

**Faculdade de Engenharia da Universidade do Porto**



## **Rede de Computadores**

### **2º trabalho laboratorial**

**Redes de Computadores**

**Turma 4 Grupo 7**

Estudantes:

Alexandre Fernandes Lopes (up202207015)

Hugo Alexandre Almeida Barbosa (up202205774)

# Resumo

Este projeto foi desenvolvido no âmbito da unidade curricular de Redes de Computadores. Os principais acontecimentos foram a implementação de uma aplicação de download usando o protocolo **FTP** (File Transfer Protocol), bem como a configuração e análise de uma rede de computadores, em ambiente laboratorial. Durante o desenvolvimento, foram realizadas várias experiências para avaliar o comportamento da rede em diferentes cenários. Foi utilizada a ferramenta [Wireshark](#) para capturar e analisar pacotes de dados.

## Introdução

O objetivo deste projeto foi implementar e testar uma aplicação de download baseada no protocolo **FTP**, em conjunto com a configuração de uma rede de computadores, para realizar a transferência de ficheiros de forma eficiente e confiável em ambiente laboratorial. O projeto inclui também uma análise detalhada do comportamento da rede, com foco na identificação de possíveis melhorias no desempenho da transferência de dados. A ferramenta Wireshark foi utilizada para capturar pacotes e monitorizar o tráfego da rede durante as transferências, a fim de compreender o desempenho da rede e a eficiência da transferência de ficheiros.

O relatório está estruturado da seguinte forma:

- **Parte 1 - Aplicação de Download:** Descrição da arquitetura da aplicação de download implementada, incluindo os detalhes do protocolo FTP utilizado e a análise dos resultados obtidos com a captura dos pacotes no Wireshark.
- **Parte 2 - Configuração e Análise da Rede:**
  - Configuração de uma rede *IP*
  - Implementação de duas bridges num switch
  - Configuração de um router no Linux
  - Configuração de um router comercial e implementação de NAT
  - DNS
  - Conexões TCP
- **Conclusões:** Reflexão sobre os resultados alcançados e os objetivos de aprendizagem obtidos ao longo do projeto.

# Parte 1: Aplicação Download

## 1. Arquitetura da Aplicação

A aplicação desenvolvida tem como objetivo realizar o download de um ficheiro através do protocolo FTP. Para tal, a comunicação entre o cliente e o servidor ocorre através de uma arquitetura cliente-servidor, com o uso da porta padrão de controlo (21) do protocolo FTP. A aplicação começa com a análise do URL fornecido, que pode incluir o nome de utilizador, senha, host e caminho do ficheiro. A partir deste URL, a aplicação extrai as informações necessárias para estabelecer a ligação: nome de utilizador e senha (com valores padrão de "anonymous", caso não sejam especificados), o host e o caminho do ficheiro, além de resolver o nome do host para um endereço *IP*.

A primeira etapa do processo é a criação de um socket de controlo que se liga ao *IP* do servidor. Após a conexão ser estabelecida, a autenticação é realizada através do envio dos comandos **USER** e **PASS**, com o nome de utilizador e senha. Em seguida, a aplicação entra no modo passivo com o comando **PASV**, que permite ao servidor fornecer o endereço *IP* e a porta para a criação de uma segunda conexão de dados. Esta segunda conexão é utilizada para a transferência do ficheiro desejado, através do comando **RETR**, que solicita o ficheiro ao servidor. Os dados são então recebidos, e gravados localmente. Caso ocorra um problema durante a transferência, é utilizado o comando **ABOR** para abortar a mesma.

A comunicação com o servidor é feita de forma robusta, verificando as respostas do servidor após cada comando enviado, garantindo que o fluxo de comunicação seja mantido de forma eficiente e que qualquer erro, como falhas no login ou na transferência, seja devidamente tratado com mensagens de erro adequadas.

Após a transferência do ficheiro, a aplicação envia o comando **QUIT** para encerrar a sessão FTP de forma limpa, fechando as conexões de controlo e de dados. O código implementa funções específicas para enviar comandos FTP, receber respostas do servidor, estabelecer a conexão de controlo, passar para o modo passivo, e realizar a transferência de ficheiros, garantindo que cada parte do processo seja tratada corretamente.

## 2. Relatório de um download bem sucedido

Através do Wireshark, foi possível ver os *packets* de comunicação entre a aplicação e o servidor, sendo possível visualizar os diferentes comandos a serem enviados e as respostas obtidas. Na [captura 1](#) é possível ver o envio do utilizador, password, assim como a passagem para o modo passivo, e ainda o pedido para transferir o ficheiro. Na [captura 2](#) visualiza-se o sucesso da transferência.

## Parte 2: Análise e Configuração da rede

### Experiência 1 - Configuração de uma rede *IP*

Nesta experiência foram configurados os Tuxes 43 e 44 para estarem conectadas à mesma subrede *172.16.40.0/24*, no *switch*, pelas interfaces *eth1*. As interfaces foram configuradas com o comando *ifconfig*, para *IP 172.16.40.1* e **MAC** *00:01:02:9f:81:2e*, no Tux 43, e com *IP 172.16.40.254* e **MAC** *00:c0:df:02:55:95*, no Tux 44. Foi usado o comando *ping* para verificar que existe conexão entre as duas. Este comando gera pacotes **ICMP** (*Internet Control Message Protocol*), que contêm o endereço *IP* de origem e de destino, configurados previamente nos dispositivos. O endereço **MAC** de origem é o endereço da máquina que envia o *ping*. Antes de enviar o pacote **ICMP**, o dispositivo resolve o endereço **MAC** de destino. Se o endereço **MAC** não for conhecido, um **ARP Request** é enviado primeiro para resolvê-lo. O **ARP** (*Address Resolution Protocol*) é um protocolo utilizado para mapear endereços *IP* a endereços **MAC**, numa rede local. Os pacotes **ARP** são enviados para solicitar ou fornecer o endereço **MAC** correspondente a um endereço *IP*, permitindo a comunicação entre dispositivos na mesma rede. Antes de continuar a experiência, foram verificadas as rotas, e as tabelas **ARP**. Para testar como é feita a procura do *IP* no momento da conexão, foram apagadas as entradas dessas tabelas, e foi feito de novo o *ping* do Tux 43 para a 44, e foi capturada a interface *eth1* do Tux 43. Os comandos usados estão presentes no [Anexo 2.1](#), e a captura do Wireshark é mostrada na [Figura 3](#). É possível ver os pacotes **ARP** e os pacotes **ICMP** enviados e recebidos. Os pacotes **ARP** contêm tanto o endereço *IP* de origem como o de destino. Por exemplo, num **ARP Request**, o endereço *IP* de origem é o do dispositivo que faz a solicitação, enquanto o de destino é o *IP* que está a ser procurado. Na **ARP Reply**, o dispositivo de destino responde com o seu endereço **MAC**, permitindo que o *requester* associe o endereço *IP* ao endereço físico correspondente, sendo este um passo importante para proceder à conexão entre ambos.

### Experiência 2 - Implementação de duas bridges num switch

Nesta experiência, o objetivo foi criar duas redes locais, configurando duas bridges no switch: a *bridge40*, conectando os *Tux43* e *Tux44*, e a *bridge41*, conectando o *Tux42*.

Primeiro, foi configurado o *IP 172.16.41.1/24* no *Tux42*, assim como na experiência anterior. A seguir, o switch foi configurado através da consola no *Tux43*, criando as bridges 40 e 41, removendo as interfaces associadas ao bridge padrão e atribuindo-se às novas bridges. O *Tux42* e *Tux44* foram atribuídos à *bridge40*, enquanto o *Tux43* à *bridge41*.

Após a configuração, foi testado o comportamento das bridges. Quando o *Tux43* enviou um *ping broadcast* para *172.16.40.255*, o tráfego foi visível apenas dentro da *bridge40*, não alcançando o *Tux42*. O mesmo comportamento foi observado ao realizar um *ping broadcast* no *Tux42* para *172.16.41.255*, que foi restrito à *bridge41*.

A análise dos logs capturados confirmou que os dispositivos em bridges diferentes não comunicavam diretamente, evidenciando o isolamento entre as duas sub-redes e a criação de dois domínios de *broadcast* diferentes.

Os logs da captura dos pacotes durante o comando *ping broadcast* nos Tuxes 43 e 42 estão presentes [Figura 4](#) e [Figura 5](#), respetivamente, e os comandos utilizados no [Anexo 2.2](#).

## Experiência 3 - Configuração de um router no Linux

Nesta experiência, o objetivo foi configurar o *Tux44* como um router entre as redes locais representadas pelas bridges *bridge40* e *bridge41*. O *Tux43* estava conectado à sub-rede *172.16.40.0/24* e o *Tux42* à sub-rede *172.16.41.0/24*, e o *Tux44* foi configurado para permitir a comunicação entre essas duas sub-redes. Para isso, foi necessário ativar o encaminhamento de pacotes no *Tux44*, configurar a interface *eth2* no *Tux44* com o *IP 172.16.41.253/24*, desativar o *icmp\_echo\_ignore\_broadcasts* e ajustar as rotas nos *Tux42* e *Tux43* de forma a que ambos usassem o *Tux44* como *gateway*.

Depois da configuração, realizamos o comando *ping* entre o *Tux43* e *Tux42* e evidenciamos que ambos estavam a usar o *Tux44* como *gateway* para se comunicarem. As rotas configuradas nos *Tux42* e *Tux43* foram verificadas e mostraram que o tráfego era encaminhado pelo *Tux44*, que redirecionava os pacotes para as sub-redes corretas. Cada entrada na tabela de encaminhamento indicava o endereço de destino, o *gateway*, a máscara de sub-rede e a interface através da qual os pacotes seriam enviados.

Foram observados pacotes **ARP** trocados entre os *Tuxes* 43 e 44, nos quais o *Tux43* enviava **ARP** requests para descobrir o endereço **MAC** do *Tux44*, uma vez que só conhecia o endereço *IP* do gateway. Também foram capturados pacotes *ICMP Echo Request* e *Echo Reply*, confirmando que a rede estava bem configurada e que as máquinas se comunicavam entre redes. Os pacotes **ICMP** continham os endereços *IP* e **MAC** das interfaces de origem e destino, garantindo que os pacotes fossem direcionados corretamente entre as redes.

Os logs da captura dos pacotes durante o comando *ping* no *Tux43* está presente na [Figura 6](#) e os comandos utilizados estão presentes no [anexo2.3](#).

## Experiência 4 - Configuração de um router comercial e implementação de NAT

Nesta experiência, o objetivo foi configurar um router comercial (RC) e implementar a tradução de endereços de rede (NAT). Inicialmente, o router RC foi resetado e configurado com os *IPs 172.16.1.41/24* e *172.16.41.254/24*, nas interfaces *ether1* e *ether2*, respetivamente. Para configurar uma rota estática num router comercial, foi necessário resetar as configurações do dispositivo, atribuir os *IPs* internos e externos e adicionar as rotas apropriadas através do terminal do router.

Após a configuração do router, as rotas nos *tuxes* foram ajustadas para garantir a conectividade entre as redes. O *Tux42* foi configurado para usar o *gateway 172.16.41.254*, o *Tux43* para o *gateway 172.16.40.254* e o *Tux44* para o *gateway 172.16.41.254*. Com as rotas configuradas, foram feitos testes de *ping* entre os dispositivos, que confirmaram que a comunicação estava a funcionar corretamente. Durante os testes, foi verificado que, sem a ligação do *Tux42* ao *Tux44*, os pacotes de dados eram redirecionados (*ICMP Redirect*) para

o router, que os encaminhava para o endereço *IP* de destino. Quando o *Tux42* foi reconectado e as rotas foram ajustadas para usar o RC como *gateway*, o tráfego seguiu um caminho mais eficiente, com o *ICMP Redirect* a garantir que os pacotes chegassem ao destino pelo caminho mais eficiente.

Em seguida, foi feito um teste de *ping* ao servidor FTP (*172.16.1.10*). Com o NAT ativado no router RC, o pacote **ICMP** enviado do *Tux43* foi traduzido corretamente para um *IP* público, e a comunicação com o servidor FTP foi bem-sucedida. No entanto, quando o NAT foi desativado no router, a comunicação falhou. Sem o NAT, o pacote não foi traduzido para um *IP* público ao sair para a Internet, e o servidor FTP não soube como encaminhar a resposta de volta ao *Tux43*. Ao reativar o NAT com o comando "*/ip firewall nat enable 0*", a comunicação foi restaurada, permitindo que o pacote fosse traduzido corretamente, o que demonstra a importância do NAT para a comunicação com a Internet. O NAT foi configurado no router RC para traduzir os endereços *IP* privados para um único *IP* público, o que permite que múltiplos dispositivos na rede interna compartilhem o mesmo *IP* ao acessar a Internet.

Os logs da captura dos pacotes durante o comando *ping* no *Tux43*, que demonstram o sucesso da comunicação com o servidor FTP com o NAT ativado, estão presentes na [Figura 20](#). Já os logs que mostram a falha na comunicação com o NAT desativado estão presentes na [Figura 21](#). Os logs do *traceroute*, que mostram o caminho mais longo e mais curto, estão na [Figura 22](#). Os comandos utilizados estão no [anexo 2.4](#).

