Protocolo de Aplicação

2º Trabalho Laboratorial

Redes de Computadores

Grupo 1 – Turma 8

Licenciatura em Engenharia Informática e Computação

Domingos José Silva Moreira dos Santos up201906680
Jorge Levi Perdigoto da Costa up201706518
Vitor de Sá Carneiro Bizarro up202007888

Índice

Sumário

Este presente relatório tem como fim destrinchar o desenvolvimento do 2º trabalho laboratorial da disciplina de Redes de Computadores, que consistia na criação de uma rede entre três computadores através de um switch e um router, e posteriormente com o acesso à internet, permitindo o download de um ficheiro.

Primeiramente, este relatório apresenta a aplicação de download criada ao longo da experiência. Como tal, os módulos utilizados para tal, são:

- Parsing: parsing dos argumentos do programa;
- Ligações: contém as funções de ligação aos sockets e de troca de informação, que culmina no download;
- Download: contém o procedimento para fazer o download de um ficheiro, com recurso aos módulos anteriormente mencionados.

De seguida, descrevemos e analisamos as duas experiências que fazem parte deste projeto, nomeadamente:

- Experiência 1: consiste em criar uma rede entre 2 computadores através de um switch. O objetivo é conhecer endereços MAC e IP, bem como pacotes ARP e comandos ping;
- Experiência 2: Através de mais um computador são criadas 2 bridges dentro do switch. Isto serve para provar que a comunicação entre dispositivos é apenas possível quando estão na mesma bridge;
- Experiência 3: Um dos computadores é configurado como um router para estabelecer ligação entre 2 bridges. Como tal, pretende-se saber o que são routes e como se configuram nesses contextos;

- Experiência 4: É configurado um router e a respetiva implementação da funcionalidade NAT, com o objetivo de preparar o acesso à internet;
- Experiência 5: Configuração de um servidor DNS em cada tux, com o intuito de traduzir os endereços normais para endereços IP;
- Experiência 6: Por fim, é testado o programa de download desenvolvido para verificar a qualidade do programa e da rede implementada;

O projeto foi demonstrado com sucesso, tendo o ficheiro sido transferido integralmente para o computador de modo satisfatório.

1. Introdução

Neste trabalho, foram aplicados os conhecimentos adquiridos ao longo de todo o semestre. Dentre estes conhecimentos estão os protocolos FTP e TCP/IP, bem como redes de computadores, routers, cabos, entre outros. O objetivo deste projeto é explorar o uso correto do protocolo FTP para realizar a transferência de um ficheiro, as funcionalidades de um switch e de um router, bem como a configuração de uma rede de computadores.

2. Parte I - Download Application

2.1. Arquitetura

A arquitetura caracteriza-se por um ficheiro main.c, que contém as funções: parser, createSocket, writeSocket, readSocket, manageSocket, writeToFile e main.

2.1.1. Parsing

A aplicação aceita como argumentos um link do tipo:
ftp://[<username>:<password>@<host>]/<url-path>
Ao executar a função “parser”, a aplicação guarda as respectivas informações do link numa estrutura chamada “url”, contendo as strings: “username”, “password”, “host” e “urlPath”.

```
struct url{
    char* username;
    char* password;
    char* host;
    char* urlPath;
```

Caso o link não contenha username e password, é guardada na estrutura o username “anonymous” e password nula.

2.1.2. Ligações

O endereço de ip é obtido através do host, com a função gethostbyname, que preenche uma estrutura “hostent” com informações do mesmo, e de seguida é criada uma socket com esse ip e server port 21.
É através desta socket que é feita a comunicação com o servidor, executando várias vezes a função “manageSocket”. Esta função envia inicialmente os comandos “user” e “pass” juntamente com as credenciais com a função “writeSocket” para efetuar login, e lê a resposta através da função “readSocket”, verificando o seu código.
O servidor entra em modo passivo com o envio do comando “pasv” e obtém-se o port para a criação de uma nova socket através da função “getPassivePort”. Essa socket é responsável pela transferência do ficheiro.

