrld 172.16.254.45 172.16.254.45 prefix-length 24` e verificando que o range de endereços se limita a um: o `172.16.254.45`. **NAT** (Network Address Translation) é a tradução de um endereço privado num endereço público. Por último chegamos à conclusão, que o Router está a usar overloading.

Testes com registos DNS

Nesta parte da experiência é pretendido que testemos como efetuar corretamente um registo DNS.

Começamos por configurar a seguinte entrada no ficheiro `/etc/hosts`: `142.250.200.142 youtubas`. No Wireshark, verificamos que não existem pacotes DNS associados ao `ping youtubas`. Repetimos os dois primeiros passos, desta vez usando `enisa.europa.eu` e, desta feita, foram capturados pacotes DNS, mas com o endereço de destino `10.0.2.4`. Por último, alteramos o ficheiro `/etc/resolv.conf`, colocando o endereço `9.9.9.9` no topo do mesmo. Após este passo, ao experimentar um `ping` ao site do parlamento (`parlamento.pt`) já capturamos pacotes DNS com o endereço de destino `9.9.9.9`, concluindo que este último método é o correto para configurar o serviço DNS num cliente.

Configuração Linux Routing

A última parte da experiência visa a configuração de rotas numa máquina Linux.

Em primeiro lugar, usamos o comando `route` para ver a *routing table* atual. Em segundo, com o auxílio do comando `route del` apagamos a `default gateway 10.0.2.1`. Desta forma, não temos ligação à internet, considerando que nem o servidor DNS é contactável sem uma rota predefinida. Com o comando `route add 104.17.113.188 default gw eth0`, definimos uma rota entre `104.17.113.188` e a `default gateway`. Esta mudança, leva a que o `traceroute` já funcione e indique a `default gateway` inicial como parte da rota. Por fim, se adicionarmos o endereço de servidor `9.9.9.9` novamente ao ficheiro `/etc/resolv.conf`, já é este endereço a aparecer no `traceroute`.

Experiência 4 - Router Configuration (Lab)

Router Linux

O objetivo desta experiência foi configurar um computador Linux (gnu54) para servir de um router entre as VLANs configuradas na Experiência 2, permitindo a comunicação entre as duas sub-redes. Para tal, ligou-se a interface eth1 do gnu54 à vlan51 e o seu endereço IP foi definido com o comando `ifconfig 172.16.51.253/24`. Para ativar o encaminhamento IP, foram configurados os ficheiros necessários:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

De seguida, foram adicionadas rotas ao gnu53 e ao gnu 52, de maneira a indicar a *gateway* (gnu54) para estes se conectarem à outra vlan.

```
gnu53: ip route add 172.16.51.0/24 via 172.16.50.254
gnu52: ip route add 172.16.50.0/24 via 172.16.51.253
```

Analisando a tabela de rotas verifica-se que, utilizando o gnu3 como exemplo, a seguinte nova entrada foi registada:

Network Destination	Netmask	Destination Gateway	Destination Interface
172.16.51.0	255.255.255.0	172.16.50.254	eth0

Além destas colunas, ainda existe a coluna `Metric`, que indica a melhor rota, estando várias disponíveis.

Estas rotas permitem aos computadores saber que endereço IP é que devem utilizar para comunicar com uma certa rede. Assim, é possível a comunicação entre todas as interfaces de rede em uso. Para verificar essa conexão, geraram-se *pings* entre os três computadores. Estes *pings* (ICMP) desencadearam pedidos ARP entre os computadores, registados nos respetivos *logs*.

Router Cisco

Verificou-se inicialmente que as interfaces do Router Cisco estavam corretamente conectadas e as VLAN's configuradas. Alteramos o ficheiro de configuração anterior de acordo com o nosso número de bancada Y (5), e o nosso W, referente à sala (2). Confirmando a configuração a correr (show running-config), copiou-se para startup-config (copy running-config startup-config). Com `ping` testou-se a conetividade, fazendo ping aos gnu's, 172.16.2.254 e a 104.17.113.188

De seguida prosseguiu-se à configuração do gnu52 e gnu54, definindo a default gateway para o Router Cisco:

```
ip route add default via 172.16.51.254
```

No gnu53, definiu-se a default gateway para o gnu54:

```
ip route add default via 172.16.50.254
```

Com o gnu3, fez-se pings a 172.16.2.254 e 104.17.113.188.

Conclusões

Este trabalho laboratorial foi desenvolvido com sucesso: permitiu-nos melhor compreensão dos temas estudados nas aulas teóricas atravês das várias experiências e da sua conjugação na aplicação de download.

No nosso caso, o desenvolvimento da aplicação de download foi relativamente tranquilo, tendo constrastado com a dificuldade em conseguir realizar a parte laboratorial dentro do horário em que havia disponibilidade do espaço. Deparamo-nos com problemas técnicos aos quais éramos, na maioria dos casos, alheios, o que nos suscitou as referidas dificuldades. Vale salientar que essas mesmas adversidades geraram um espírito de entre-ajuda por parte de todos os grupos, que foi de louvar. Toda esta situação, todavia, deixou-nos a ideia de que se, aleatoriamente, tivessemos escolhido outra bancada na aula inicial, teríamos poupado bastante tempo nas preparações.

Em conclusão, cumprimos os objetivos propostos e adquirimos conhecimentos importantes teóricos e práticos, no contexto da Unidade Curricular de Redes de Computadores.