## Experiência 5 - DNS

O foco desta experiência foi a configuração do **DNS** nos Tuxes 42, 43 e 44, e testar o seu funcionamento. Foi adicionada a linha "*nameserver 10.227.20.3*" ao arquivo */etc/resolv.conf* dos três Tuxes. Este arquivo define o servidor **DNS** a ser utilizado para resolver nomes de domínio. O **DNS** (*Domain Name System*) é um sistema de nomes de domínio que funciona como uma espécie de "lista telefônica". O sistema traduz nomes de domínio para endereços *IP*, que são utilizados pelos computadores para se localizarem e comunicarem com os servidores. Foi possível testar o correto funcionamento do DNS fazendo *ping* ao servidor [google.com](http://google.com), e em seguida, ver os pacotes **DNS** a serem trocados, fazendo uma captura com o Wireshark. Este teste foi realizado nas três Tuxes, correndo os comandos no [Anexo 2.5](#), estando o resultado das capturas nas Figuras 7, 8, e 9. Os pacotes trocados pelo **DNS** são principalmente consultas (*queries*) e respostas (*replies*). Uma consulta **DNS** inclui o nome de domínio que está a ser solicitado, e a resposta inclui o endereço *IP* correspondente ao nome de domínio, ou um erro, se o domínio não for encontrado. É possível observar que nas capturas estão presentes os pacotes **DNS**, onde se pode identificar os pacotes de consulta, seguidos das respostas com a tradução em *IP*, verificando-se que o **DNS** funciona corretamente.

## Experiência 6 - Conexões TCP

Esta experiência teve como objetivo utilizar a aplicação desenvolvida para transferir um arquivo presente no servidor *netlab*, e verificar os pacotes trocados com o mesmo. Além

disso, foi também testada a transferência mútua do mesmo ficheiro, em dois Tuxes diferentes (Tux 43 e Tux 42).

Para executar o programa utilizou-se os comandos presentes no [Anexo 2.6](#). Na primeira transferência, a captura foi feita ao Tux 43, com o Wireshark, e está presente nas Figuras 1 e 2. Já o segundo momento de transferência está relatado nas Figuras 15, 16 e 17. É possível observar na transferência mútua, que a partir de certo ponto, existe um congestionamento da ligação, havendo uma intervenção por parte do **TCP** (*Transmission Control Protocol*) para conseguir gerir as duas transferências. O **FTP** (*File Transfer Protocol*) usa duas conexões **TCP**: uma para controlo e outra para a transferência de dados. A informação de controlo do **FTP** é transportada pela conexão **TCP** estabelecida. Essa conexão gerencia a comunicação entre o cliente e o servidor para controlo de comandos e respostas. As fases de uma conexão **TCP** incluem o estabelecimento, através do processo de *3-way handshake* (**SYN**, **SYN-ACK**, **ACK**), a transferência de dados, que após o estabelecimento troca os dados entre o cliente e o servidor, e o término, onde a conexão é encerrada com o processo de *4-way handshake* (**FIN**, **ACK**, **FIN**, **ACK**). O mecanismo **ARQ** (*Automatic Repeat request*) garante a entrega confiável dos pacotes. Ele utiliza números de sequência e confirmações (**ACKs**). Numa rede congestionada, onde pacotes podem ser perdidos, o **TCP** usa os mecanismos *Additive Increase* e *Multiplicative Decrease* para ajustar a janela de congestionamento. O **Timeout**, que quando um pacote não é confirmado, a janela de congestionamento é reduzida drasticamente (*Multiplicative Decrease*) e depois aumentada gradualmente (*Slow Start*) até o valor inicial. Os **3 ACKs seguidos**, em que se o **TCP** recebe três confirmações duplicadas, ele reduz a janela de congestionamento (*Multiplicative Decrease*) e começa a aumentar a janela de congestionamento de forma incremental (*Additive Increase*). Nas capturas, pode-se observar os números de sequência e os **ACKs** para cada pacote transmitido, bem como retransmissões caso algum pacote seja perdido. O **TCP** ajusta a **janela de congestionamento** (*congestion window*) com base no nível de congestionamento da rede. O mecanismo funciona da seguinte forma: Se o congestionamento aumenta, a janela diminui (*Multiplicative Decrease*), e se o congestionamento diminui, a janela aumenta (*Additive Increase*). O **throughput** pode aumentar à medida que a janela de congestionamento cresce, mas, quando o congestionamento é detectado, o **throughput** diminui, com ajustes automáticos na janela. A evolução do **throughput** reflete esse comportamento, com variações conforme o congestionamento da rede. Ao criar mais do que uma conexão **TCP**, a largura de banda disponível é dividida entre as conexões, o que pode reduzir a velocidade de cada uma. O **TCP** ajusta automaticamente a janela de congestionamento com base no tráfego, o que pode levar a uma diminuição no **throughput** da conexão original devido à competição pelos recursos da rede.

O **throughput** e o gráfico de I/O podem ser vistos nas Figuras 11 e 14, para a 1ª transferência, e nas Figuras 18 e 19, respectivamente. Verifica-se que no gráfico da Figura 18, o **throughput** duplica, pelo que isso ocorre no momento em que a 2ª transferência se inicia.

## Conclusões

Este projeto permitiu-nos consolidar o entendimento sobre o protocolo FTP e outros protocolos utilizados na transferência de dados em redes de computadores, com especial foco na propagação desses dados na *network layer* e na sua transição para a *link layer*,



cumprindo todos os objetivos propostos.

## Referências

1. POSTEL, J.; REYNOLDS, J. *File Transfer Protocol (FTP)*. RFC 959. ISI, outubro de 1985. Disponível em: <https://datatracker.ietf.org/doc/html/rfc959>.

## Anexos (Código da aplicação, comandos da configuração, logs)

### Anexo 1 - Código da Aplicação

```
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <netdb.h>
#include <regex.h>

#define SERVER_PORT 21 //FTP standard control port
#define MAX_LENGTH 256
#define MAX_PATH_LENGTH 1024
#define IP_MAX_LENGTH 32

//server responses
#define SERVICE_READY_USER 220
#define USER_OK_NEED_PASS 331
#define USER_LOGGED_IN 230
#define ENTER_PASSIVE_MODE 227
#define FILE_OK_OPEN_DATA 150
#define CLOSING_DATA_CONNECTION 226
#define CLOSING_CONTROL_CONNECTION 221
#define DATA_CONNECTION_ALREADY_OPEN 125
#define ERROR_CLOSING_DATA_CONNECTION 426

typedef struct {
```



```

    char user[MAX_LENGTH];
    char password[MAX_LENGTH];
    char host[MAX_LENGTH];
    char path[MAX_PATH_LENGTH];
    char ip[IP_MAX_LENGTH];
} URL;

int parse_url(char* url, URL *parsed_url) {
    regex_t regex;

    regcomp(&regex, "^ftp://", 0);
    if (regexec(&regex, url, 0, NULL, 0)) {
        printf("Invalid URL format: must start with ftp://\n");
        return -1;
    }

    regcomp(&regex, "@", 0);
    if (regexec(&regex, url, 0, NULL, 0) != 0) { // Format:
ftp://<host>/<path>
        sscanf(url, "ftp://%255[^/]/%1023s", parsed_url->host,
parsed_url->path);
        strcpy(parsed_url->user, "anonymous");
        strcpy(parsed_url->password, "anonymous");
    } else { // Format: ftp://<user>:<password>@<host>/<path>
        sscanf(url, "ftp://%255[^:]:%255[^@]@%255[^/]/%1023s",
parsed_url->user, parsed_url->password, parsed_url->host,
parsed_url->path);
    }

    // Resolve hostname to IP
    struct hostent *he = gethostbyname(parsed_url->host);
    if (he == NULL) {
        printf("Could not resolve hostname %s\n",
parsed_url->host);
        return -1;
    }

    struct in_addr **addr_list = (struct in_addr
**)he->h_addr_list;
    strcpy(parsed_url->ip, inet_ntoa(*addr_list[0])); // Store
the first IP address

```

```

    return 0;
}

int openSocket(char* ip, int port) {

    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(ip);    /*32 bit
Internet address network byte ordered*/
    server_addr.sin_port = htons(port);            /*server TCP port
must be network byte ordered */

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

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

    return sockfd;
}

int receive_server_response(int sockfd, char *response) {
    int responseCode = 0, n;
    while (responseCode <= 0) {
        n = recv(sockfd, response, MAX_LENGTH - 1, 0);
        if (n < 0) {
            perror("recv()");
            exit(-1);
        }
        response[n] = '\0';
    }
}

```

```

        sscanf(response, "%d", &responseCode);
    }
    printf("Server: %s\n", response);
    printf("Received: Code %d\n", responseCode);
    return responseCode;
}

// Function to send and receive FTP commands
void send_ftp_command(int sockfd, const char *command) {
    char buffer[MAX_LENGTH];
    send(sockfd, command, strlen(command), 0);
    printf("Sent: %s\n", command);
}

int send_user_command(int sockfd, const char *user) {
    char response[MAX_LENGTH];
    char command[MAX_LENGTH];
    snprintf(command, sizeof(command), "USER %s\r\n", user);
    send_ftp_command(sockfd, command);
    int code = 0;
    do {
        code = receive_server_response(sockfd, response);
    } while (code == SERVICE_READY_USER);

    return code;
}

int send_pass_command(int sockfd, const char *password) {
    char response[MAX_LENGTH];
    char command[MAX_LENGTH];
    snprintf(command, sizeof(command), "PASS %s\r\n", password);
    send_ftp_command(sockfd, command);
    int code = 0;
    do {
        code = receive_server_response(sockfd, response);
    } while (code == USER_OK_NEED_PASS);

    return code;
}

void passive_mode(int sockfd, char *data_ip, int *data_port) {

```

```

char response[MAX_LENGTH];
int h1, h2, h3, h4, p1, p2;

send_ftp_command(sockfd, "PASV\r\n");

int code = 0;
do {
    code = receive_server_response(sockfd, response);
} while (code == USER_LOGGED_IN);

if (code != ENTER_PASSIVE_MODE) {
    printf("Failed to enter Passive mode. Aborting\n");
    exit(-1);
}

sscanf(response, "%*[^ (] (%d,%d,%d,%d,%d,%d)%*[^\\n$)]", &h1,
&h2, &h3, &h4, &p1, &p2);
printf("p1 = %d p2 = %d\n", p1, p2);
// Construct the IP address
snprintf(data_ip, IP_MAX_LENGTH, "%d.%d.%d.%d", h1, h2, h3,
h4);

// Calculate the data port
*data_port = (p1 * 256) + p2;

printf("Data Connection Info - IP: %s, Port: %d\n", data_ip,
*data_port);
}

int retrieve_file(int control_sock, int data_sock, const char
*file_path) {
    char command[MAX_LENGTH];
    char buffer[MAX_LENGTH];
    char response[MAX_LENGTH];
    FILE *file;

    // Send RETR command
    snprintf(command, sizeof(command), "RETR %s\r\n", file_path);
    send_ftp_command(control_sock, command);
    int code = 0;
    do {

```

```

        code = receive_server_response(control_sock, response);
    } while (code == ENTER_PASSIVE_MODE);

    if (code != FILE_OK_OPEN_DATA && code !=
DATA_CONNECTION_ALREADY_OPEN) {
        printf("Could not find resource %s\n", file_path);
        close(control_sock);
        close(data_sock);
        exit(-1);
    }

    // Open file for writing
    file = fopen("downloaded_file", "wb");
    if (!file) {
        perror("fopen()");
        exit(-1);
    }

    // Receive data from data socket
    int bytes_received;
    while ((bytes_received = recv(data_sock, buffer,
sizeof(buffer), 0)) > 0) {
        if (fwrite(buffer, 1, bytes_received, file) < 0) {
            printf("Error downloading file\n");
            fclose(file);
            close(control_sock);
            close(data_sock);
            exit(-1);
        }
    }

    if (bytes_received < 0) {
        perror("recv()");
        fclose(file);
        close(control_sock);
        close(data_sock);
        exit(-1);
    }

    fclose(file);
    do {

```

```

        code = receive_server_response(control_sock, response);
    } while (code == FILE_OK_OPEN_DATA || code ==
DATA_CONNECTION_ALREADY_OPEN);

    if (code == ERROR_CLOSING_DATA_CONNECTION) {

        printf("Error while closing connection. Handling.\n");

        memset(command, 0, sizeof(command));
        snprintf(command, sizeof(command), "ABOR\r\n");
        send_ftp_command(control_sock, command);
        do {
            code = receive_server_response(control_sock, response);
        } while (code == ERROR_CLOSING_DATA_CONNECTION);
    }

    return code;
}

int open_data_socket(const char *data_ip, int data_port) {
    return openSocket((char *)data_ip, data_port);
}

int closeSockets(int control_socket, int data_socket) {
    char response[MAX_LENGTH];
    char command[MAX_LENGTH];
    snprintf(command, sizeof(command), "QUIT\r\n");
    send_ftp_command(control_socket, command);
    int code = 0;
    do {
        code = receive_server_response(control_socket, response);
    } while (code == CLOSING_DATA_CONNECTION);

    if (code != CLOSING_CONTROL_CONNECTION) return -1;

    if (close(control_socket) < 0 || close(data_socket) < 0)
return -1;

    return 1;
}