2.1.3. Download

Para iniciar o download, é enviado o comando “retr” pela socket 1, e é executada a função “writeToFile” que abre um ficheiro com o filename definido, e recebido através da socket 2.
No final, são fechadas as respectivas sockets.

3. Part II - Configuração da Network e análise

Experiência 1

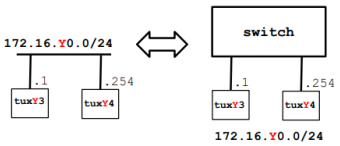


Figura 1: Estrutura da Experiência 1

O objetivo desta experiência é entender o que são pacotes ARP e a sua utilidade, que tipo de pacotes é que o comando ping gera e o que são endereços MAC e endereços IP.

Análise:

- Após esta experiência, concluiu-se que os pacotes ARP , sabendo o endereço IP de uma máquina, são utilizados para obter o endereço MAC. Estes identificam placas de rede, enquanto que os endereços IP assumem o papel de identificadores públicos que permitem uma máquina comunicar-se com outras, através de uma rede.
- Uma máquina pode assumir vários IPs, mas apenas somente 1 endereço MAC.
- O comando ping cria pacotes ICMP, que são utilizados por hosts e/ou routers para enviar erros da camada 3. Além disso, podem também enviar mensagens de controle para outros routers ou hosts.
- No contexto desta experiência, o comando ping permite concluir sobre a existência de conectividade entre computadores.

- Um loopback é uma interface de rede virtual, que fornece um endereço de IP sempre ativo ao router, não havendo interface física.
- Deste modo, ao analisar os logs desta experiência, verifica-se o formato dos pacotes ARP que são enviados aquando do ping, bem como os pacotes ICMP de ping.

Experiência 2

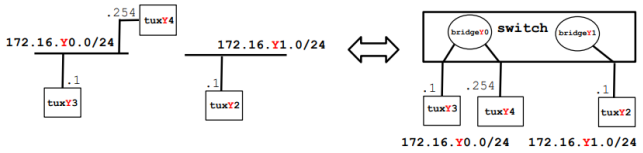


Figura 2: Estrutura da Experiência 2

O objetivo desta experiência foi estabelecer duas bridges e compreender a conectividade dos tux, após a sua configuração em cada uma das sub-redes.

Análise:

- De modo a configurar as bridges, obtivemos a bridge 30 e 31 e deste modo, associamos os tux 33 e 34 à bridge 30, e o tux 32 à 31.
- Configurou-se a bridge 30 através do switch, criando a bridge com o comando “/interface bridge add name=bridge30”, removemos as portas dos respectivos tux da bridge default com “/interface bridge port remove [find interface =ether2]” e ”/interface bridge port remove [find interface =ether3]”, e adicionamos essas portas à respectiva bridge com “/interface bridge port add bridge=bridge31 interface=ether2” e “/interface bridge port add bridge=bridge31 interface=ether3”.
- A fim de testar a conectividade entre os tux, recorreu-se ao ping do tux 33 até ao tux 34, obtendo-se o resultado esperado.
- Por outro lado, como não havia rota entre as bridges, entre o tux 33 e o tux 32 o ping não obteve resposta.
- Assim, apenas existem dois domínios de transmissão, pois ao fazer ping broadcast nos tuxs da bridge 30, respondem um ao outro, enquanto na bridge 31 o tux 32 está isolado.

Experiência 3

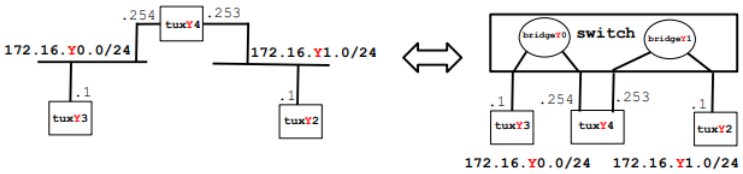


Figura 3: Estrutura da Experiência 3

O objetivo desta experiência foi configurar o tux 4 como router, para funcionar como um intercomunicador entre as bridges.