Anexo I - Código fonte

```c
#include <stdio.h>
#include <stdlib.h>

#include <sys/socket.h>
#include <netdb.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <string.h>
#include <regex.h>
#include <error.h>
#include <errno.h>
#include <libgen.h>
#include <fcntl.h>

#define BUFFER_SIZE 256

/*
 * Establishes a connection to a certain IP and port, and
     returns the respective socket.
 *
 * server_address: the server's address
 * server_port: the server's port
 *
 * returns: On success, a file descriptor for the new socket is
```

```
     returned. On error, -1 is returned, and errno is set to
     indicate the error.
 */
int connection(char *server_address, int server_port)
{
    struct sockaddr_in server_addr;
    int sockfd;

    /*server address handling*/
    bzero((char *)&server_addr, sizeof(server_addr));
    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = inet_addr(server_address);
        /*32 bit Internet address network byte ordered*/
    server_addr.sin_port = htons(server_port);
        /*server TCP port must be network byte ordered */

    /*open a TCP socket*/
    if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
    {
        return -1;
    }

    /*connect to the server*/
    if (connect(sockfd, (struct sockaddr *)&server_addr,
        sizeof(server_addr)) < 0)
    {
        return -1;
    }

    return sockfd;
}

/*
 * Reads a reply from the FTP server
 *
 * reply_buffer: a pointer to store the address of the buffer
     containing the reply
 * reply_buffer_size: a pointer to store the allocated size of
     the buffer
 * stream: the socket's file stream
 * expected_code: the reply code that is expected
 * error_msg: the message to show in case the reply code is not
     the expected one
 *
 * returns: On success, the number of characters read,
 * including the delimiter character, but not including the
```

```
     terminating null byte.
 * On error, -1 is returned, and errno is set to indicate the
     error.
 */
int read_reply(char **reply_buffer, size_t *reply_buffer_size,
    FILE *stream, char *expected_code, char *error_msg)
{
    regex_t reply_regex;
    regcomp(&reply_regex, "^[0-9]{3} ", REG_EXTENDED |
        REG_NOSUB);

    ssize_t nread;
    nread = getline(reply_buffer, reply_buffer_size, stream);
    while (nread >= 0 && regexec(&reply_regex, *reply_buffer, 0,
        NULL, 0) != 0)
    {
        free(*reply_buffer);
        *reply_buffer = NULL;
        nread = getline(reply_buffer, reply_buffer_size, stream);
    }

    if (nread == -1)
    {
        error(1, errno, "error reading the server's reply");
    }

    if (strncmp(*reply_buffer, expected_code, 3) != 0)
    {
        error(1, 0, "%s. Response: %s", error_msg,
            *reply_buffer);
    }

    return nread;
}

int main(int argc, char **argv)
{
    if (argc != 2)
    {
        fprintf(stderr, "Invalid number of arguments.\n\nUsage:
            %s ftp://[<user>:<password>@]<host>/<url-path>\n",
            argv[0]);
        exit(-1);
    }

    regex_t regex;
```

```
regcomp(&regex,
    "^ftp://((([[:alnum:]+._-][[:alnum:]+._-]*)(:([[:alnum:]+._-]*)){0,1}){0,1}@){0,1}([
    REG_EXTENDED);

regmatch_t url_regmatch[8];

if (regexec(&regex, argv[1], 8, url_regmatch, 0) != 0)
{
    fprintf(stderr, "Invalid ftp URL.\n\nUsage: %s
        ftp://[<user>:<password>@]<host>/<url-path>\n",
        argv[0]);
    exit(-1);
}

char *user, *password, *host, *path;
int len;

/* Check if the login details were not given */
if (url_regmatch[1].rm_so == -1)
{
    user = (char *)malloc(10 * sizeof(char));
    memcpy(user, "anonymous", 10);

    password = (char *)malloc(1 * sizeof(char));
    memcpy(password, "", 1);
}
/* Check if the user is not empty  */
if (url_regmatch[3].rm_so != -1)
{
    len = url_regmatch[3].rm_eo - url_regmatch[3].rm_so;
    user = (char *)malloc((len + 1) * sizeof(char));
    memcpy(user, argv[1] + url_regmatch[3].rm_so *
        sizeof(char), len);
    user[len] = 0;
}

/* Check if the password is present */
if (url_regmatch[4].rm_so != -1)
{
    len = url_regmatch[5].rm_eo - url_regmatch[5].rm_so;
    password = (char *)malloc((len + 1) * sizeof(char));
    memcpy(password, argv[1] + url_regmatch[5].rm_so *
        sizeof(char), len);
    password[len] = 0;
}
```

```
    else
    {
        password = (char *)malloc(1 * sizeof(char));
        memcpy(password, "", 1);
    }

    len = url_regmatch[6].rm_eo - url_regmatch[6].rm_so;
    host = (char *)malloc((len + 1) * sizeof(char));
    memcpy(host, argv[1] + url_regmatch[6].rm_so * sizeof(char),
        len);
    host[len] = 0;

    len = url_regmatch[7].rm_eo - url_regmatch[7].rm_so;
    path = (char *)malloc((len + 1) * sizeof(char));
    memcpy(path, argv[1] + url_regmatch[7].rm_so * sizeof(char),
        len);
    path[len] = 0;

#ifdef DEBUG
    printf("Parsed input:\n\tUser: %s\n\tPass: %s\n\tHost:
        %s\n\tPath: %s\n\n", user, password, host, path);
#endif

    struct hostent *h;

    if ((h = gethostbyname(host)) == NULL)
    {
        error(1, errno, "error getting hostname");
    }
    free(host);

#ifdef DEBUG
    printf("Resolved host:\n\tHost name: %s\n\tIP Address:
        %s\n", h->h_name, inet_ntoa(*((struct in_addr
        *)h->h_addr)));
#endif

    int socket_fd = connection(inet_ntoa(*((struct in_addr
        *)h->h_addr)), 21);

    if (socket_fd == -1)
    {
        error(1, errno, "connection()");
    }

    char *reply = NULL;
    size_t reply_len = 0;
    FILE *fp = fdopen(socket_fd, "r");

    read_reply(&reply, &reply_len, fp, "220", "server not ready
        for commands");
    free(reply);
    reply = NULL;

    dprintf(socket_fd, "user %s\r\n", user);
    read_reply(&reply, &reply_len, fp, "331", "login was
        unsuccessful");
    free(reply);
    reply = NULL;
    free(user);

    dprintf(socket_fd, "pass %s\r\n", password);
    read_reply(&reply, &reply_len, fp, "230", "login was
        unsuccessful");
    free(reply);
    reply = NULL;
    free(password);

    dprintf(socket_fd, "pasv\r\n");
    read_reply(&reply, &reply_len, fp, "227", "error entering
        passive mode");

    regcomp(&regex,
        "\\(([0-9]*),([0-9]*),([0-9]*),([0-9]*),([0-9]*),([0-9]*)\\)",
        REG_EXTENDED);

    regmatch_t ip_regmatch[7];

    if (regexec(&regex, reply, 7, ip_regmatch, 0) != 0)
    {
        error(1, 0, "there was no match for an ip and port in
            227 response. Response: %s", reply);
    }

    len = ip_regmatch[4].rm_eo - ip_regmatch[1].rm_so;
    char *ip, *port_1, *port_2;

    ip = (char *)malloc((len + 1) * sizeof(char));

    memcpy(ip, reply + ip_regmatch[1].rm_so * sizeof(char), len);
    ip[ip_regmatch[1].rm_eo - ip_regmatch[1].rm_so] = '.';
    ip[ip_regmatch[2].rm_eo - ip_regmatch[1].rm_so] = '.';
    ip[ip_regmatch[3].rm_eo - ip_regmatch[1].rm_so] = '.';
    ip[len] = 0;

    len = ip_regmatch[5].rm_eo - ip_regmatch[5].rm_so;
    port_1 = (char *)malloc((len + 1) * sizeof(char));
    memcpy(port_1, reply + ip_regmatch[5].rm_so * sizeof(char),
        len);
    port_1[len] = 0;
    len = ip_regmatch[6].rm_eo - ip_regmatch[6].rm_so;
    port_2 = (char *)malloc((len + 1) * sizeof(char));
    memcpy(port_2, reply + ip_regmatch[6].rm_so * sizeof(char),
        len);
    port_2[len] = 0;

    int port = 256 * atoi(port_1) + atoi(port_2);

    free(port_1);
    free(port_2);
    free(reply);
    reply = NULL;

#ifdef DEBUG
    printf("\tFile address: %s:%d\n\n", ip, port);
#endif

    char *filename = basename(path);

    int file_socket_fd = connection(ip, port);
    if (file_socket_fd == -1)
    {
        error(1, errno, "error connecting to file server");
    }
    free(ip);

    dprintf(socket_fd, "retr %s\r\n", path);
    read_reply(&reply, &reply_len, fp, "150", "error retrieving
        requested file");
    free(reply);
    reply = NULL;

    printf("Downloading to %s...\n", filename);
    int output_fd = creat(filename, 0666);
    if (output_fd == -1)
    {
        error(1, errno, "error creating new file");
    }
    free(path);

    char readbuf[BUFFER_SIZE];
    size_t nread;
    while ((nread = read(file_socket_fd, readbuf, BUFFER_SIZE))
        > 0)
    {
        if (write(output_fd, readbuf, nread) == -1)
        {
            error(1, errno, "error writing to the new file");
        }
    }

    if (nread == -1)
    {
        error(1, errno, "error reading the file from the
            server");
    }

    if (close(output_fd) == -1)
    {
        error(1, errno, "error closing the output file");
    }

    if (close(file_socket_fd) == -1)
    {
        error(1, errno, "error closing the connection to the
            file server");
    }

    read_reply(&reply, &reply_len, fp, "226", "error completing
        transfer");
    free(reply);
    reply = NULL;
    dprintf(socket_fd, "quit\r\n");

    if (close(socket_fd) == -1)
    {
        error(1, errno, "error closing the connection to the
            server");
    }

    printf("Transfer completed! Exiting.\n");
    return 0;
}
```