```

```
int main(int argc, char *argv[]) {
    if (argc != 2) {
        printf("Usage: ./download
ftp://[<user>:<password>@]<host>/<url-path>\n");
        exit(-1);
    }

    URL parsed_url;
    int result = parse_url(argv[1], &parsed_url);
    char response[MAX_LENGTH];

    if (result != 0) {
        printf("Failed to parse URL\n");
        exit(-1);
    }

    printf("User: %s\n", parsed_url.user);
    printf("Password: %s\n", parsed_url.password);
    printf("Host: %s\n", parsed_url.host);
    printf("Path: %s\n", parsed_url.path);
    printf("IP: %s\n", parsed_url.ip);

    // Open control socket
    int control_sock = openSocket(parsed_url.ip, SERVER_PORT);
    if (control_sock < 0 || receive_server_response(control_sock,
response) != SERVICE_READY_USER) {
        printf("Failed to connect to %s\n", parsed_url.ip);
        exit(-1);
    }
    printf("Connected to server at %s\n", parsed_url.ip);

    printf("Server ready to receive new User\n");

    // Send USER and PASS commands
    if (send_user_command(control_sock, parsed_url.user) !=
USER_OK_NEED_PASS) {
        printf("User %s is not known. Aborting\n",
parsed_url.user);
        close(control_sock);
        exit(-1);
    }
}
```



```

    }

    printf("User %s is known. Asking for Password\n",
parsed_url.user);

    if (send_pass_command(control_sock, parsed_url.password) !=
USER_LOGGED_IN) {
        printf("Could not log the User %s with password:%s in.
Aborting\n", parsed_url.user, parsed_url.password);
        close(control_sock);
        exit(-1);
    }

    printf("Logged the User %s with password:%s in.\nProceeding to
enter Passive Mode\n", parsed_url.user, parsed_url.password);

    // Enter passive mode
    char data_ip[IP_MAX_LENGTH];
    int data_port;
    passive_mode(control_sock, data_ip, &data_port);

    // Open data socket
    int data_sock = open_data_socket(data_ip, data_port);
    if (data_sock < 0) {
        printf("Failed to open data socket\n");
        exit(-1);
    }

    // Retrieve file
    if (retrieve_file(control_sock, data_sock, parsed_url.path) !=
CLOSING_DATA_CONNECTION) {
        printf("Error downloading file %s.\n", parsed_url.path);
        close(control_sock);
        close(data_sock);
        exit(-1);
    }

    printf("File %s was downloaded successfully.\n",
parsed_url.path);

    // Close sockets

```

```
    if (closeSockets(control_sock, data_sock) < 0) {  
        printf("Error closing sockets\n");  
        exit(-1);  
    }  
  
    printf("Connection closed successfully.\n");  
  
    printf("Ending program\n");  
    return 0;  
}
```

## Anexo 2 - Comandos da Configuração

### 2.1 Experiência 1

Tux43:

ifconfig eth1 up

ifconfig eth1 172.16.40.1/24

Tux44:

ifconfig eth1 up

ifconfig eth1 172.16.40.254/24

Tux43:

ping 172.16.40.254 (responde)

Tux44:

ping 172.16.40.1 (responde)

Tux43:

arp -a (? (172.16.40.254) at 00:c0:df:02:55:95 [ether] on eth1

? (10.227.20.254) at e4:8d:8c:20:25:c8 [ether] on eth0)

arp -d 172.16.40.254/24

arp -a (? (10.227.20.254) at e4:8d:8c:20:25:c8 [ether] on eth0)

ping 172.16.40.254 (responde)

### 2.2 Experiência 2

Tux42:

```
ifconfig eth1 up
ifconfig eth1 172.16.41.1/24
```

Switch console:

```
/system reset-configuration
/interface bridge add name=bridge40
/interface bridge add name=bridge41
/interface bridge port remove [find interface=ether2]
/interface bridge port remove [find interface=ether3]
/interface bridge port remove [find interface=ether4]
/interface bridge port add bridge=bridge41 interface=ether2
/interface bridge port add bridge=bridge40 interface=ether3
/interface bridge port add bridge=bridge40 interface=ether4
```

Tux43:

```
ping 172.16.40.254 (responde)
ping 172.16.41.1 (não responde)
ping -b 172.16.40.255 (broadcast)
```

Tux42:

```
ping -b 172.16.41.255 (broadcast)
```

## 2.3 Experiência 3

Switch console:

```
/interface bridge port remove [find interface=ether7]
/interface bridge port add bridge=bridge41 interface=ether7
```

Tux44:

```
ifconfig eth2 up
ifconfig eth2 172.16.41.253/24
sysctl net.ipv4.ip_forward=1
sysctl net.ipv4.icmp_echo_ignore_broadcasts=0
```

Tux42:

```
route add -net 172.16.40.0/24 gw 172.16.41.253
```

Tux43:

```
route add -net 172.16.41.0/24 gw 172.16.40.254
```

Tux43:

```
ping 172.16.40.254 (responde)
```

ping 172.16.41.253 (responde)  
ping 172.16.41.1 (responde)

Tux42:  
arp -d 172.16.41.253

Tux43:  
arp -d 172.16.40.254

Tux44:  
arp -d 172.16.40.1  
arp -d 172.16.41.1

Tux43:  
ping 172.16.41.1 (responde)

## 2.4 Experiência 4

Switch console:  
/interface bridge port remove [find interface=ether6]  
/interface bridge port add bridge=bridge41 interface=ether6

Router console:  
/system reset-configuration  
/ip address add address=172.16.1.41/24 interface=ether1  
/ip address add address=172.16.41.254/24 interface=ether2  
/ip route add dst-address=172.16.40.0/24 gateway=172.16.41.253

Tux42:  
route add -net 172.16.1.0/24 gw 172.16.41.254

Tux43:  
route add -net 172.16.1.0/24 gw 172.16.40.254

Tux44:  
route add -net 172.16.1.0/24 gw 172.16.41.254

Tux43:  
ping 172.16.41.1 (responde)  
ping 172.16.40.254 (responde)  
ping 172.16.41.254 (responde)

Tux 42:

```
sysctl net.ipv4.conf.eth1.accept_redirects=0
sysctl net.ipv4.conf.all.accept_redirects=0
route del -net 172.16.40.0 gw 172.16.41.253 netmask 255.255.255.0
ping 172.16.40.1 (responde)
traceroute -n 172.16.40.1
route add -net 172.16.40.0/24 gw 172.16.41.253
traceroute -n 172.16.50.1
sysctl net.ipv4.conf.eth1.accept_redirects=1
sysctl net.ipv4.conf.all.accept_redirects=1
```

Tux43:

```
ping 172.16.1.10 (responde)
```

Router console:

```
/ip firewall nat disable 0
```

Tux43:

```
ping 172.16.1.10 (não responde)
```

Router console:

```
/ip firewall nat enable 0
```

## 2.5 Experiência 5

Tux42, Tux43, Tux44:

```
ping google.com (responde, depois de configurarmos o DNS)
```

## 2.6 Experiência 6

Tux43:

```
gcc -o download project.c
./download ftp://rcom:rcom@ftp.netlab.fe.up.pt/README
```

Tux43:

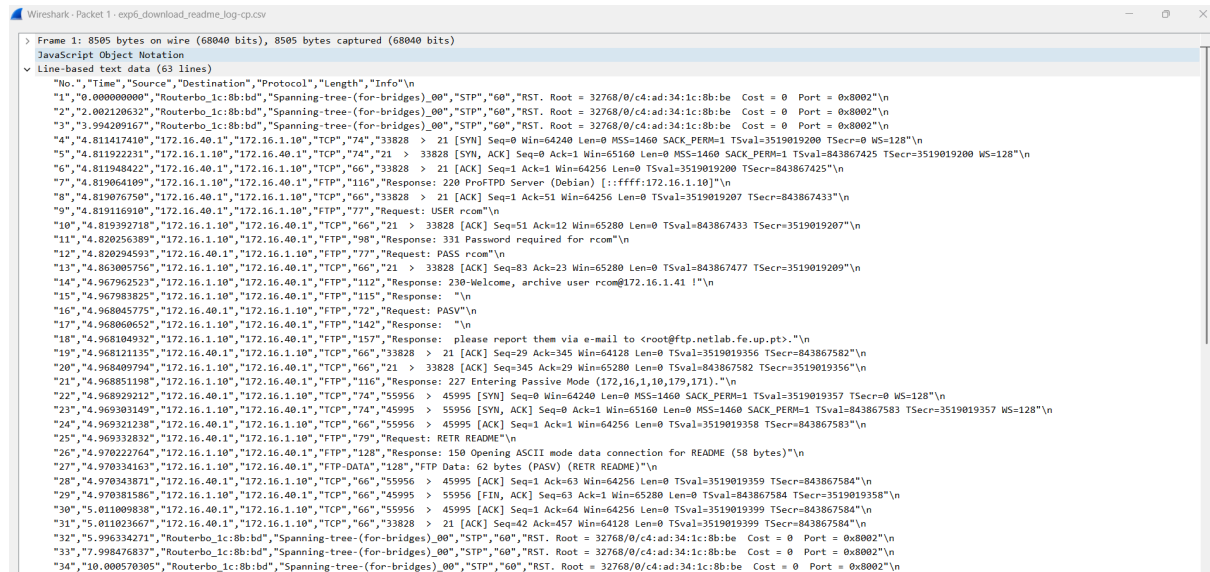
```
./download ftp://rcom:rcom@ftp.netlab.fe.up.pt/ubuntu.iso
```

Tux42:

gcc -o download project.c

./download ftp://rcom:rcom@ftp.netlab.fe.up.pt/ubuntu.iso

## Anexo 3 - PrintScreens de logs obtidos no Wireshark



```
> Frame 1: 8595 bytes on wire (68040 bits), 8595 bytes captured (68040 bits) on interface 0
JavaScript Object Notation
Line-based text data (63 lines)
No.,Time,Source,Destination,Protocol,Length,Info
1,"0.000000000","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"
2,"2.002120632","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"
3,"3.994209167","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"
4,"4.811417410","172.16.40.1","172.16.1.10","TCP","74","33828 > 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3519019200 TSecr=0 WS=128"
5,"4.811922231","172.16.1.10","172.16.40.1","TCP","74","21 > 33828 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=843867425 TSecr=3519019200 WS=128"
6,"4.811948422","172.16.40.1","172.16.1.10","TCP","66","33828 > 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3519019200 TSecr=843867425"
7,"4.819064109","172.16.1.10","172.16.40.1","FTP","116","Response: 220 ProFTPD Server (Debian) [::ffff:172.16.1.10]"
8,"4.819076750","172.16.40.1","172.16.1.10","TCP","66","33828 > 21 [ACK] Seq=1 Ack=51 Win=64256 Len=0 TSval=3519019207 TSecr=843867433"
9,"4.819116910","172.16.40.1","172.16.1.10","FTP","77","Request: USER rcom"
10,"4.819392718","172.16.1.10","172.16.40.1","TCP","66","21 > 33828 [ACK] Seq=51 Ack=12 Win=65280 Len=0 TSval=843867433 TSecr=3519019207"
11,"4.820256389","172.16.1.10","172.16.40.1","FTP","98","Response: 331 Password required for rcom"
12,"4.820294593","172.16.40.1","172.16.1.10","FTP","77","Request: PASS rcom"
13,"4.863905756","172.16.1.10","172.16.40.1","TCP","66","21 > 33828 [ACK] Seq=83 Ack=23 Win=65280 Len=0 TSval=843867477 TSecr=3519019209"
14,"4.867962523","172.16.1.10","172.16.40.1","FTP","112","Response: 230 Welcome, archive user rcom@172.16.1.41 !"
15,"4.867983829","172.16.1.10","172.16.40.1","FTP","115","Response: "
16,"4.968045775","172.16.40.1","172.16.1.10","FTP","72","Request: PASV"
17,"4.96806652","172.16.1.10","172.16.40.1","FTP","142","Response: "
18,"4.968104932","172.16.1.10","172.16.40.1","FTP","157","Response: please report them via e-mail to <root@ftp.netlab.fe.up.pt>."
19,"4.968121135","172.16.40.1","172.16.1.10","TCP","66","33828 > 21 [ACK] Seq=29 Ack=345 Win=64128 Len=0 TSval=3519019356 TSecr=843867582"
20,"4.968409794","172.16.1.10","172.16.40.1","TCP","66","21 > 33828 [ACK] Seq=345 Ack=29 Win=65280 Len=0 TSval=843867582 TSecr=3519019356"
21,"4.968851198","172.16.1.10","172.16.40.1","FTP","116","Response: 227 Entering Passive Mode (172,16,1,10,179,171)."
22,"4.968929212","172.16.40.1","172.16.1.10","TCP","74","55956 > 45995 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3519019357 TSecr=0 WS=128"
23,"4.969303149","172.16.1.10","172.16.40.1","TCP","74","45995 > 55956 [FIN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=843867583 TSecr=3519019357 WS=128"
24,"4.969321230","172.16.40.1","172.16.1.10","TCP","66","55956 > 45995 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3519019358 TSecr=843867583"
25,"4.969332832","172.16.40.1","172.16.1.10","FTP","79","Request: RETR README"
26,"4.970222764","172.16.1.10","172.16.40.1","FTP","128","Response: 150 Opening ASCII mode data connection for README (58 bytes)"
27,"4.970334163","172.16.1.10","172.16.40.1","FTP-DATA","62","FTP Data: 62 bytes (PASV) (RETR README)"
28,"4.970343871","172.16.40.1","172.16.1.10","TCP","66","55956 > 45995 [ACK] Seq=1 Ack=63 Win=64256 Len=0 TSval=3519019359 TSecr=843867584"
29,"4.970381586","172.16.1.10","172.16.40.1","TCP","66","45995 > 55956 [FIN, ACK] Seq=63 Ack=1 Win=65280 Len=0 TSval=843867584 TSecr=3519019358"
30,"5.011009838","172.16.40.1","172.16.1.10","TCP","66","55956 > 45995 [ACK] Seq=1 Ack=64 Win=64256 Len=0 TSval=3519019399 TSecr=843867584"
31,"5.011023667","172.16.40.1","172.16.1.10","TCP","66","33828 > 21 [ACK] Seq=42 Ack=457 Win=64128 Len=0 TSval=3519019399 TSecr=843867584"
32,"5.996334271","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"
33,"7.998476837","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"
34,"10.000570305","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"
```