Análise:

- Para isso, configuramos a porta eth1 do tux 4, adicionamos à bridge 31 e definimos as rotas dos tux. Foi usado o comando “route add”: no tux 3, adicionamos a rota para a gama 1 (172.16.31.0/24) e como gateway o ip do tux 4 de gama 0 (172.16.30.254); no tux 2, adicionamos a rota para a gama 0 (172.16.30.0/24) e como gateway o ip do tux 4 de gama 1 (172.16.31.253).
- Assim foi possível fazer ping do tux 3 para o tux 2 e vice versa, havendo intercomunicação entre bridges.
- Analisando os logs, ao fazer ping do tux 3 para o 2, verifica-se que no pacote ICMP, o endereço de destino é o endereço MAC do tux 4. O mesmo se verifica do tux 2 para o 3. Podemos verificar isso na imagem “Experiência 3 - Tux 3 para Tux 2” dos logs dos anexos.

Experiência 4

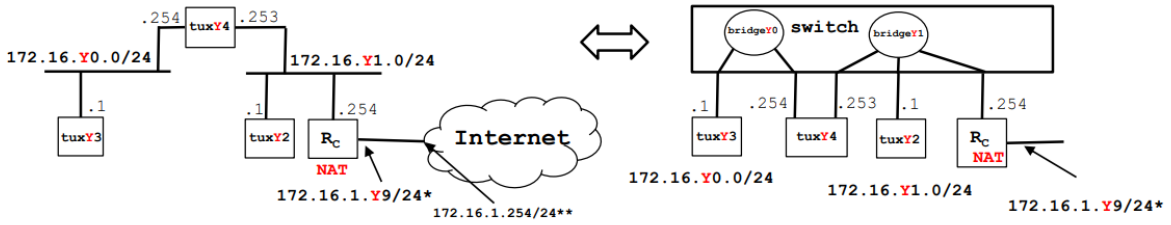


Figura 4: Estrutura da Experiência 4

O objetivo desta experiência é estabelecer uma ligação com a rede dos laboratórios e assim implementar rotas num router comercial e adicionar-lhe funcionalidade NAT.

Análise:

- Nesta experiência, o primeiro passo foi configurar as interfaces do router, usando os comandos “/ip address add address=172.16.Z.Y9/24 interface=ether1” para permitir a ligação com a rede dos laboratórios, e “/ip address add address=172.16.Y1.254/24 interface=ether2” para fazer conexão ao switch.
- De seguida definimos a rota do router usando o comando “/ip route add dst-address=172.16.Y0.0/24 gateway=172.16.Y1.253” para estabelecer ligação com a bridge 30, removemos a respectiva porta no switch, e adicionamos à bridge 31.
- Além disso foram definidas as rotas default, para todos os tux, sendo a do tux 3 o ip do tux 4, e as restantes os ip do router.
- Dessa forma, ao dar delete da rota do tux 2 para a bridge 30, e com o NAT ativo, mesmo não havendo ligação direta, é possível observar comunicação entre o tux 2 e 3 na imagem “Experiência 4 - Tux 2 para Tux 3 com redirect” dos logs nos anexos. Isto verifica-se pois o NAT tem como objetivo traduzir endereços privados e que os mesmos se consigam ligar a uma rede pública através do router.

Experiência 5

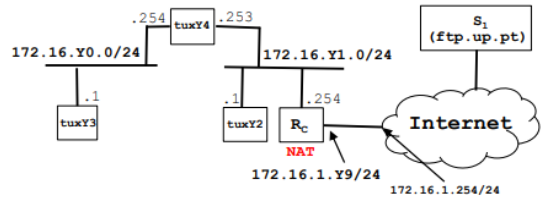


Figura 5: Estrutura da Experiência 5

Esta experiência tem como objetivo configurar o DNS em cada Tux, para assim poder haver acesso à internet.