Anexo II - Comandos de configuração

Configuração do Switch

```
enable
configure terminal
vlan 50
vlan 51
interface fa 0/1
switchport mode access
switchport access vlan 50
interface fa 0/2
switchport mode access
switchport access vlan 50
interface fa 0/13
switchport mode access
switchport access vlan 51
interface fa 0/14
switchport mode access
switchport access vlan 51
end
```

Configuração do Router

```
// Configuring NAT inside
conf t
interface fa 0/0
ip address 172.16.51.254 255.255.255.0
no shutdown
ip nat inside
exit

// Configuring NAT outside
interface fa 0/1
ip address 172.16.2.59 255.255.255.0
no shutdown
ip nat outside
exit

// Configuring nat properties
ip nat pool ovrld 172.16.2.59 172.16.2.59 prefix 24
ip nat inside source list 1 pool ovrld overload

// Declaring the access list
access-list 1 permit 172.16.50.0 0.0.0.7
access-list 1 permit 172.16.51.0 0.0.0.7
```

```
// Configuring router IP routing
ip route 0.0.0.0 0.0.0.0 172.16.2.254
ip route 172.16.50.0 255.255.255.0 172.16.51.253
end
```

Anexo III - Logs Capturados

Experiência 1

```
No Time           Source                Destination
    Protocol Length Info
24 33.310806486   HewlettP_61:2c:54     Broadcast
    ARP      42     Who has 172.16.50.254? Tell 172.16.50.1
25 33.310940650   HewlettP_19:09:5c     HewlettP_61:2c:54
    ARP      60     172.16.50.254 is at 00:22:64:19:09:5c
26 33.310957691   172.16.50.1           172.16.50.254
    ICMP     98     Echo (ping) request  id=0x1ab5, seq=1/256,
    ttl=64 (reply in 27)
27 33.311097302   172.16.50.254         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x1ab5, seq=1/256,
    ttl=64 (request in 26)
28 34.083043616   Cisco_7b:d5:01
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/1/00:1e:14:7b:d5:00  Cost = 0  Port = 0x8001
29 34.167162937   Cisco_7b:d5:01        Cisco_7b:d5:01
    LOOP     60     Reply
30 34.314122381   172.16.50.1           172.16.50.254
    ICMP     98     Echo (ping) request  id=0x1ab5, seq=2/512,
    ttl=64 (reply in 31)
31 34.314252215   172.16.50.254         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x1ab5, seq=2/512,
    ttl=64 (request in 30)
32 35.338119077   172.16.50.1           172.16.50.254
    ICMP     98     Echo (ping) request  id=0x1ab5, seq=3/768,
    ttl=64 (reply in 33)
33 35.338281527   172.16.50.254         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x1ab5, seq=3/768,
    ttl=64 (request in 32)
34 36.088241158   Cisco_7b:d5:01
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/1/00:1e:14:7b:d5:00  Cost = 0  Port = 0x8001
35 36.362123945   172.16.50.1           172.16.50.254
    ICMP     98     Echo (ping) request  id=0x1ab5, seq=4/1024,
    ttl=64 (reply in 36)
36 36.362253569   172.16.50.254         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x1ab5, seq=4/1024,
    ttl=64 (request in 35)
```

```
37 37.386120781   172.16.50.1           172.16.50.254
   ICMP     98     Echo (ping) request  id=0x1ab5, seq=5/1280,
   ttl=64 (reply in 38)
38 37.386252430   172.16.50.254         172.16.50.1
   ICMP     98     Echo (ping) reply    id=0x1ab5, seq=5/1280,
   ttl=64 (request in 37)
39 38.092847918   Cisco_7b:d5:01
   Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
   32768/1/00:1e:14:7b:d5:00  Cost = 0  Port = 0x8001
40 38.410138709   172.16.50.1           172.16.50.254
   ICMP     98     Echo (ping) request  id=0x1ab5, seq=6/1536,
   ttl=64 (reply in 41)
41 38.410272244   172.16.50.254         172.16.50.1
   ICMP     98     Echo (ping) reply    id=0x1ab5, seq=6/1536,
   ttl=64 (request in 40)
42 38.473037289   HewlettP_19:09:5c     HewlettP_61:2c:54
   ARP      60     Who has 172.16.50.1? Tell 172.16.50.254
43 38.473044762   HewlettP_61:2c:54     HewlettP_19:09:5c
   ARP      42     172.16.50.1 is at 00:21:5a:61:2c:54
44 39.434123392   172.16.50.1           172.16.50.254
   ICMP     98     Echo (ping) request  id=0x1ab5, seq=7/1792,
   ttl=64 (reply in 45)
45 39.434256928   172.16.50.254         172.16.50.1
   ICMP     98     Echo (ping) reply    id=0x1ab5, seq=7/1792,
   ttl=64 (request in 44)
46 40.097699609   Cisco_7b:d5:01
   Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
   32768/1/00:1e:14:7b:d5:00  Cost = 0  Port = 0x8001
47 40.408097474   Cisco_7b:d5:01        CDP/VTP/DTP/PAgP/UDLD
   DTP      60     Dynamic Trunk Protocol
48 40.408198673   Cisco_7b:d5:01        CDP/VTP/DTP/PAgP/UDLD
   DTP      90     Dynamic Trunk Protocol
49 40.458120647   172.16.50.1           172.16.50.254
   ICMP     98     Echo (ping) request  id=0x1ab5, seq=8/2048,
   ttl=64 (reply in 50)
50 40.458273878   172.16.50.254         172.16.50.1
   ICMP     98     Echo (ping) reply    id=0x1ab5, seq=8/2048,
   ttl=64 (request in 49)
51 41.482120975   172.16.50.1           172.16.50.254
   ICMP     98     Echo (ping) request  id=0x1ab5, seq=9/2304,
   ttl=64 (reply in 52)
52 41.482257304   172.16.50.254         172.16.50.1
   ICMP     98     Echo (ping) reply    id=0x1ab5, seq=9/2304,
   ttl=64 (request in 51)
53 42.106598427   Cisco_7b:d5:01
   Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
   32768/1/00:1e:14:7b:d5:00  Cost = 0  Port = 0x8001
54 42.506119906   172.16.50.1           172.16.50.254
   ICMP     98     Echo (ping) request  id=0x1ab5, seq=10/2560,
   ttl=64 (reply in 55)
55 42.506245969   172.16.50.254         172.16.50.1
   ICMP     98     Echo (ping) reply    id=0x1ab5, seq=10/2560,
   ttl=64 (request in 54)
56 43.530122120   172.16.50.1           172.16.50.254
   ICMP     98     Echo (ping) request  id=0x1ab5, seq=11/2816,
   ttl=64 (reply in 57)
57 43.530256354   172.16.50.254         172.16.50.1
   ICMP     98     Echo (ping) reply    id=0x1ab5, seq=11/2816,
   ttl=64 (request in 56)
58 44.107526120   Cisco_7b:d5:01
   Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
   32768/1/00:1e:14:7b:d5:00  Cost = 0  Port = 0x8001
59 44.174732194   Cisco_7b:d5:01        Cisco_7b:d5:01
   LOOP     60     Reply
60 44.554120353   172.16.50.1           172.16.50.254
   ICMP     98     Echo (ping) request  id=0x1ab5, seq=12/3072,
   ttl=64 (reply in 61)
61 44.554280078   172.16.50.254         172.16.50.1
   ICMP     98     Echo (ping) reply    id=0x1ab5, seq=12/3072,
   ttl=64 (request in 60)
```