Fig 1 - *Tux43* Conecta ao servidor *netlab* e pede o ficheiro README

```
,,"4.970334163","172.16.1.10","172.16.40.1","FTP-DATA","128","FTP Data: 62 bytes (PASV) (RETR README)"
,"4.970343871","172.16.40.1","172.16.1.10","TCP","66","55956 > 45995 [ACK] Seq=1 Ack=63 Win=64256 Len=0 TSval=3519019359 TSecr=843867584"
,"4.970381586","172.16.1.10","172.16.40.1","TCP","66","45995 > 55956 [FIN, ACK] Seq=63 Ack=1 Win=65280 Len=0 TSval=843867584 TSecr=3519019358"
,"5.011009838","172.16.40.1","172.16.1.10","TCP","66","55956 > 45995 [ACK] Seq=1 Ack=64 Win=64256 Len=0 TSval=3519019399 TSecr=843867584"
,"5.011023667","172.16.40.1","172.16.1.10","TCP","66","33828 > 21 [ACK] Seq=42 Ack=457 Win=64128 Len=0 TSval=3519019399 TSecr=843867584"
,"5.996334271","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"
,"7.998476837","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"
,"10.000570305","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"
,"10.042630886","Kye_02:55:95","3Com_9f:81:2e","ARP","60","Who has 172.16.40.1? Tell 172.16.40.254"
,"10.042643318","3Com_9f:81:2e","Kye_02:55:95","ARP","42","172.16.40.1 is at 00:01:02:9f:81:2e"
,"12.002699115","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"
,"13.994693300","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"
,"15.772518662","172.16.1.10","172.16.40.1","FTP","89","Response: 226 Transfer complete"
,"15.772558402","172.16.40.1","172.16.1.10","TCP","66","33828 > 21 [ACK] Seq=42 Ack=480 Win=64128 Len=0 TSval=3519030161 TSecr=843878387"
,"15.772612740","172.16.40.1","172.16.1.10","FTP","72","Request: QUIT"
,"15.773249563","172.16.1.10","172.16.40.1","FTP","80","Response: 221 Goodbye."
,"15.773293284","172.16.40.1","172.16.1.10","TCP","66","33828 > 21 [FIN, ACK] Seq=48 Ack=494 Win=64128 Len=0 TSval=3519030162 TSecr=843878387"
,"15.773309907","172.16.40.1","172.16.1.10","TCP","66","55956 > 45995 [FIN, ACK] Seq=1 Ack=64 Win=64256 Len=0 TSval=3519030162 TSecr=843867584"
,"15.773312351","172.16.40.1","172.16.1.10","TCP","66","[TCP Out-Of-Order] 55956 > 45995 [FIN, ACK] Seq=1 Ack=64 Win=64256 Len=0 TSval=3519030162 TSecr=843867584"
,"15.773314027","172.16.40.1","172.16.1.10","TCP","66","[TCP Out-Of-Order] 55956 > 45995 [FIN, ACK] Seq=1 Ack=64 Win=64256 Len=0 TSval=3519030162 TSecr=843867584"
,"15.773517618","172.16.1.10","172.16.40.1","TCP","66","21 > 33828 [FIN, ACK] Seq=494 Ack=48 Win=65280 Len=0 TSval=843878388 TSecr=3519030161"
,"15.773529352","172.16.40.1","172.16.1.10","TCP","66","33828 > 21 [ACK] Seq=49 Ack=495 Win=64128 Len=0 TSval=3519030162 TSecr=843878388"
,"15.773580965","172.16.1.10","172.16.40.1","TCP","66","21 > 33828 [ACK] Seq=495 Ack=49 Win=65280 Len=0 TSval=843878388 TSecr=3519030162"
,"15.974959265","172.16.40.1","172.16.1.10","TCP","66","[TCP Retransmission] 55956 > 45995 [FIN, ACK] Seq=1 Ack=64 Win=64256 Len=0 TSval=3519030363 TSecr=843867584"
,"15.996801509","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"
,"16.178950371","172.16.40.1","172.16.1.10","TCP","66","[TCP Retransmission] 55956 > 45995 [FIN, ACK] Seq=1 Ack=64 Win=64256 Len=0 TSval=3519030567 TSecr=843867584"
```

Fig 2 - Transferência do ficheiro README, do servidor *netlab*, no *Tux43*



"No.", "Time", "Source", "Destination", "Protocol", "Length", "Info"			
"1", "0.000000000", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"2", "2.002127787", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"3", "4.004292329", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"4", "6.006448574", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"5", "8.008605878", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"6", "10.010825352", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"7", "12.012984069", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"8", "12.446240761", "3Com_9f:81:2e", "Broadcast", "ARP", "42", "Who has 172.16.40.254? Tell 172.16.40.1"			
"9", "12.446331069", "Kye_02:55:95", "3Com_9f:81:2e", "ARP", "60", "172.16.40.254 is at 00:c0:df:02:55:95"			
"10", "12.446350696", "172.16.40.1", "172.16.40.254", "ICMP", "98", "Echo (ping) request id=0x0670, seq=1/256, ttl=64 (reply in 11)"			
"11", "12.446449665", "172.16.40.254", "172.16.40.1", "ICMP", "98", "Echo (ping) reply id=0x0670, seq=1/256, ttl=64 (request in 10)"			
"12", "13.455376680", "172.16.40.1", "172.16.40.254", "ICMP", "98", "Echo (ping) request id=0x0670, seq=2/512, ttl=64 (reply in 13)"			
"13", "13.455481656", "172.16.40.254", "172.16.40.1", "ICMP", "98", "Echo (ping) reply id=0x0670, seq=2/512, ttl=64 (request in 12)"			
"14", "14.015151381", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"15", "14.479378272", "172.16.40.1", "172.16.40.254", "ICMP", "98", "Echo (ping) request id=0x0670, seq=3/768, ttl=64 (reply in 16)"			
"16", "14.479480943", "172.16.40.254", "172.16.40.1", "ICMP", "98", "Echo (ping) reply id=0x0670, seq=3/768, ttl=64 (request in 15)"			
"17", "15.503374801", "172.16.40.1", "172.16.40.254", "ICMP", "98", "Echo (ping) request id=0x0670, seq=4/1024, ttl=64 (reply in 18)"			
"18", "15.503465110", "172.16.40.254", "172.16.40.1", "ICMP", "98", "Echo (ping) reply id=0x0670, seq=4/1024, ttl=64 (request in 17)"			
"19", "16.017316251", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"20", "16.527374860", "172.16.40.1", "172.16.40.254", "ICMP", "98", "Echo (ping) request id=0x0670, seq=5/1280, ttl=64 (reply in 21)"			
"21", "16.527466705", "172.16.40.254", "172.16.40.1", "ICMP", "98", "Echo (ping) reply id=0x0670, seq=5/1280, ttl=64 (request in 20)"			
"22", "17.592867642", "Kye_02:55:95", "3Com_9f:81:2e", "ARP", "60", "Who has 172.16.40.1? Tell 172.16.40.254"			
"23", "17.592891739", "3Com_9f:81:2e", "Kye_02:55:95", "ARP", "42", "172.16.40.1 is at 00:01:02:9f:81:2e"			
"24", "18.019502982", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"25", "20.021634325", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"26", "22.023830898", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"27", "24.025988911", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"28", "26.028157881", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"29", "27.216421407", "172.16.40.1", "172.16.40.255", "NBNS", "92", "Name query NB WORKGROUP<id>"			
"30", "28.030302256", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"31", "28.217945369", "172.16.40.1", "172.16.40.255", "NBNS", "92", "Name query NB WORKGROUP<1b>"			
"32", "30.032435582", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"33", "32.034600884", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"34", "34.036765121", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"35", "36.038949035", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			
"36", "38.041113443", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bb Cost = 0 Port = 0x8003"			

Fig 3 - Captura da *Tux43*, no momento do *Ping* para o *Tux43*

54,"98.105191919", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"			
55,"98.564488865", "172.16.40.1", "172.16.40.255", "ICMP", "98", "Echo (ping) request id=0x09f4, seq=2/512, ttl=64 (no response found!)"			
56,"99.588487773", "172.16.40.1", "172.16.40.255", "ICMP", "98", "Echo (ping) request id=0x09f4, seq=3/768, ttl=64 (no response found!)"			
57,"100.107333907", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"			
58,"100.612491023", "172.16.40.1", "172.16.40.255", "ICMP", "98", "Echo (ping) request id=0x09f4, seq=4/1024, ttl=64 (no response found!)"			
59,"101.636489815", "172.16.40.1", "172.16.40.255", "ICMP", "98", "Echo (ping) request id=0x09f4, seq=5/1280, ttl=64 (no response found!)"			
60,"102.109476430", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"			
61,"102.660488689", "172.16.40.1", "172.16.40.255", "ICMP", "98", "Echo (ping) request id=0x09f4, seq=6/1536, ttl=64 (no response found!)"			
62,"103.578534008", "0.0.0.0", "255.255.255.255", "MNDP", "159", "5678 > 5678 Len=117"			
63,"103.578562155", "Routerbo_1c:8b:bd", "CDP/VTP/DTP/PagP/UDLD", "CDP", "93", "Device ID: MikroTik Port ID: bridge40 "			
64,"103.578608392", "Routerbo_1c:8b:bd", "LLDP_Multicast", "LLDP", "110", "TTL = 120 System Name = MikroTik System Description = MikroTik RouterOS 6.43.16 (long-term) CRS326-24G-2S+ "			
65,"103.684489111", "172.16.40.1", "172.16.40.255", "ICMP", "98", "Echo (ping) request id=0x09f4, seq=7/1792, ttl=64 (no response found!)"			
66,"104.111607613", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"			
67,"104.708488917", "172.16.40.1", "172.16.40.255", "ICMP", "98", "Echo (ping) request id=0x09f4, seq=8/2048, ttl=64 (no response found!)"			
68,"105.732488384", "172.16.40.1", "172.16.40.255", "ICMP", "98", "Echo (ping) request id=0x09f4, seq=9/2304, ttl=64 (no response found!)"			
69,"106.113744359", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"			
70,"106.756495617", "172.16.40.1", "172.16.40.255", "ICMP", "98", "Echo (ping) request id=0x09f4, seq=10/2560, ttl=64 (no response found!)"			
71,"107.780498600", "172.16.40.1", "172.16.40.255", "ICMP", "98", "Echo (ping) request id=0x09f4, seq=11/2816, ttl=64 (no response found!)"			
72,"108.115882687", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"			
73,"108.804494052", "172.16.40.1", "172.16.40.255", "ICMP", "98", "Echo (ping) request id=0x09f4, seq=12/3072, ttl=64 (no response found!)"			
74,"109.828489725", "172.16.40.1", "172.16.40.255", "ICMP", "98", "Echo (ping) request id=0x09f4, seq=13/3328, ttl=64 (no response found!)"			
75,"110.118035446", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"			
76,"110.852495536", "172.16.40.1", "172.16.40.255", "ICMP", "98", "Echo (ping) request id=0x09f4, seq=14/3584, ttl=64 (no response found!)"			
77,"111.876489556", "172.16.40.1", "172.16.40.255", "ICMP", "98", "Echo (ping) request id=0x09f4, seq=15/3840, ttl=64 (no response found!)"			
78,"112.120177353", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"			
79,"112.900504959", "172.16.40.1", "172.16.40.255", "ICMP", "98", "Echo (ping) request id=0x09f4, seq=16/4096, ttl=64 (no response found!)"			
80,"114.122288433", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"			
81,"116.104457008", "Routerbo_1c:8b:bd", "Spanning-tree-(for-bridges)_00", "STP", "60", "RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"			

Fig 4 - Captura do *Tux43*, no momento do *Ping broadcast*

9,"16.016548518","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
10,"18.018617223","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
11,"20.020744332","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
12,"22.022801826","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
13,"24.024885316","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
14,"26.026953036","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
15,"28.029043118","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
16,"30.031083974","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
17,"32.033145444","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
18,"34.035257767","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
19,"36.037323026","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
20,"38.039357143","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
21,"40.041431216","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
22,"42.043493071","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
23,"43.212615274","0.0.0.0","255.255.255.255","MNDP","159","5678 > 5678 Len=117"  
24,"43.212640487","Routerbo\_1c:8b:bd","CDP/VTP/DTP/PAGP/UDLD","CDP","93","Device ID: MikroTik Port ID: bridge40 "  
25,"43.212690144","Routerbo\_1c:8b:bd","LLDP\_Multicast","LLDP","110","TTL = 120 System Name = MikroTik System Description = MikroTik RouterOS 6.43.16 (long-term) CRS326-24G-2S+ "  
26,"44.045575673","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
27,"46.047648639","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
28,"48.049718465","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
29,"50.051816227","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
30,"52.053875576","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
31,"54.056006161","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
32,"56.058058382","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
33,"58.060126942","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
34,"60.062170075","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"  
35,"62.064197069","Routerbo\_1c:8b:bd","Spanning-tree-(for-bridges)\_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"