Análise:

- A configuração é feita editando um ficheiro de texto chamado “etc/resolv.conf” e escrevendo no mesmo o DNS fornecido ou o DNS da Google. Neste caso usamos o comando “nano”.
- Ao enviar um pacote ao servidor, este pede o seu ip e responde enviando outro pacote com o ip do host.
- Podemos verificar o exemplo ao fazer ping ao DNS da Google na imagem “Experiência 5 - Tux 3 para Google” dos logs nos anexos.

Experiência 6

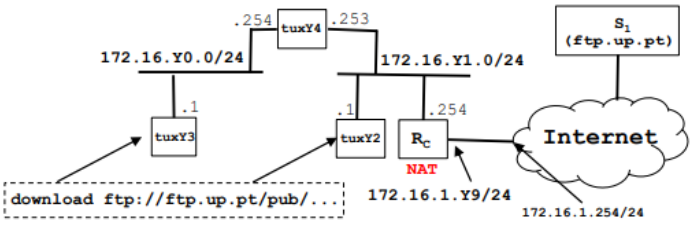


Figura 6: Estrutura da Experiência 6

Nesta experiência foi executada a aplicação de download desenvolvida.

Análise:

- Durante a execução da aplicação são abertas duas ligações TCP, sendo primeiramente uma para a porta 21 de controlo e de seguida uma para a transferência de dados, que no final ambas são fechadas.
- O receptor impede o processamento de pacotes caso tenha detectado um erro, usando o ARC TCP, o que faz com que o receptor continue a processar os seguintes, mas com um “acknowledgment number” (ack). Esse ack permite verificar

no final os pacotes não processados e reenviá-los. Além disso é de notar o “sequence number” que identifica o byte do pacote a ser enviado.

- Ao realizarem-se dois downloads simultaneamente, a velocidade de download diminui para metade.

4. Conclusões

Este projecto laboratorial permitiu-nos colocar em prática alguns dos conhecimentos que só tínhamos tido a oportunidade de discutir de modo teórico. Foram aprofundadas técnicas como a criação de redes de computadores, aplicações de download, funcionamento do router e switch.
O relatório também nos permitiu fazer a ponte entre a esfera prática e teórica, de modo que conseguimos perceber melhor como ambas funcionam.
Conclui-se que o projeto correu com sucesso, uma vez que os erros previstos foram suportados, os objetivos de cada experiência foram cumpridos e o ficheiro foi descarregado com sucesso.

Anexos - Código Fonte

main.c

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

#define IP_MAX_SIZE 16

#define SERVER_PORT 21

struct url{
    char* username;
    char* password;
    char* host;
    char* urlPath;

};
static char filename[500];
```

```c
int parser(int argc, char** argv, struct url *url){
    //ftp://[<user>:<password>@]<host>/<url-path>
    if (argc < 2){
        printf("Not enought arguments\n");
        return -1;
    }

    char *url_string;
    /* get initial */
    url_string = strtok(argv[1], "//");

    printf("ftp : %s\n", url_string);

    if (strcmp(url_string, "ftp:")!=0) {
        printf("Usage: ftp://[<user>:<password>@]<host>/<url-path>\n");
        return -1;
    }
    url_string = strtok(NULL, "\0");

    //get host with user:password
    char *up_host;
    up_host = strtok(url_string, "/");
    printf("up_host : %s\n", up_host);

    //get url
    url->urlPath = strtok(NULL, "\0");
    printf("url : %s\n", url->urlPath);

    char copy[500];

    strcpy(copy,url->urlPath);
    char* url_string2 = strtok(copy, "/");
```

```
while( url_string2 != NULL ) {
    strcpy(filename,url_string2);
    url_string2 = strtok(NULL, "/");
}

//user:password@host
if (strstr(up_host, "@") != NULL) {
    if (strstr(up_host, ":") == NULL) {
        printf("Usage: ftp://[<user>:<password>@]<host>/<url-path>\n");
        return -1;
    }