Experiência 2

```
No Time           Source                Destination
    Protocol Length Info
15 23.722413969   172.16.50.1           172.16.50.254
    ICMP     98     Echo (ping) request  id=0x1e24, seq=1/256,
    ttl=64 (reply in 16)
16 23.722584659   172.16.50.254         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x1e24, seq=1/256,
    ttl=64 (request in 15)
17 24.062751414   Cisco_7b:d5:01
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/50/00:1e:14:7b:d5:00  Cost = 0  Port = 0x8001
18 24.732080965   172.16.50.1           172.16.50.254
    ICMP     98     Echo (ping) request  id=0x1e24, seq=2/512,
    ttl=64 (reply in 19)
19 24.732240900   172.16.50.254         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x1e24, seq=2/512,
    ttl=64 (request in 18)
20 25.756089395   172.16.50.1           172.16.50.254
    ICMP     98     Echo (ping) request  id=0x1e24, seq=3/768,
    ttl=64 (reply in 21)
21 25.756224467   172.16.50.254         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x1e24, seq=3/768,
    ttl=64 (request in 20)
22 26.063537190   Cisco_7b:d5:01
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/50/00:1e:14:7b:d5:00  Cost = 0  Port = 0x8001
23 26.780092307   172.16.50.1           172.16.50.254
    ICMP     98     Echo (ping) request  id=0x1e24, seq=4/1024,
    ttl=64 (reply in 24)
24 26.780223398   172.16.50.254         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x1e24, seq=4/1024,
    ttl=64 (request in 23)
25 28.068400055   Cisco_7b:d5:01
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/50/00:1e:14:7b:d5:00  Cost = 0  Port = 0x8001
26 28.827727487   HewlettP_19:09:5c     HewlettP_61:2c:54
    ARP      60     Who has 172.16.50.1? Tell 172.16.50.254
27 28.827747810   HewlettP_61:2c:54     HewlettP_19:09:5c
    ARP      42     172.16.50.1 is at 00:21:5a:61:2c:54
28 28.924055655   HewlettP_61:2c:54     HewlettP_19:09:5c
    ARP      42     Who has 172.16.50.254? Tell 172.16.50.1
29 28.924144981   HewlettP_19:09:5c     HewlettP_61:2c:54
    ARP      60     172.16.50.254 is at 00:22:64:19:09:5c
```