Fig 5 - Captura do Tux42, no momento do *Ping broadcast*

```
"3","4.005012658","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"4","6.007090428","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"5","8.009202397","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"6","10.011299884","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"7","12.013407473","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"8","14.015544999","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"9","16.017633820","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"10","18.019737422","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"11","19.252527665","172.16.40.1","172.16.40.1","ICMP","98","Echo (ping) request id=0x0ebd, seq=1/256, ttl=64 (reply in 12)"\n
"12","19.252828270","172.16.40.1","172.16.40.1","ICMP","98","Echo (ping) reply id=0x0ebd, seq=1/256, ttl=63 (request in 11)"\n
"13","20.022166607","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"14","20.256221978","172.16.40.1","172.16.40.1","ICMP","98","Echo (ping) request id=0x0ebd, seq=2/512, ttl=64 (reply in 15)"\n
"15","20.256495204","172.16.40.1","172.16.40.1","ICMP","98","Echo (ping) reply id=0x0ebd, seq=2/512, ttl=63 (request in 14)"\n
"16","21.280210354","172.16.40.1","172.16.40.1","ICMP","98","Echo (ping) request id=0x0ebd, seq=3/768, ttl=64 (reply in 17)"\n
"17","21.280451732","172.16.40.1","172.16.40.1","ICMP","98","Echo (ping) reply id=0x0ebd, seq=3/768, ttl=63 (request in 16)"\n
"18","22.023962499","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"19","22.304280879","172.16.40.1","172.16.40.1","ICMP","98","Echo (ping) request id=0x0ebd, seq=4/1024, ttl=64 (reply in 20)"\n
"20","22.304477345","172.16.40.1","172.16.40.1","ICMP","98","Echo (ping) reply id=0x0ebd, seq=4/1024, ttl=63 (request in 19)"\n
"21","23.328205299","172.16.40.1","172.16.40.1","ICMP","98","Echo (ping) request id=0x0ebd, seq=5/1280, ttl=64 (reply in 22)"\n
"22","23.328441718","172.16.40.1","172.16.40.1","ICMP","98","Echo (ping) reply id=0x0ebd, seq=5/1280, ttl=63 (request in 21)"\n
"23","24.026100875","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"24","24.352209634","172.16.40.1","172.16.40.1","ICMP","98","Echo (ping) request id=0x0ebd, seq=6/1536, ttl=64 (reply in 25)"\n
"25","24.352443678","172.16.40.1","172.16.40.1","ICMP","98","Echo (ping) reply id=0x0ebd, seq=6/1536, ttl=63 (request in 24)"\n
"26","24.384280294","3Com_9f:81:2e","Kye_02:55:95","ARP","42","Who has 172.16.40.254? Tell 172.16.40.1"\n
"27","24.384395116","Kye_02:55:95","3Com_9f:81:2e","ARP","60","172.16.40.254 is at 00:c0:df:02:55:95"\n
"28","24.401562848","Kye_02:55:95","3Com_9f:81:2e","ARP","60","Who has 172.16.40.1? Tell 172.16.40.254"\n
"29","24.401579890","3Com_9f:81:2e","Kye_02:55:95","ARP","42","172.16.40.1 is at 00:01:02:9f:81:2e"\n
"30","26.028180907","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"31","28.030304145","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"32","30.032461650","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"33","30.256193805","0.0.0.0","255.255.255.255","MNDP","159","5678 > 5678 Len=117"\n
"34","30.256222162","Routerbo_1c:8b:bd","CDP/VTP/DTP/PAGP/UDLD","CDP","93","Device ID: MikroTik Port ID: bridge40 "\n
"35","30.256271481","Routerbo_1c:8b:bd","LLDP_Multicast","LLDP","110","TTL = 120 System Name = MikroTik System Description = MikroTik RouterOS 6.43.16 (long-term) CRS326-24G-2S+ "\n
"36","32.034605160","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"37","34.036737468","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"38","36.038855292","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"39","38.040989850","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"\n
"40","40.043121588","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:bd Cost = 0 Port = 0x8001"
```

Fig 6 - Captura do Tux43, no momento da *Ping* para o *Tux42*

42	"4.454451412", "Cisco_7c:8f:86", "Spanning-tree-(for-bridges)_00", "STP", "60", "Conf. Root = 32768/0/4c:00:82:2e:9a:00 Cost = 19 Port = 0x8006"				
43	"4.712153626", "f0:2f:74:2e:20:7c", "Broadcast", "ARP", "60", "Who has 192.168.109.126? Tell 192.168.109.123"				
44	"4.712166477", "f0:2f:74:2e:20:7c", "Broadcast", "ARP", "60", "Who has 192.168.109.125? Tell 192.168.109.123"				
45	"4.712169201", "f0:2f:74:2e:20:7c", "Broadcast", "ARP", "60", "Who has 192.168.109.124? Tell 192.168.109.123"				
46	"4.712171366", "f0:2f:74:2e:20:7c", "Broadcast", "ARP", "60", "Who has 192.168.109.122? Tell 192.168.109.123"				
47	"4.712173531", "f0:2f:74:2e:20:7c", "Broadcast", "ARP", "60", "Who has 192.168.109.121? Tell 192.168.109.123"				
48	"5.004862810", "AsustekC_b3:e9:e8", "Broadcast", "ARP", "60", "Who has 192.168.109.1? Tell 192.168.109.113"				
49	"5.049593940", "10.227.20.187", "10.227.20.3", "DNS", "70", "Standard query 0x255c A google.com"				
50	"5.049608676", "10.227.20.187", "10.227.20.3", "DNS", "70", "Standard query 0xc165 AAAA google.com"				
51	"5.050185575", "10.227.20.3", "10.227.20.187", "DNS", "86", "Standard query response 0x255c A google.com A 142.250.200.142"				
52	"5.050204502", "10.227.20.3", "10.227.20.187", "DNS", "98", "Standard query response 0xc165 AAAA google.com AAAA 2a00:1450:4003:80f::200e"				
53	"5.050624325", "10.227.20.187", "142.250.200.142", "ICMP", "98", "Echo (ping) request id=0x1eeb, seq=1/256, ttl=64 (reply in 54)"				
54	"5.067843481", "142.250.200.142", "10.227.20.187", "ICMP", "98", "Echo (ping) reply id=0x1eeb, seq=1/256, ttl=112 (request in 53)"				
55	"5.068025561", "10.227.20.187", "10.227.20.3", "DNS", "88", "Standard query 0x25cf PTR 142.200.250.142.in-addr.arpa"				
56	"5.068538902", "10.227.20.3", "10.227.20.187", "DNS", "127", "Standard query response 0x25cf PTR 142.200.250.142.in-addr.arpa PTR mad41s14-in-f14.1e100.net"				
57	"6.010720048", "AsustekC_b3:e9:e8", "Broadcast", "ARP", "60", "Who has 192.168.109.1? Tell 192.168.109.113"				
58	"6.028919655", "AsustekC_b3:e9:e8", "Broadcast", "ARP", "60", "Who has 192.168.109.116? Tell 192.168.109.113"				
59	"6.028946822", "AsustekC_b3:e9:e8", "Broadcast", "ARP", "60", "Who has 192.168.109.115? Tell 192.168.109.113"				
60	"6.028950244", "AsustekC_b3:e9:e8", "Broadcast", "ARP", "60", "Who has 192.168.109.114? Tell 192.168.109.113"				
61	"6.028953806", "AsustekC_b3:e9:e8", "Broadcast", "ARP", "60", "Who has 192.168.109.112? Tell 192.168.109.113"				
62	"6.028956390", "AsustekC_b3:e9:e8", "Broadcast", "ARP", "60", "Who has 192.168.109.111? Tell 192.168.109.113"				
63	"6.051391518", "10.227.20.187", "142.250.200.142", "ICMP", "98", "Echo (ping) request id=0x1eeb, seq=2/512, ttl=64 (reply in 64)"				
64	"6.068828862", "142.250.200.142", "10.227.20.187", "ICMP", "98", "Echo (ping) reply id=0x1eeb, seq=2/512, ttl=112 (request in 63)"				
65	"6.298532233", "f0:2f:74:2e:20:7c", "Broadcast", "ARP", "60", "Who has 192.168.109.1? Tell 192.168.109.123"				
66	"6.478808551", "Cisco_7c:8f:86", "Spanning-tree-(for-bridges)_00", "STP", "60", "Conf. Root = 32768/0/4c:00:82:2e:9a:00 Cost = 19 Port = 0x8006"				
67	"6.882568843", "f0:2f:74:2e:20:7c", "Broadcast", "ARP", "60", "Who has 192.168.109.126? Tell 192.168.109.123"				

Fig 7 - Captura do Tux 43, no momento do *Ping* para [google.com](https://www.google.com)

42	"4.165224751", "AsustekC_b3:e9:e8", "Broadcast", "ARP", "60", "Who has 192.168.109.112? Tell 192.168.109.113"				
43	"4.165226242", "AsustekC_b3:e9:e8", "Broadcast", "ARP", "60", "Who has 192.168.109.111? Tell 192.168.109.113"				
44	"4.223991950", "f0:2f:74:2e:20:7c", "Broadcast", "ARP", "60", "Who has 192.168.109.126? Tell 192.168.109.123"				
45	"4.223995581", "f0:2f:74:2e:20:7c", "Broadcast", "ARP", "60", "Who has 192.168.109.125? Tell 192.168.109.123"				
46	"4.223997258", "f0:2f:74:2e:20:7c", "Broadcast", "ARP", "60", "Who has 192.168.109.124? Tell 192.168.109.123"				
47	"4.223998864", "f0:2f:74:2e:20:7c", "Broadcast", "ARP", "60", "Who has 192.168.109.122? Tell 192.168.109.123"				
48	"4.224000540", "f0:2f:74:2e:20:7c", "Broadcast", "ARP", "60", "Who has 192.168.109.121? Tell 192.168.109.123"				
49	"4.598034027", "Cisco_7c:8f:86", "Spanning-tree-(for-bridges)_00", "STP", "60", "Conf. Root = 32768/0/4c:00:82:2e:9a:00 Cost = 19 Port = 0x8005"				
50	"5.193876554", "AsustekC_b3:e9:e8", "Broadcast", "ARP", "60", "Who has 192.168.109.1? Tell 192.168.109.113"				
51	"5.402608128", "10.227.20.72", "10.227.20.3", "DNS", "70", "Standard query 0x8b37 A google.com"				
52	"5.402619303", "10.227.20.72", "10.227.20.3", "DNS", "70", "Standard query 0x6843 AAAA google.com"				
53	"5.403127818", "10.227.20.3", "10.227.20.72", "DNS", "86", "Standard query response 0x8b37 A google.com A 142.250.200.142"				
54	"5.403145279", "10.227.20.3", "10.227.20.72", "DNS", "98", "Standard query response 0x6843 AAAA google.com AAAA 2a00:1450:4003:80f::200e"				
55	"5.403442664", "10.227.20.72", "142.250.200.142", "ICMP", "98", "Echo (ping) request id=0x1a03, seq=1/256, ttl=64 (reply in 59)"				
56	"5.420558574", "Routerbo_20:25:c8", "Broadcast", "ARP", "60", "Who has 10.227.20.72? Tell 10.227.20.254"				
57	"5.420571843", "HewlettP_19:02:ba", "Routerbo_20:25:c8", "ARP", "42", "10.227.20.72 is at 00:22:64:19:02:ba"				
58	"5.420711177", "142.250.200.142", "10.227.20.72", "ICMP", "98", "Echo (ping) reply id=0x1a03, seq=1/256, ttl=112 (request in 55)"				
59	"5.420824809", "10.227.20.72", "10.227.20.3", "DNS", "88", "Standard query 0x0134 PTR 142.200.250.142.in-addr.arpa"				
60	"5.421190150", "10.227.20.3", "10.227.20.72", "DNS", "127", "Standard query response 0x0134 PTR 142.200.250.142.in-addr.arpa PTR mad41s14-in-f14.1e100.net"				
61	"6.114012239", "AsustekC_b3:e9:e8", "Broadcast", "ARP", "60", "Who has 192.168.109.116? Tell 192.168.109.113"				
62	"6.114028024", "AsustekC_b3:e9:e8", "Broadcast", "ARP", "60", "Who has 192.168.109.115? Tell 192.168.109.113"				
63	"6.114029909", "AsustekC_b3:e9:e8", "Broadcast", "ARP", "60", "Who has 192.168.109.114? Tell 192.168.109.113"				
64	"6.114031516", "AsustekC_b3:e9:e8", "Broadcast", "ARP", "60", "Who has 192.168.109.112? Tell 192.168.109.113"				
65	"6.114033541", "AsustekC_b3:e9:e8", "Broadcast", "ARP", "60", "Who has 192.168.109.111? Tell 192.168.109.113"				
66	"6.213067770", "AsustekC_b3:e9:e8", "Broadcast", "ARP", "60", "Who has 192.168.109.1? Tell 192.168.109.113"				
67	"6.378550222", "f0:2f:74:2e:20:7c", "Broadcast", "ARP", "60", "Who has 192.168.109.126? Tell 192.168.109.123"				

Fig 8 - Captura do Tux 42, no momento do *Ping* para [google.com](https://www.google.com)