    //user
    url->username = strtok(up_host, ":");
    if (strstr(url->username, "@") != NULL) {
        printf("Usage: ftp://[<user>:<password>@]<host>/<url-path>\n");
        return -1;
    }
    printf("user : %s\n", url->username);
    up_host = strtok(NULL, "\0");

    //password
    if (up_host[0] != '@') {
        url->password = strtok(up_host, "@");
        printf("password : %s\n", url->password);
        up_host = strtok(NULL, "\0");
    } else {
        printf("Usage: ftp://[<user>:<password>@]<host>/<url-path>\n");
        return -1;
    }

    //host
    url->host = strtok(up_host, "\0");
    printf("host : %s\n", url->host);
```

```
    } else {
        //host
        url->username = "anonymous";
        url->password = "";
        url->host = strtok(up_host, "\0");
        printf("host : %s\n", url->host);
    }

    return 0;
}

int writeSocket(int fd, char* cmd) {

    int bytesSent;

    if ((bytesSent = write(fd, cmd, strlen(cmd))) != strlen(cmd)) {
        fprintf(stderr, "Error writing message to socket!\n");
        return -1;
    }

    printf("Command: %s", cmd);

    return bytesSent;
}
```

```c
int readSocket(char* response, FILE* socket){
    int H, T, U, space, responseCode;

    H = fgetc(socket);
    T = fgetc(socket);
    U = fgetc(socket);
    space = fgetc(socket);

    responseCode = (H-'0')*100 + (T-'0')*10 + U-'0';

    if(space != ' '){
        fgets(response, 200, socket);
        while(response[3] != ' ') {
            fgets(response, 200, socket);
        }
    } else {
        fgets(response, 200, socket);
    }

    printf("\n%s\n", response);

    return responseCode;
}
```

```c
int createSocket(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);
    server_addr.sin_port = htons(port);

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

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

    return sockfd;
}
```

```c
int manageSocket(char* cmd, char* arg1, char* arg2, int sockfd1, FILE* sockfile1, char* buf, int code){

    int responseCode;

    free(cmd);
    cmd = malloc(256);
    strcat(cmd, arg1);
    strcat(cmd, arg2);
    strcat(cmd, "\n");

    if(writeSocket(sockfd1, cmd) < 0) {
        printf("Error writing to socket");
        return -1;
    }

    if((responseCode = readSocket(buf, sockfile1)) < 0){
        printf("Error reading socket\n");
        return -1;
    }

    if(responseCode != code){
        printf("Bad response\n");
        return -1;
    }
}

int getPassivePort(int *pasv_port, char* buf){

    int a,b,c,d,e,f;
    sscanf(buf, "Entering Passive Mode (%d,%d,%d,%d,%d,%d).", &a,&b,&c,&d,&e,&f);

    *pasv_port = e * 256 + f;