Experiência 3

```
1 0.000000000    10.0.2.4              142.250.200.142
    ICMP     98     Echo (ping) request  id=0x0008, seq=1/256,
    ttl=64 (reply in 2)
 2 0.020601661    142.250.200.142       10.0.2.4
     ICMP     98     Echo (ping) reply    id=0x0008, seq=1/256,
     ttl=58 (request in 1)
 3 1.001365563    10.0.2.4              142.250.200.142
     ICMP     98     Echo (ping) request  id=0x0008, seq=2/512,
     ttl=64 (reply in 4)
 4 1.021110666    142.250.200.142       10.0.2.4
     ICMP     98     Echo (ping) reply    id=0x0008, seq=2/512,
     ttl=58 (request in 3)
 5 2.003131425    10.0.2.4              142.250.200.142
     ICMP     98     Echo (ping) request  id=0x0008, seq=3/768,
     ttl=64 (reply in 6)
 6 2.023176628    142.250.200.142       10.0.2.4
     ICMP     98     Echo (ping) reply    id=0x0008, seq=3/768,
     ttl=58 (request in 5)
 7 3.006825290    10.0.2.4              142.250.200.142
```

```
    ICMP     98     Echo (ping) request  id=0x0008, seq=4/1024,
    ttl=64 (reply in 8)
 8 3.027420714    142.250.200.142       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0008, seq=4/1024,
    ttl=58 (request in 7)
 9 4.010360541    10.0.2.4              142.250.200.142
    ICMP     98     Echo (ping) request  id=0x0008, seq=5/1280,
    ttl=64 (reply in 10)
10 4.031128601    142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=5/1280,
   ttl=58 (request in 9)
11 5.008326822    PcsCompu_2b:a7:69     RealtekU_12:35:00
   ARP      42     Who has 10.0.2.1? Tell 10.0.2.4
12 5.008760177    RealtekU_12:35:00     PcsCompu_2b:a7:69
   ARP      60     10.0.2.1 is at 52:54:00:12:35:00
13 5.013227611    10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=6/1536,
   ttl=64 (reply in 14)
14 5.033888355    142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=6/1536,
   ttl=58 (request in 13)
15 6.048180732    10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=7/1792,
   ttl=64 (reply in 16)
16 6.068574209    142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=7/1792,
   ttl=58 (request in 15)
17 7.050729296    10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=8/2048,
   ttl=64 (reply in 18)
18 7.070541429    142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=8/2048,
   ttl=58 (request in 17)
19 8.051775712    10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=9/2304,
   ttl=64 (reply in 20)
20 8.071807636    142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=9/2304,
   ttl=58 (request in 19)
21 9.065304226    10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=10/2560,
   ttl=64 (reply in 22)
22 9.086522385    142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=10/2560,
   ttl=58 (request in 21)
23 10.066694342   10.0.2.4              142.250.200.142
```

```
   ICMP     98     Echo (ping) request  id=0x0008, seq=11/2816,
   ttl=64 (reply in 24)
24 10.086461923   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=11/2816,
   ttl=58 (request in 23)
25 11.067847791   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=12/3072,
   ttl=64 (reply in 26)
26 11.088010201   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=12/3072,
   ttl=58 (request in 25)
27 12.071100347   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=13/3328,
   ttl=64 (reply in 28)
28 12.091203470   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=13/3328,
   ttl=58 (request in 27)
29 13.076086580   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=14/3584,
   ttl=64 (reply in 30)
30 13.095927811   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=14/3584,
   ttl=58 (request in 29)
31 14.077394072   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=15/3840,
   ttl=64 (reply in 32)
32 14.097328311   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=15/3840,
   ttl=58 (request in 31)
33 15.134092368   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=16/4096,
   ttl=64 (reply in 34)
34 15.154776745   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=16/4096,
   ttl=58 (request in 33)
35 16.140464456   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=17/4352,
   ttl=64 (reply in 36)
36 16.160644925   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=17/4352,
   ttl=58 (request in 35)
37 17.220168351   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=18/4608,
   ttl=64 (reply in 38)
38 17.240946170   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=18/4608,
   ttl=58 (request in 37)
39 18.221393234   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=19/4864,
   ttl=64 (reply in 40)
40 18.241931647   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=19/4864,
   ttl=58 (request in 39)
41 19.223049116   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=20/5120,
   ttl=64 (reply in 42)
42 19.243151713   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=20/5120,
   ttl=58 (request in 41)
43 20.224821616   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=21/5376,
   ttl=64 (reply in 44)
44 20.245536754   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=21/5376,
   ttl=58 (request in 43)
45 21.229443726   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=22/5632,
   ttl=64 (reply in 46)
46 21.249404664   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=22/5632,
   ttl=58 (request in 45)
47 22.231131629   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=23/5888,
   ttl=64 (reply in 48)
48 22.250861280   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=23/5888,
   ttl=58 (request in 47)
49 23.234071857   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=24/6144,
   ttl=64 (reply in 50)
50 23.254585650   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=24/6144,
   ttl=58 (request in 49)
51 24.236482040   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=25/6400,
   ttl=64 (reply in 52)
52 24.256523477   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=25/6400,
   ttl=58 (request in 51)
53 25.237978626   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=26/6656,
   ttl=64 (reply in 54)
```

```
54 25.257568934   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=26/6656,
   ttl=58 (request in 53)
55 26.239935750   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=27/6912,
   ttl=64 (reply in 56)
56 26.260612233   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=27/6912,
   ttl=58 (request in 55)
57 27.244948501   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=28/7168,
   ttl=64 (reply in 58)
58 27.265532578   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=28/7168,
   ttl=58 (request in 57)
59 28.247184707   10.0.2.4              142.250.200.142
   ICMP     98     Echo (ping) request  id=0x0008, seq=29/7424,
   ttl=64 (reply in 60)
60 28.267058540   142.250.200.142       10.0.2.4
   ICMP     98     Echo (ping) reply    id=0x0008, seq=29/7424,
   ttl=58 (request in 59)
```

```
No Time           Source                Destination
   Protocol Length Info
 1 0.000000000    10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=1/256,
    ttl=64 (reply in 2)
 2 0.107039262    212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=1/256,
    ttl=48 (request in 1)
 3 0.108319934    10.0.2.4              62.169.70.160
    DNS      99     Standard query 0xa733 PTR
    104.105.146.212.in-addr.arpa OPT
 4 1.119325486    62.169.70.160         10.0.2.4
    DNS      128    Standard query response 0xa733 PTR
    104.105.146.212.in-addr.arpa PTR enisa.europa.eu OPT
 5 1.120104472    10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=2/512,
    ttl=64 (reply in 6)
 6 1.194446739    212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=2/512,
    ttl=48 (request in 5)
 7 2.121634348    10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=3/768,
    ttl=64 (reply in 8)
 8 2.196064949    212.146.105.104       10.0.2.4
```

```
     ICMP     98     Echo (ping) reply    id=0x0009, seq=3/768,
     ttl=48 (request in 7)
 9 3.124537879    10.0.2.4              212.146.105.104