15,"1.212737511","AsustekC\_b3:e9:e8","Broadcast","ARP","60","Who has 192.168.109.114? Tell 192.168.109.113"  
16,"1.212739327","AsustekC\_b3:e9:e8","Broadcast","ARP","60","Who has 192.168.109.115? Tell 192.168.109.113"  
17,"1.212741003","AsustekC\_b3:e9:e8","Broadcast","ARP","60","Who has 192.168.109.116? Tell 192.168.109.113"  
18,"1.2127405892","AsustekC\_b3:e9:e8","Broadcast","ARP","60","Who has 192.168.109.117? Tell 192.168.109.113"  
19,"1.560453909","f0:2f:74:2e:20:7c","Broadcast","ARP","60","Who has 192.168.109.118? Tell 192.168.109.123"  
20,"1.875438922","f0:2f:74:2e:20:7c","Broadcast","ARP","60","Who has 192.168.109.122? Tell 192.168.109.123"  
21,"1.875445347","f0:2f:74:2e:20:7c","Broadcast","ARP","60","Who has 192.168.109.124? Tell 192.168.109.123"  
22,"1.875447233","f0:2f:74:2e:20:7c","Broadcast","ARP","60","Who has 192.168.109.125? Tell 192.168.109.123"  
23,"1.875448909","f0:2f:74:2e:20:7c","Broadcast","ARP","60","Who has 192.168.109.126? Tell 192.168.109.123"  
24,"2.013262949","Cisco\_7c:8f:84","Spanning-tree-(for-bridges)\_00","STP","60","Conf. Root = 32768/0/4c:00:82:2e:9a:00 Cost = 19 Port = 0x8004"  
25,"2.100313537","10.227.20.74","10.227.20.3","DNS","70","Standard query 0xe662 A google.com"  
26,"2.100322407","10.227.20.74","10.227.20.3","DNS","70","Standard query 0x926b AAAA google.com"  
27,"2.100797536","10.227.20.3","10.227.20.74","DNS","86","Standard query response 0xe662 A google.com A 142.250.200.142"  
28,"2.100818908","10.227.20.3","10.227.20.74","DNS","98","Standard query response 0x926b AAAA google.com AAAA 2a00:1450:4003:80f::200e"  
29,"2.101107282","10.227.20.74","142.250.200.142","ICMP","98","Echo (ping) request id=0x1c4f, seq=1/256, ttl=64 (reply in 32)"  
30,"2.119031374","Routerbo\_20:25:c8","Routerbo\_20:25:c8","ARP","42","10.227.20.74 is at 00:21:5a:c3:78:76"  
31,"2.119037730","HewlettP\_c3:78:76","Routerbo\_20:25:c8","ARP","42","10.227.20.74 is at 00:21:5a:c3:78:76"  
32,"2.119175107","142.250.200.142","10.227.20.74","ICMP","98","Echo (ping) reply id=0x1c4f, seq=1/256, ttl=112 (request in 29)"  
33,"2.119269602","10.227.20.74","10.227.20.3","DNS","88","Standard query 0xb09c PTR 142.200.250.142.in-addr.arpa"  
34,"2.119632496","10.227.20.3","10.227.20.74","DNS","127","Standard query response 0xb09c PTR 142.200.250.142.in-addr.arpa PTR mad41s14-in-f14.1e100.net"  
35,"2.236672997","AsustekC\_b3:e9:e8","Broadcast","ARP","60","Who has 192.168.109.116? Tell 192.168.109.113"  
36,"2.236676489","AsustekC\_b3:e9:e8","Broadcast","ARP","60","Who has 192.168.109.115? Tell 192.168.109.113"  
37,"2.236678515","AsustekC\_b3:e9:e8","Broadcast","ARP","60","Who has 192.168.109.114? Tell 192.168.109.113"  
38,"2.236680121","AsustekC\_b3:e9:e8","Broadcast","ARP","60","Who has 192.168.109.112? Tell 192.168.109.113"  
39,"2.236681727","AsustekC\_b3:e9:e8","Broadcast","ARP","60","Who has 192.168.109.111? Tell 192.168.109.113"  
40,"2.300648362","AsustekC\_b3:e9:e8","Broadcast","ARP","60","Who has 192.168.109.110? Tell 192.168.109.113"  
41,"2.579409795","f0:2f:74:2e:20:7c","Broadcast","ARP","60","Who has 192.168.109.119? Tell 192.168.109.123"  
42,"2.800453905","f0:2f:74:2e:20:7c","Broadcast","ARP","60","Who has 192.168.109.120? Tell 192.168.109.123"

Fig 9 - Captura do Tux 44, no momento do Ping para [google.com](https://www.google.com)

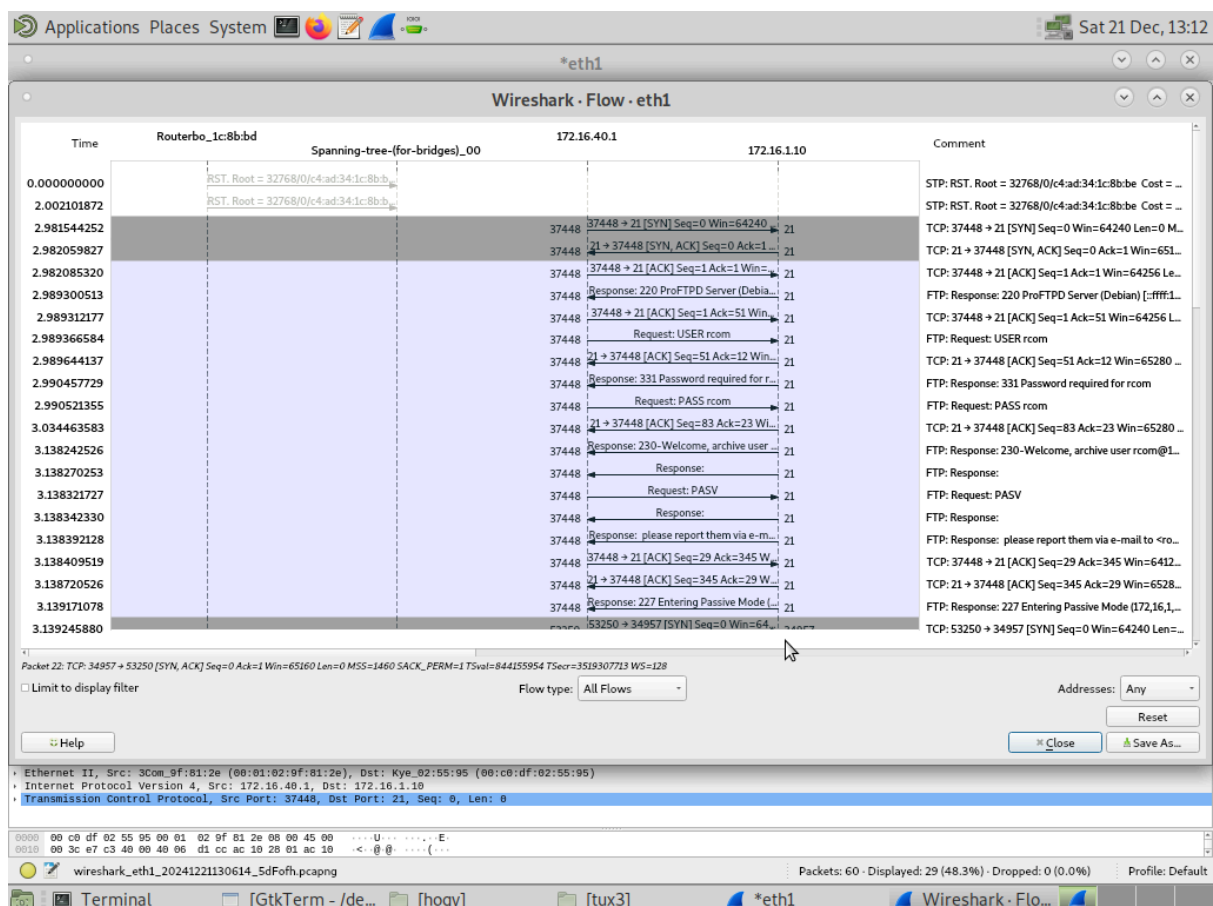


Fig 10 - Gráfico de Flow da primeira transferência

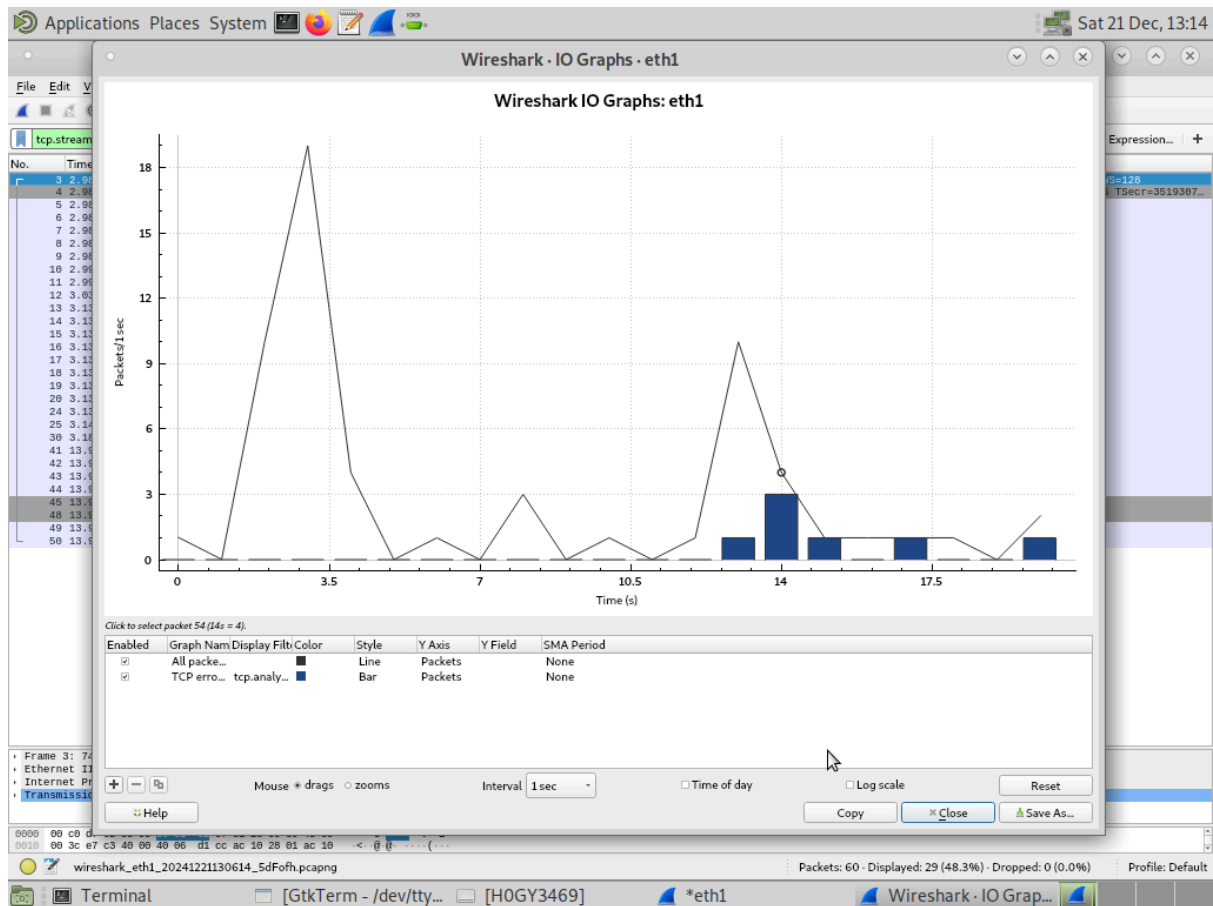


Fig 11 - Gráfico de I/O da primeira transferência

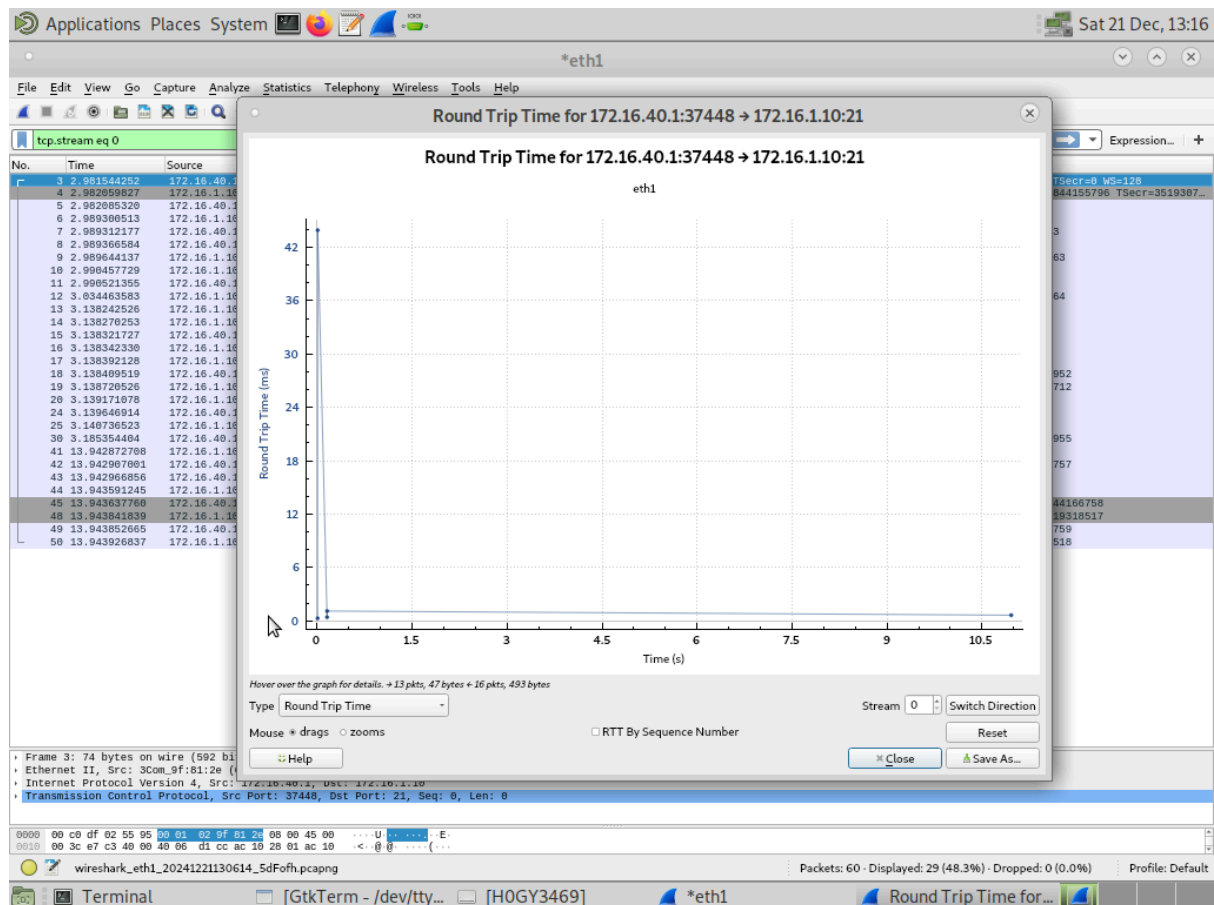


Fig 12 - Gráfico de *Round Trip Time* da primeira transferência

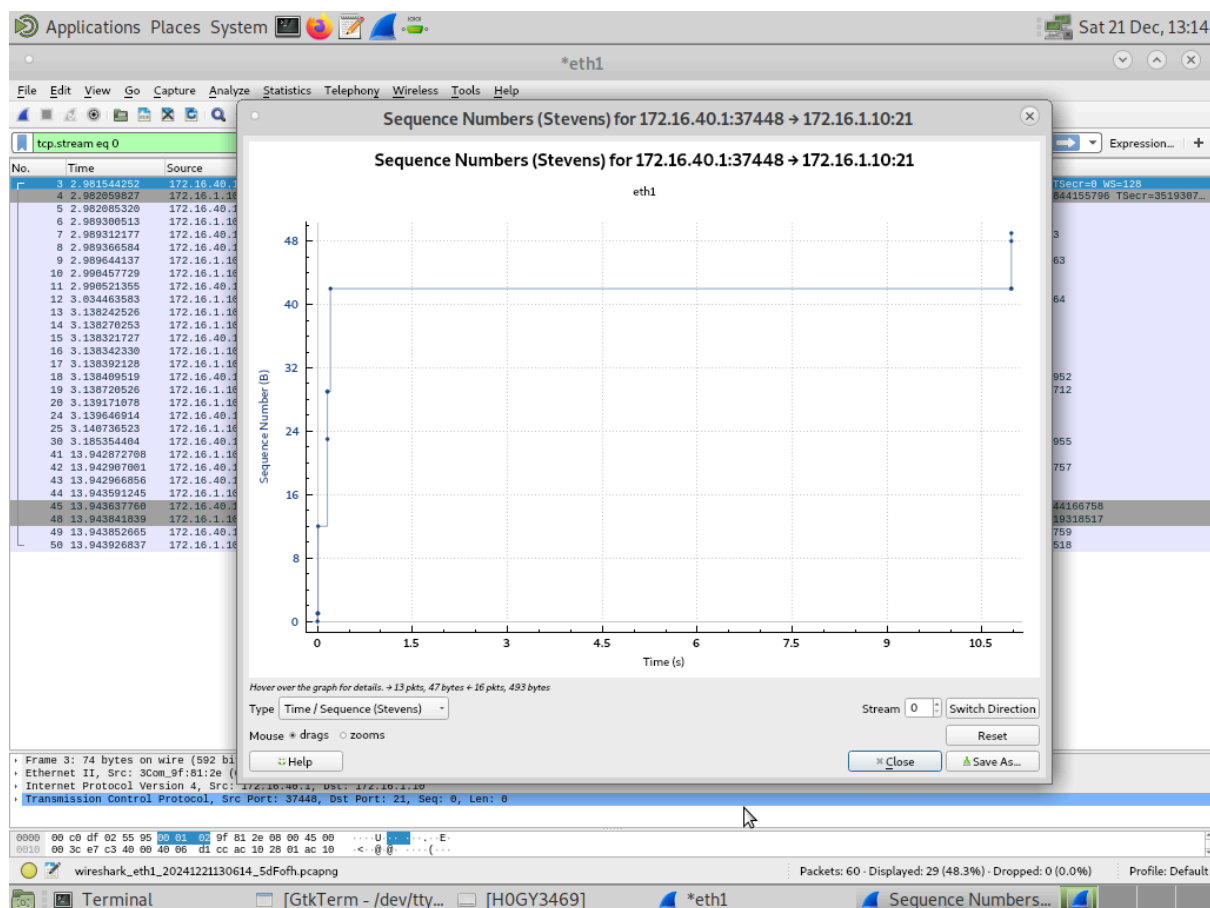


Fig 13 - Gráfico de Números de sequência da primeira transferência



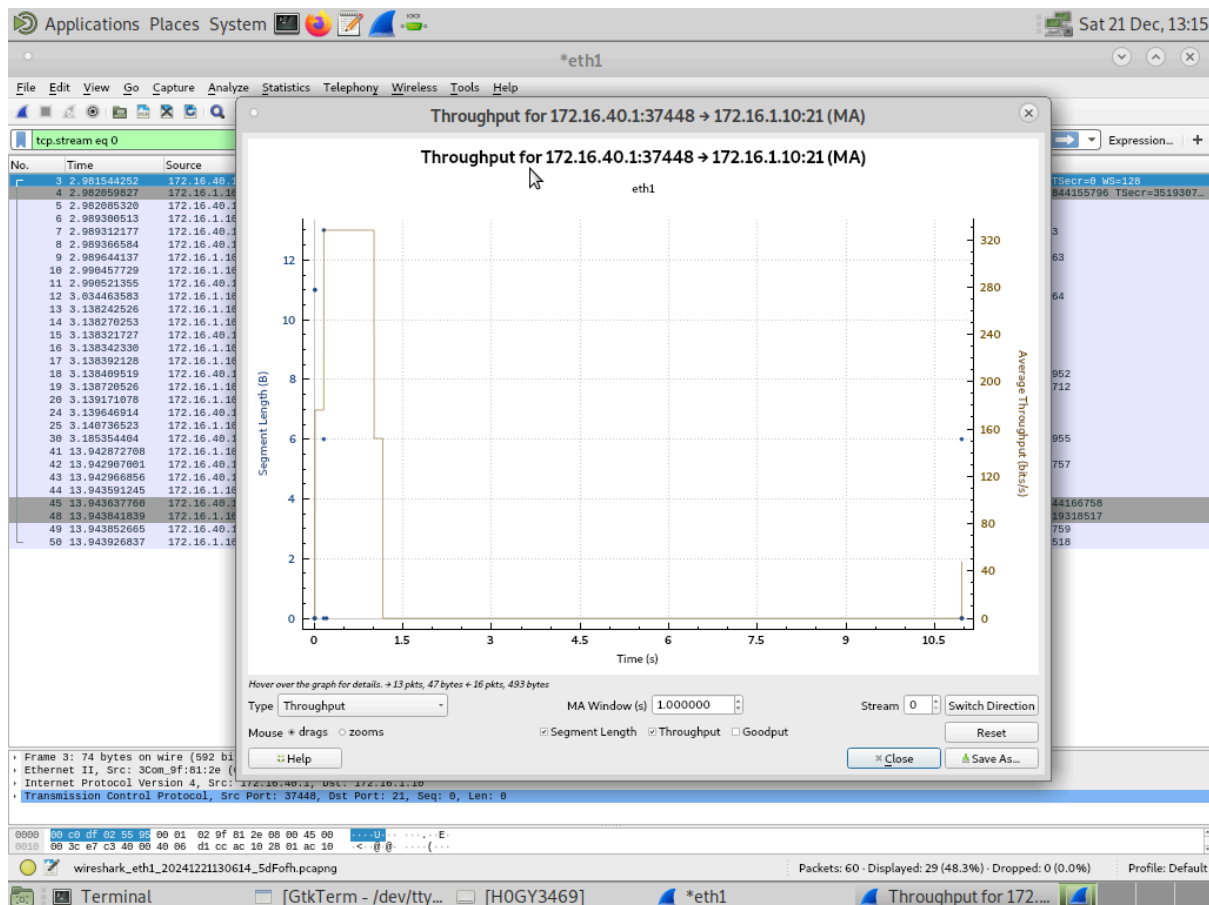


Fig 14 - Gráfico de *Throughput* da primeira transferência

```

36,"26.575113290","172.16.40.1","172.16.1.10","TCP","74","41600 > 39043 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3520620389 TSecr=0 WS=128"
37,"26.575490090","172.16.1.10","172.16.40.1","TCP","74","39043 > 41600 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=845468695 TSecr=3520620389 WS=128"
38,"26.575509227","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3520620389 TSecr=845468695"
39,"26.575520471","172.16.40.1","172.16.1.10","FTP","83","Request: RETR ubuntu.iso"
40,"26.576589828","172.16.1.10","172.16.40.1","FTP","140","Response: 150 Opening ASCII mode data connection for ubuntu.iso (2773874688 bytes)"
41,"26.577217991","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"
42,"26.577227978","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=1449 Win=64128 Len=0 TSval=3520620391 TSecr=845468696"
43,"26.577340285","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"
44,"26.577349714","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=2897 Win=63488 Len=0 TSval=3520620391 TSecr=845468696"
45,"26.577463347","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"
46,"26.577471170","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=4345 Win=62592 Len=0 TSval=3520620391 TSecr=845468696"
47,"26.577587806","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"
48,"26.577595419","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=5793 Win=61568 Len=0 TSval=3520620391 TSecr=845468696"
49,"26.577716666","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"
50,"26.577726863","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=7241 Win=60672 Len=0 TSval=3520620391 TSecr=845468696"
51,"26.577839658","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"
52,"26.577848319","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=8689 Win=59648 Len=0 TSval=3520620391 TSecr=845468696"
53,"26.577961603","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"
54,"26.577970194","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=10137 Win=58752 Len=0 TSval=3520620392 TSecr=845468696"
55,"26.578086341","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"
56,"26.578094373","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=11585 Win=57728 Len=0 TSval=3520620392 TSecr=845468696"
57,"26.578207029","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"
58,"26.578215550","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=13033 Win=56832 Len=0 TSval=3520620392 TSecr=845468696"
59,"26.578330091","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"
60,"26.578337704","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=14481 Win=55808 Len=0 TSval=3520620392 TSecr=845468696"
61,"26.578456227","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"
62,"26.578464329","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=15929 Win=54912 Len=0 TSval=3520620392 TSecr=845468697"
63,"26.578576621","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"

```

Fig 15 - Início da transferência mútua

70,"26.576960000","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=21721 Win=51072 Len=0 TSval=3520620393 TSecr=845468698"					
71,"26.579073913","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
72,"26.579081596","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=23169 Win=50048 Len=0 TSval=3520620393 TSecr=845468698"					
73,"26.579197255","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
74,"26.579319619","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
75,"26.579443589","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
76,"26.579565813","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
77,"26.579689294","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
78,"26.579811309","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
79,"26.579935279","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
80,"26.580057573","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
81,"26.580181823","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
82,"26.580303698","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
83,"26.580426760","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
84,"26.580550032","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
85,"26.580673304","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
86,"26.580796366","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
87,"26.580919568","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
88,"26.581042561","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
89,"26.589848430","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
90,"26.589958223","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
91,"26.590080377","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
92,"26.590202950","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
93,"26.590325664","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
94,"26.590448866","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
95,"26.590576468","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
96,"26.611680327","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=56473 Win=27648 Len=0 TSval=3520620425 TSecr=845468698"					
97,"26.612091838","172.16.40.1","172.16.1.10","TCP","66","[TCP Window Update] 41600 > 39043 [ACK] Seq=1 Ack=56473 Win=62592 Len=0 TSval=3520620426 TSecr=845468698"					

Fig 16 - Meio da transferência mútua

98,"26.612534430","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
99,"26.612667759","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
100,"26.612676699","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=59369 Win=64128 Len=0 TSval=3520620426 TSecr=845468731"					
101,"26.612791659","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
102,"26.612914512","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
103,"26.612922893","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=62265 Win=64128 Len=0 TSval=3520620427 TSecr=845468731"					
104,"26.613037854","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
105,"26.613159868","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
106,"26.613168389","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=65161 Win=64128 Len=0 TSval=3520620427 TSecr=845468731"					
107,"26.613282931","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
108,"26.613406552","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
109,"26.613415492","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=68057 Win=64128 Len=0 TSval=3520620427 TSecr=845468731"					
110,"26.613529265","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
111,"26.613652467","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
112,"26.613660708","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=70953 Win=64128 Len=0 TSval=3520620427 TSecr=845468731"					
113,"26.613775040","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
114,"26.613898801","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
115,"26.613907601","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=73849 Win=64128 Len=0 TSval=3520620428 TSecr=845468731"					
116,"26.614022771","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
117,"26.614144437","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					
118,"26.614152957","172.16.40.1","172.16.1.10","TCP","66","41600 > 39043 [ACK] Seq=1 Ack=76745 Win=64128 Len=0 TSval=3520620428 TSecr=845468731"					
119,"26.614267220","172.16.1.10","172.16.40.1","FTP-DATA","1514","FTP Data: 1448 bytes (PASV) (RETR ubuntu.iso)"					