     ICMP     98     Echo (ping) request  id=0x0009, seq=4/1024,
     ttl=64 (reply in 10)
10 3.197542340    212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=4/1024,
    ttl=48 (request in 9)
11 4.125798878    10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=5/1280,
    ttl=64 (reply in 12)
12 4.198944729    212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=5/1280,
    ttl=48 (request in 11)
13 5.083594116    PcsCompu_2b:a7:69     RealtekU_12:35:00
    ARP      42     Who has 10.0.2.1? Tell 10.0.2.4
14 5.084477589    RealtekU_12:35:00     PcsCompu_2b:a7:69
    ARP      60     10.0.2.1 is at 52:54:00:12:35:00
15 5.128047922    10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=6/1536,
    ttl=64 (reply in 16)
16 5.203049996    212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=6/1536,
    ttl=48 (request in 15)
17 6.130199248    10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=7/1792,
    ttl=64 (reply in 18)
18 6.204260107    212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=7/1792,
    ttl=48 (request in 17)
19 7.161956860    10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=8/2048,
    ttl=64 (reply in 20)
20 7.235901011    212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=8/2048,
    ttl=48 (request in 19)
21 8.163695026    10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=9/2304,
    ttl=64 (reply in 22)
22 8.237700346    212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=9/2304,
    ttl=48 (request in 21)
23 9.164645457    10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=10/2560,
    ttl=64 (reply in 24)
24 9.238656505    212.146.105.104       10.0.2.4
```

```
    ICMP     98     Echo (ping) reply    id=0x0009, seq=10/2560,
    ttl=48 (request in 23)
25 10.168845999   10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=11/2816,
    ttl=64 (reply in 26)
26 10.242979780   212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=11/2816,
    ttl=48 (request in 25)
27 11.170487423   10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=12/3072,
    ttl=64 (reply in 28)
28 11.244100801   212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=12/3072,
    ttl=48 (request in 27)
29 12.172241650   10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=13/3328,
    ttl=64 (reply in 30)
30 12.245356659   212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=13/3328,
    ttl=48 (request in 29)
31 13.181419400   10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=14/3584,
    ttl=64 (reply in 32)
32 13.254616090   212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=14/3584,
    ttl=48 (request in 31)
33 14.185837128   10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=15/3840,
    ttl=64 (reply in 34)
34 14.259544199   212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=15/3840,
    ttl=48 (request in 33)
35 15.197764554   10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=16/4096,
    ttl=64 (reply in 36)
36 15.271835148   212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=16/4096,
    ttl=48 (request in 35)
37 16.199548088   10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=17/4352,
    ttl=64 (reply in 38)
38 16.272902644   212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=17/4352,
    ttl=48 (request in 37)
39 17.200379522   10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=18/4608,
    ttl=64 (reply in 40)
40 17.274438395   212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=18/4608,
    ttl=48 (request in 39)
41 18.201221224   10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=19/4864,
    ttl=64 (reply in 42)
42 18.274619642   212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=19/4864,
    ttl=48 (request in 41)
43 19.203464874   10.0.2.4              212.146.105.104
    ICMP     98     Echo (ping) request  id=0x0009, seq=20/5120,
    ttl=64 (reply in 44)
44 19.276920376   212.146.105.104       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x0009, seq=20/5120,
    ttl=48 (request in 43)
```

```
No Time           Source                Destination
    Protocol Length Info
 1 0.000000000    10.0.2.4              216.21.3.77
     TLSv1.2  85     Encrypted Alert
 2 0.000304739    10.0.2.4              216.21.3.77
     TCP      54     57734 -> 443 [FIN, ACK] Seq=32 Ack=1
     Win=62780 Len=0
 3 0.000951757    216.21.3.77           10.0.2.4
     TCP      60     443 -> 57734 [ACK] Seq=1 Ack=33 Win=32093
     Len=0
 4 0.001671802    10.0.2.4              216.21.3.77
     TLSv1.2  85     Encrypted Alert
 5 0.001929592    10.0.2.4              216.21.3.77
     TCP      54     57732 -> 443 [FIN, ACK] Seq=32 Ack=1
     Win=62780 Len=0
 6 0.002162015    216.21.3.77           10.0.2.4
     TCP      60     443 -> 57732 [ACK] Seq=1 Ack=33 Win=32093
     Len=0
 7 0.165530554    216.21.3.77           10.0.2.4
     TLSv1.2  270    Application Data
 8 0.165557390    10.0.2.4              216.21.3.77
     TCP      54     57734 -> 443 [RST] Seq=33 Win=0 Len=0
 9 0.168770583    216.21.3.77           10.0.2.4
     TLSv1.2  301    Application Data, Encrypted Alert
10 0.168802033    10.0.2.4              216.21.3.77
    TCP      54     57732 -> 443 [RST] Seq=33 Win=0 Len=0
11 2.558322192    10.0.2.4              9.9.9.9
    DNS      73     Standard query 0xfbc7 A parlamento.pt
12 2.558426619    10.0.2.4              9.9.9.9
```

```
    DNS      73     Standard query 0x1cd9 AAAA parlamento.pt
13 2.585301271    9.9.9.9              10.0.2.4
    DNS      89     Standard query response 0xfbc7 A
    parlamento.pt A 88.157.195.115
14 2.625595562    9.9.9.9              10.0.2.4
    DNS      123    Standard query response 0x1cd9 AAAA
    parlamento.pt SOA ns2.parlamento.pt
15 2.626007543    10.0.2.4             88.157.195.115
    ICMP     98     Echo (ping) request  id=0x000e, seq=1/256,
    ttl=64 (reply in 16)
16 2.645968622    88.157.195.115       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x000e, seq=1/256,
    ttl=121 (request in 15)
17 2.646642968    10.0.2.4             9.9.9.9
    DNS      87     Standard query 0xe5df PTR
    115.195.157.88.in-addr.arpa
18 2.692567041    9.9.9.9              10.0.2.4
    DNS      157    Standard query response 0xe5df PTR
    115.195.157.88.in-addr.arpa PTR parlamento.pt PTR
    www.parlamento.pt PTR biblioteca.parlamento.pt
19 3.634304441    10.0.2.4             88.157.195.115
    ICMP     98     Echo (ping) request  id=0x000e, seq=2/512,
    ttl=64 (reply in 20)
20 3.656005096    88.157.195.115       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x000e, seq=2/512,
    ttl=121 (request in 19)
21 4.635584333    10.0.2.4             88.157.195.115
    ICMP     98     Echo (ping) request  id=0x000e, seq=3/768,
    ttl=64 (reply in 22)
22 4.648954212    88.157.195.115       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x000e, seq=3/768,
    ttl=121 (request in 21)
23 5.636701733    10.0.2.4             88.157.195.115
    ICMP     98     Echo (ping) request  id=0x000e, seq=4/1024,
    ttl=64 (reply in 24)
24 5.650374161    88.157.195.115       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x000e, seq=4/1024,
    ttl=121 (request in 23)
25 6.638748234    10.0.2.4             88.157.195.115
    ICMP     98     Echo (ping) request  id=0x000e, seq=5/1280,
    ttl=64 (reply in 26)
26 6.652163571    88.157.195.115       10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x000e, seq=5/1280,
    ttl=121 (request in 25)
27 7.644416550    10.0.2.4             88.157.195.115
    ICMP     98     Echo (ping) request  id=0x000e, seq=6/1536,
    ttl=64 (reply in 28)
28 7.664253139    88.157.195.115        10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x000e, seq=6/1536,
    ttl=121 (request in 27)
29 8.646247416    10.0.2.4              88.157.195.115
    ICMP     98     Echo (ping) request  id=0x000e, seq=7/1792,
    ttl=64 (reply in 30)
30 8.659803290    88.157.195.115        10.0.2.4
    ICMP     98     Echo (ping) reply    id=0x000e, seq=7/1792,
    ttl=121 (request in 29)
```