Fig 17 - Fim da transferência mútua

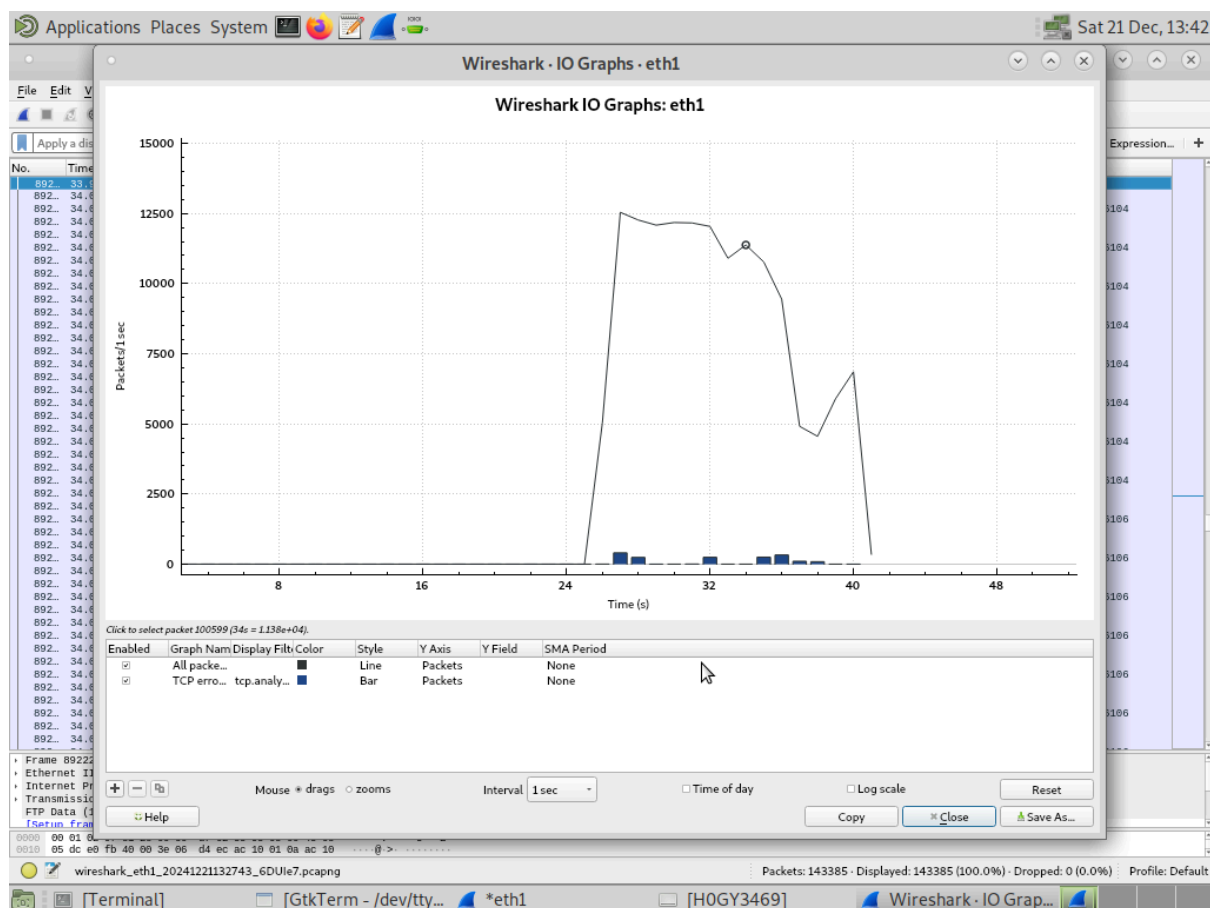


Fig 18 - Gráfico de I/O da segunda e terceira transferência

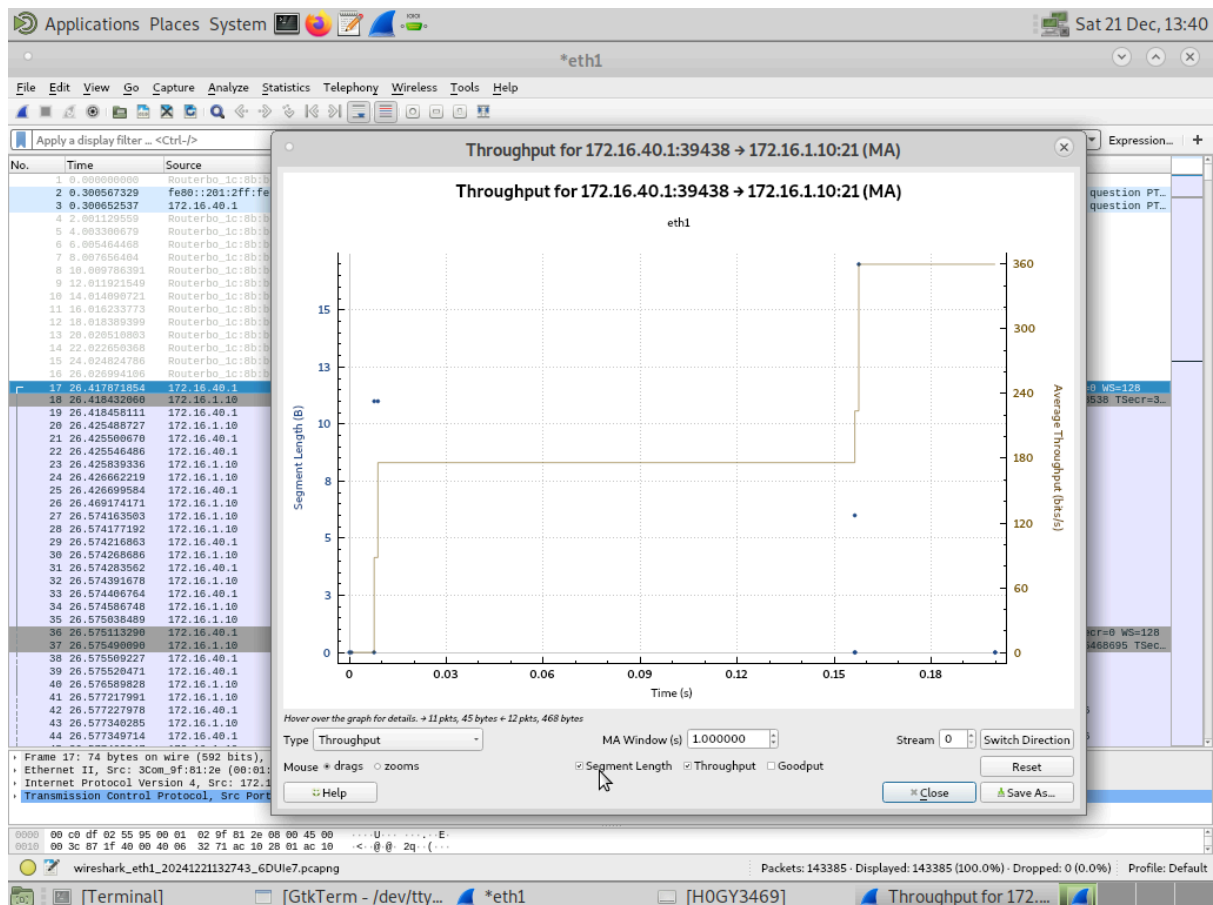


Fig 19 - Gráfico de *Throughput* da segunda e terceira transferência

```

"2","2.002581124","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002\n"
"3","4.003808985","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002\n"
"4","6.005891372","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002\n"
"5","8.007996530","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002\n"
"6","10.008068550","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002\n"
"7","12.002192992","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002\n"
"8","14.004297812","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002\n"
"9","16.006389366","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002\n"
"10","18.008590479","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002\n"
"11","20.010608312","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002\n"
"12","20.285244203","172.16.40.1","172.16.1.10","ICMP","98","Echo (ping) request id=0x1ac7, seq=1/256, ttl=64 (reply in 13)\n"
"13","20.285752725","172.16.1.10","172.16.40.1","ICMP","98","Echo (ping) reply id=0x1ac7, seq=1/256, ttl=62 (request in 12)\n"
"14","21.296597092","172.16.40.1","172.16.1.10","ICMP","98","Echo (ping) request id=0x1ac7, seq=2/512, ttl=64 (reply in 15)\n"
"15","21.296925169","172.16.1.10","172.16.40.1","ICMP","98","Echo (ping) reply id=0x1ac7, seq=2/512, ttl=62 (request in 14)\n"
"16","22.012748867","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002\n"
"17","22.320590579","172.16.40.1","172.16.1.10","ICMP","98","Echo (ping) request id=0x1ac7, seq=3/768, ttl=64 (reply in 18)\n"
"18","22.320947640","172.16.1.10","172.16.40.1","ICMP","98","Echo (ping) reply id=0x1ac7, seq=3/768, ttl=62 (request in 17)\n"
"19","23.344499234","172.16.40.1","172.16.1.10","ICMP","98","Echo (ping) request id=0x1ac7, seq=4/1024, ttl=64 (reply in 20)\n"
"20","23.344931070","172.16.1.10","172.16.40.1","ICMP","98","Echo (ping) reply id=0x1ac7, seq=4/1024, ttl=62 (request in 19)\n"
"21","24.014905263","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002\n"
"22","24.368592990","172.16.40.1","172.16.1.10","ICMP","98","Echo (ping) request id=0x1ac7, seq=5/1280, ttl=64 (reply in 23)\n"
"23","24.368938248","172.16.1.10","172.16.40.1","ICMP","98","Echo (ping) reply id=0x1ac7, seq=5/1280, ttl=62 (request in 22)\n"
"24","25.303921417","Kye_02:55:95","3Com_9f:81:2e","ARP","60","Who has 172.16.40.1? Tell 172.16.40.254\n"
"25","25.303947468","3Com_9f:81:2e","Kye_02:55:95","ARP","42","172.16.40.1 is at 00:01:02:9f:81:2e\n"
"26","25.392496758","172.16.40.1","172.16.1.10","ICMP","98","Echo (ping) request id=0x1ac7, seq=6/1536, ttl=64 (reply in 27)\n"
"27","25.392896886","172.16.1.10","172.16.40.1","ICMP","98","Echo (ping) reply id=0x1ac7, seq=6/1536, ttl=62 (request in 26)\n"
"28","25.488470647","3Com_9f:81:2e","Kye_02:55:95","ARP","42","Who has 172.16.40.254? Tell 172.16.40.1\n"
"29","25.488578623","Kye_02:55:95","3Com_9f:81:2e","ARP","60","172.16.40.254 is at 00:c0:df:02:55:95\n"
"30","26.016998463","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002\n"
"31","26.416498630","172.16.40.1","172.16.1.10","ICMP","98","Echo (ping) request id=0x1ac7, seq=7/1792, ttl=64 (reply in 32)\n"
"32","26.416914332","172.16.1.10","172.16.40.1","ICMP","98","Echo (ping) reply id=0x1ac7, seq=7/1792, ttl=62 (request in 31)\n"
"33","28.019103377","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002\n"
"34","30.021218701","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002\n"
"35","31.207789420","0.0.0.0","255.255.255.255","MNDP","159","5678 Len=117\n"
"36","31.207826018","Routerbo_1c:8b:be","CDP/VTP/DTP/PAGP/UDLD","CDP","93","Device ID: MikroTik Port ID: bridge40\n"
"37","31.207873207","Routerbo_1c:8b:be","LLDP","110","TTL = 120 System Name = MikroTik System Description = MikroTik RouterOS 6.43.16 (long-term) CRS326-24G-25+ \n"
"38","32.023167384","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002\n"
"39","34.025452222","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002\n"

```

Fig 20 - Captura do Tux43, no momento do *Ping* para o FTP server com NAT ativado

```
"10","12.002898752","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"11","14.005944116","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"12","16.007196366","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"13","18.013306999","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"14","20.015450044","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"15","22.017595318","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"16","23.438466417","172.16.40.1","172.16.1.10","ICMP","98","Echo (ping) request id=0x1b4b, seq=1/256, ttl=64 (no response found!)"\n
"17","24.019747439","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"18","24.448597444","172.16.40.1","172.16.1.10","ICMP","98","Echo (ping) request id=0x1b4b, seq=2/512, ttl=64 (no response found!)"\n
"19","25.472591875","172.16.40.1","172.16.1.10","ICMP","98","Echo (ping) request id=0x1b4b, seq=3/768, ttl=64 (no response found!)"\n
"20","26.021904802","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"21","26.496587011","172.16.40.1","172.16.1.10","ICMP","98","Echo (ping) request id=0x1b4b, seq=4/1024, ttl=64 (no response found!)"\n
"22","27.520594053","172.16.40.1","172.16.1.10","ICMP","98","Echo (ping) request id=0x1b4b, seq=5/1280, ttl=64 (no response found!)"\n
"23","28.024049526","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"24","28.544585521","172.16.40.1","172.16.1.10","ICMP","98","Echo (ping) request id=0x1b4b, seq=6/1536, ttl=64 (no response found!)"\n
"25","28.608541834","3Com_9f:81:2e","Kye_02:55:95","ARP","42","Who has 172.16.40.254? Tell 172.16.40.1"\n
"26","28.608645341","Kye_02:55:95","3Com_9f:81:2e","ARP","60","172.16.40.254 is at 00:c0:df:02:55:95"\n
"27","30.026184684","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"28","32.028572884","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"29","34.030494959","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"30","36.032640882","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"31","38.034786877","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"32","40.026915027","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"33","42.029076393","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"34","44.031245863","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"35","46.033396130","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"36","48.035530265","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"37","50.037710430","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"38","52.039848273","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"39","54.042044925","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"40","56.044194228","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"41","58.046338364","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"42","60.048444230","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"43","61.143419370","0.0.0.0","255.255.255.255","MNDP","159","5678 > 5678 Len=117"\n
"44","61.143450241","Routerbo_1c:8b:be","CDP/VTP/DTP/PAGP/ULLDP","CDP","93","Device ID: MikroTik Port ID: bridge40 "\n
"45","61.143499829","Routerbo_1c:8b:be","LLDP/Multicast","LLDP","110","TTL = 120 System Name = MikroTik System Description = MikroTik RouterOS 6.43.16 (Long-term) CRS326-24G-25+ "\n
"46","62.05061273","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"\n
"47","64.052743482","Routerbo_1c:8b:bd","Spanning-tree-(for-bridges)_00","STP","60","RST. Root = 32768/0/c4:ad:34:1c:8b:be Cost = 0 Port = 0x8002"
```

Fig 21 - Captura do Tux43, no momento do *Ping* para o FTP server com NAT desativado

```
root@gnu42:~# traceroute -n 172.16.40.1
traceroute to 172.16.40.1 (172.16.40.1), 30 hops max, 60 byte packets
 1  172.16.41.254  0.326 ms  0.313 ms  0.328 ms
 2  172.16.41.253  0.629 ms  0.623 ms  0.638 ms
 3  172.16.40.1  0.807 ms  0.833 ms  0.832 ms
root@gnu42:~# route add -net 172.16.40.0/24 gw 172.16.41.253
root@gnu42:~# traceroute -n 172.16.40.1
traceroute to 172.16.40.1 (172.16.40.1), 30 hops max, 60 byte packets
 1  172.16.41.253  0.181 ms  0.167 ms  0.152 ms
 2  172.16.40.1  0.369 ms  1.315 ms  2.287 ms
```

Fig 22 - Traceroutes realizados no Tux42 com a conexão ao Tux44 desconectada (caminho mais longo) e reconectada (caminho mais curto)