Experiência 4

```
No Time           Source                Destination
    Protocol Length Info
13 14.841622737   172.16.50.1           172.16.50.254
    ICMP     98     Echo (ping) request  id=0x2a5f, seq=1/256,
    ttl=64 (reply in 14)
14 14.841795525   172.16.50.254         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a5f, seq=1/256,
    ttl=64 (request in 13)
15 15.225100343   Cisco_78:94:83
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8003
16 15.846834573   172.16.50.1           172.16.50.254
    ICMP     98     Echo (ping) request  id=0x2a5f, seq=2/512,
    ttl=64 (reply in 17)
17 15.846972719   172.16.50.254         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a5f, seq=2/512,
    ttl=64 (request in 16)
18 15.898314164   0.0.0.0               255.255.255.255
    DHCP     342    DHCP Discover - Transaction ID 0xd8287826
19 16.870834692   172.16.50.1           172.16.50.254
    ICMP     98     Echo (ping) request  id=0x2a5f, seq=3/768,
    ttl=64 (reply in 20)
20 16.871003010   172.16.50.254         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a5f, seq=3/768,
    ttl=64 (request in 19)
21 17.234354313   Cisco_78:94:83
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8003
22 17.894834253   172.16.50.1           172.16.50.254
    ICMP     98     Echo (ping) request  id=0x2a5f, seq=4/1024,
    ttl=64 (reply in 23)
23 17.894975611   172.16.50.254         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a5f, seq=4/1024,
    ttl=64 (request in 22)
24 18.918834302   172.16.50.1           172.16.50.254
    ICMP     98     Echo (ping) request  id=0x2a5f, seq=5/1280,
    ttl=64 (reply in 25)
25 18.918969026   172.16.50.254         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a5f, seq=5/1280,
    ttl=64 (request in 24)
26 19.235193666   Cisco_78:94:83
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8003
27 19.846798116   HewlettP_5a:7d:16     HewlettP_5a:7b:3f
    ARP      42     Who has 172.16.50.254? Tell 172.16.50.1
28 19.846922923   HewlettP_5a:7b:3f     HewlettP_5a:7d:16
    ARP      60     172.16.50.254 is at 00:21:5a:5a:7b:3f
29 19.942831907   172.16.50.1           172.16.50.254
    ICMP     98     Echo (ping) request  id=0x2a5f, seq=6/1536,
    ttl=64 (reply in 30)
30 19.942959297   172.16.50.254         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a5f, seq=6/1536,
    ttl=64 (request in 29)
31 20.015330791   Cisco_78:94:83        Cisco_78:94:83
    LOOP     60     Reply
32 20.051898080   HewlettP_5a:7b:3f     HewlettP_5a:7d:16
    ARP      60     Who has 172.16.50.1? Tell 172.16.50.254
33 20.051908766   HewlettP_5a:7d:16     HewlettP_5a:7b:3f
    ARP      42     172.16.50.1 is at 00:21:5a:5a:7d:16
34 20.966831607   172.16.50.1           172.16.50.254
    ICMP     98     Echo (ping) request  id=0x2a5f, seq=7/1792,
    ttl=64 (reply in 35)
35 20.966966889   172.16.50.254         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a5f, seq=7/1792,
    ttl=64 (request in 34)
36 21.239825260   Cisco_78:94:83
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8003
37 23.244909845   Cisco_78:94:83
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8003
38 25.249785396   Cisco_78:94:83
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8003
39 26.457580395   172.16.50.1           172.16.51.253
    ICMP     98     Echo (ping) request  id=0x2a69, seq=1/256,
    ttl=64 (reply in 40)
40 26.457750179   172.16.51.253         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a69, seq=1/256,
    ttl=64 (request in 39)
41 27.258588470   Cisco_78:94:83
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8003
42 27.462828828   172.16.50.1           172.16.51.253
    ICMP     98     Echo (ping) request  id=0x2a69, seq=2/512,
    ttl=64 (reply in 43)
43 27.462996517   172.16.51.253         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a69, seq=2/512,
    ttl=64 (request in 42)
44 28.486830414   172.16.50.1           172.16.51.253
    ICMP     98     Echo (ping) request  id=0x2a69, seq=3/768,
    ttl=64 (reply in 45)
45 28.486968211   172.16.51.253         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a69, seq=3/768,
    ttl=64 (request in 44)
46 29.259487258   Cisco_78:94:83
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8003
47 29.510839123   172.16.50.1           172.16.51.253
    ICMP     98     Echo (ping) request  id=0x2a69, seq=4/1024,
    ttl=64 (reply in 48)
48 29.510974895   172.16.51.253         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a69, seq=4/1024,
    ttl=64 (request in 47)
49 30.027072228   Cisco_78:94:83        Cisco_78:94:83
    LOOP     60     Reply
50 30.534832328   172.16.50.1           172.16.51.253
    ICMP     98     Echo (ping) request  id=0x2a69, seq=5/1280,
    ttl=64 (reply in 51)
51 30.534970404   172.16.51.253         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a69, seq=5/1280,
    ttl=64 (request in 50)
52 31.264393957   Cisco_78:94:83
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8003
53 31.558833076   172.16.50.1           172.16.51.253
    ICMP     98     Echo (ping) request  id=0x2a69, seq=6/1536,
    ttl=64 (reply in 54)
54 31.559002371   172.16.51.253         172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a69, seq=6/1536,
    ttl=64 (request in 53)
55 32.582831938   172.16.50.1           172.16.51.253
    ICMP     98     Echo (ping) request  id=0x2a69, seq=7/1792,
    ttl=64 (reply in 56)
56 32.583000674   172.16.51.253         172.16.50.1
```

```
    ICMP     98     Echo (ping) reply    id=0x2a69, seq=7/1792,
    ttl=64 (request in 55)
57 33.269318746   Cisco_78:94:83
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8003
58 35.274237039   Cisco_78:94:83
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8003
59 37.283296222   Cisco_78:94:83
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8003
60 38.433658289   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2a70, seq=1/256,
    ttl=64 (reply in 61)
61 38.433982981   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a70, seq=1/256,
    ttl=63 (request in 60)
62 39.284064965   Cisco_78:94:83
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8003
63 39.462833857   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2a70, seq=2/512,
    ttl=64 (reply in 64)
64 39.463110009   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a70, seq=2/512,
    ttl=63 (request in 63)
65 40.022472621   Cisco_78:94:83        Cisco_78:94:83
    LOOP     60     Reply
66 40.486827132   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2a70, seq=3/768,
    ttl=64 (reply in 67)
67 40.487074719   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a70, seq=3/768,
    ttl=63 (request in 66)
68 41.288969010   Cisco_78:94:83
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8003
69 41.510827670   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2a70, seq=4/1024,
    ttl=64 (reply in 70)
70 41.511072673   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a70, seq=4/1024,
    ttl=63 (request in 69)
71 42.534826252   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2a70, seq=5/1280,
    ttl=64 (reply in 72)
```

```
72 42.535115674   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a70, seq=5/1280,
    ttl=63 (request in 71)
73 43.293886675   Cisco_78:94:83
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8003
74 43.558828955   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2a70, seq=6/1536,
    ttl=64 (reply in 75)
75 43.559106714   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a70, seq=6/1536,
    ttl=63 (request in 74)
76 44.582828935   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2a70, seq=7/1792,
    ttl=64 (reply in 77)
77 44.583081271   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a70, seq=7/1792,
    ttl=63 (request in 76)
78 45.298869990   Cisco_78:94:83
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8003
79 45.606829124   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2a70, seq=8/2048,
    ttl=64 (reply in 80)
80 45.607073149   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2a70, seq=8/2048,
    ttl=63 (request in 79)
```

```
No Time           Source                Destination
    Protocol Length Info
24 17.387755766   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=1/256,
    ttl=63 (reply in 27)
25 17.387900200   HewlettP_5a:7e:51     Broadcast
    ARP      60     Who has 172.16.51.253? Tell 172.16.51.1
26 17.387924575   CameoCom_6f:b6:a5     HewlettP_5a:7e:51
    ARP      42     172.16.51.253 is at 00:40:f4:6f:b6:a5
27 17.388019422   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=1/256,
    ttl=64 (request in 24)
28 17.387584092   HewlettP_5a:7d:16     Broadcast
    ARP      60     Who has 172.16.50.254? Tell 172.16.50.1
29 17.387608537   HewlettP_5a:7b:3f     HewlettP_5a:7d:16
    ARP      42     172.16.50.254 is at 00:21:5a:5a:7b:3f
30 17.387742775   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=1/256,
    ttl=64 (reply in 31)
31 17.388032971   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=1/256,
    ttl=63 (request in 30)
32 18.044263131   Cisco_78:94:86
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/11/00:1e:bd:78:94:80  Cost = 0  Port = 0x8006
33 18.407785427   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=2/512,
    ttl=63 (reply in 34)
34 18.407902832   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=2/512,
    ttl=64 (request in 33)
35 18.407759515   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=2/512,
    ttl=64 (reply in 36)
36 18.407927138   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=2/512,
    ttl=63 (request in 35)
37 18.816182851   Cisco_78:94:84        CDP/VTP/DTP/PAgP/UDLD
    CDP      601    Device ID: gnu-sw1  Port ID: FastEthernet0/4
38 19.021443242   Cisco_78:94:84
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8004
39 19.431787182   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=3/768,
    ttl=63 (reply in 40)
40 19.431904379   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=3/768,
    ttl=64 (request in 39)
41 19.431760642   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=3/768,
    ttl=64 (reply in 42)
42 19.431934900   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=3/768,
    ttl=63 (request in 41)
43 20.049227679   Cisco_78:94:86
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/11/00:1e:bd:78:94:80  Cost = 0  Port = 0x8006
44 20.455774900   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=4/1024,
    ttl=63 (reply in 45)
45 20.455920871   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=4/1024,
    ttl=64 (request in 44)
46 20.455748988   172.16.50.1           172.16.51.1
```

```
    ICMP     98     Echo (ping) request  id=0x2afa, seq=4/1024,
    ttl=64 (reply in 47)
47 20.455946364   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=4/1024,
    ttl=63 (request in 46)
48 21.022555895   Cisco_78:94:84
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8004
49 21.466683941   Cisco_78:94:86        Cisco_78:94:86
    LOOP     60     Reply
50 21.479757169   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=5/1280,
    ttl=63 (reply in 51)
51 21.479870315   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=5/1280,
    ttl=64 (request in 50)
52 21.458643229   Cisco_78:94:84        Cisco_78:94:84
    LOOP     60     Reply
53 21.479738661   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=5/1280,
    ttl=64 (reply in 54)
54 21.479888893   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=5/1280,
    ttl=63 (request in 53)
55 22.054269054   Cisco_78:94:86
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/11/00:1e:bd:78:94:80  Cost = 0  Port = 0x8006
56 22.388728052   HewlettP_5a:7b:3f     HewlettP_5a:7d:16
    ARP      42     Who has 172.16.50.1? Tell 172.16.50.254
57 22.388864943   HewlettP_5a:7d:16     HewlettP_5a:7b:3f
    ARP      60     172.16.50.1 is at 00:21:5a:5a:7d:16
58 22.388722814   CameoCom_6f:b6:a5     HewlettP_5a:7e:51
    ARP      42     Who has 172.16.51.1? Tell 172.16.51.253
59 22.388825273   HewlettP_5a:7e:51     CameoCom_6f:b6:a5
    ARP      60     172.16.51.1 is at 00:21:5a:5a:7e:51
60 22.503784069   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=6/1536,
    ttl=63 (reply in 61)
61 22.503897912   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=6/1536,
    ttl=64 (request in 60)
62 22.503758017   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=6/1536,
    ttl=64 (reply in 63)
63 22.503922217   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=6/1536,
    ttl=63 (request in 62)
64 23.027381665   Cisco_78:94:84
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8004
65 23.527769132   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=7/1792,
    ttl=63 (reply in 66)
66 23.527885699   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=7/1792,
    ttl=64 (request in 65)
67 23.527743360   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=7/1792,
    ttl=64 (reply in 68)
68 23.527910982   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=7/1792,
    ttl=63 (request in 67)
69 24.059067585   Cisco_78:94:86
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/11/00:1e:bd:78:94:80  Cost = 0  Port = 0x8006
70 24.551767815   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=8/2048,
    ttl=63 (reply in 71)
71 24.551909665   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=8/2048,
    ttl=64 (request in 70)
72 24.551742881   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=8/2048,
    ttl=64 (reply in 73)
73 24.551934878   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=8/2048,
    ttl=63 (request in 72)
74 25.032073547   Cisco_78:94:84
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8004
75 25.575771107   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=9/2304,
    ttl=63 (reply in 76)
76 25.575895916   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=9/2304,
    ttl=64 (request in 75)
77 25.575745335   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=9/2304,
    ttl=64 (reply in 78)
78 25.575920570   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=9/2304,
    ttl=63 (request in 77)
```

```
79 26.068000387   Cisco_78:94:86
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/11/00:1e:bd:78:94:80  Cost = 0  Port = 0x8006
80 26.603791193   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=10/2560,
    ttl=63 (reply in 81)
81 26.603909157   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=10/2560,
    ttl=64 (request in 80)
82 26.603765142   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=10/2560,
    ttl=64 (reply in 83)
83 26.603934371   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=10/2560,
    ttl=63 (request in 82)
84 27.037000310   Cisco_78:94:84
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8004
85 27.623769660   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=11/2816,
    ttl=63 (reply in 86)
86 27.623882944   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=11/2816,
    ttl=64 (request in 85)
87 27.623745145   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=11/2816,
    ttl=64 (reply in 88)
88 27.623906970   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=11/2816,
    ttl=63 (request in 87)
89 28.068800353   Cisco_78:94:86
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/11/00:1e:bd:78:94:80  Cost = 0  Port = 0x8006
90 28.647769599   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=12/3072,
    ttl=63 (reply in 91)
91 28.647888262   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=12/3072,
    ttl=64 (request in 90)
92 28.647744735   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=12/3072,
    ttl=64 (reply in 93)
93 28.647913406   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=12/3072,
    ttl=63 (request in 92)
94 29.046271289   Cisco_78:94:84
```

```
    Spanning-tree-(for-bridges)_00 STP      60     Conf. Root =
    32768/10/00:1e:bd:78:94:80  Cost = 0  Port = 0x8004
95 29.671776244   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=13/3328,
    ttl=63 (reply in 96)
96 29.671919212   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=13/3328,
    ttl=64 (request in 95)
97 29.671751310   172.16.50.1           172.16.51.1
    ICMP     98     Echo (ping) request  id=0x2afa, seq=13/3328,
    ttl=64 (reply in 98)
98 29.671943517   172.16.51.1           172.16.50.1
    ICMP     98     Echo (ping) reply    id=0x2afa, seq=13/3328,
    ttl=63 (request in 97)
```