    return 0;
}
```

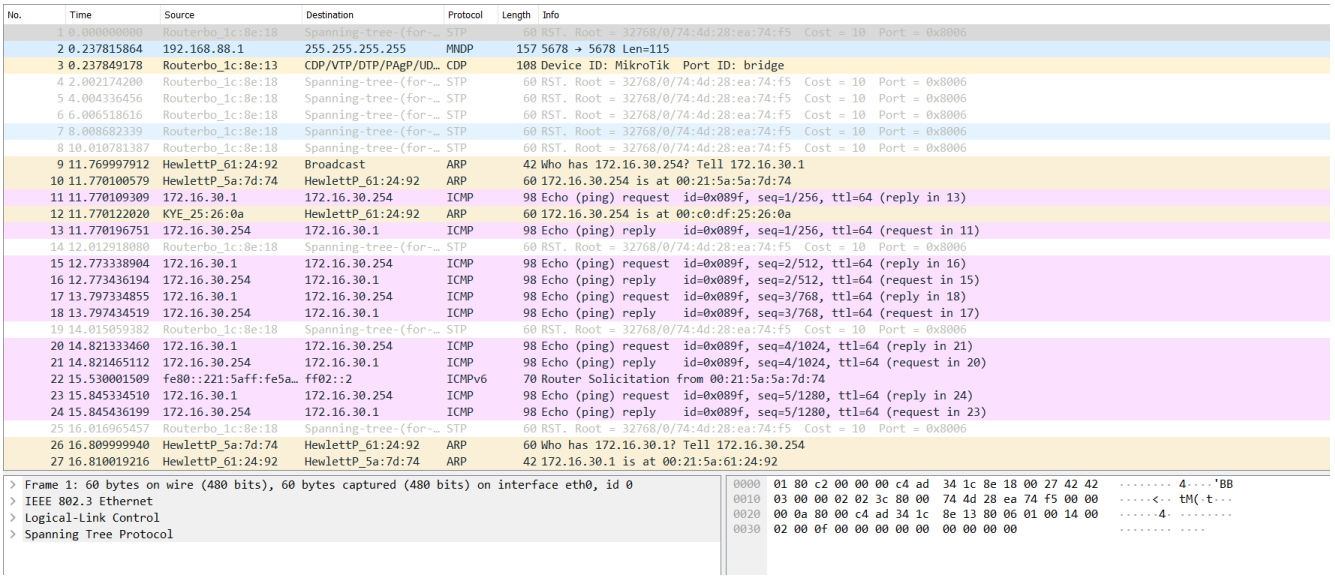
```c
int writeToFile(FILE* socket_file2){

    FILE* myFile = fopen(filename, "w");

    int s;
    while(true){
        unsigned char content[300];
        bool end;
        s = fread(content, 1, 300, socket_file2);
        if(s < 300) end = true;
        s = fwrite(content, 1, s, myFile);
        if(end) break;
    }
}


int main(int argc, char** argv)
{
    struct url url;
    if(parser(argc, argv, &url) < 0) {
        return -1;
    }

    struct hostent *h;
    if ((h = gethostbyname(url.host)) == NULL) {
        herror("gethostbyname()");
        return -1;
    }

    char ip[IP_MAX_SIZE];
    strcpy(ip, inet_ntoa(*((struct in_addr *) h->h_addr)));
    printf("IP Address : %s\n", ip);
```

```
int sockfd1;
if ((sockfd1 = createSocket(ip, SERVER_PORT)) < 0) {
    return -1;
}

FILE * sockfile1 = fdopen(sockfd1, "r+");

char buf[1000];
int responseCode;

//welcome message
if((responseCode = readSocket(buf, sockfile1)) < 0){
    printf("Error reading socket\n");
    return -1;
}

if(responseCode != 220){
    printf("Bad response\n");
    return -1;
}

char* cmd = malloc(256);

//username
manageSocket(cmd, "user ", url.username, sockfd1, sockfile1, buf, 331);

//password
manageSocket(cmd, "pass ", url.password, sockfd1, sockfile1, buf, 230);

//passive mode
manageSocket(cmd, "pasv ", "", sockfd1, sockfile1, buf, 227);

int pasv_port;

getPassivePort(&pasv_port, buf);
```

```
    int sockfd2;

    if ((sockfd2 = createSocket(ip, pasv_port)) < 0) {
        return -1;
    }

    FILE * socket_file2 = fdopen(sockfd2, "r+");

    //retrieve command
    manageSocket(cmd, "retr ", url.urlPath, sockfd1, sockfile1, buf, 150);

    //Write to file
    writeToFile(socket_file2);

    //Read transfer complete
    if((responseCode = readSocket(buf, sockfile1)) < 0){
        printf("Error reading socket\n");
        return -1;
    }

    if(responseCode != 226){
        printf("Bad response\n");
        return -1;
    }

    if (close(sockfd2)<0) {
        perror("close()");
        return -1;
    }

    if (close(sockfd1)<0) {
        perror("close()");
        return -1;
    }
    return 0;
}
```

Logs

Experiência 1:

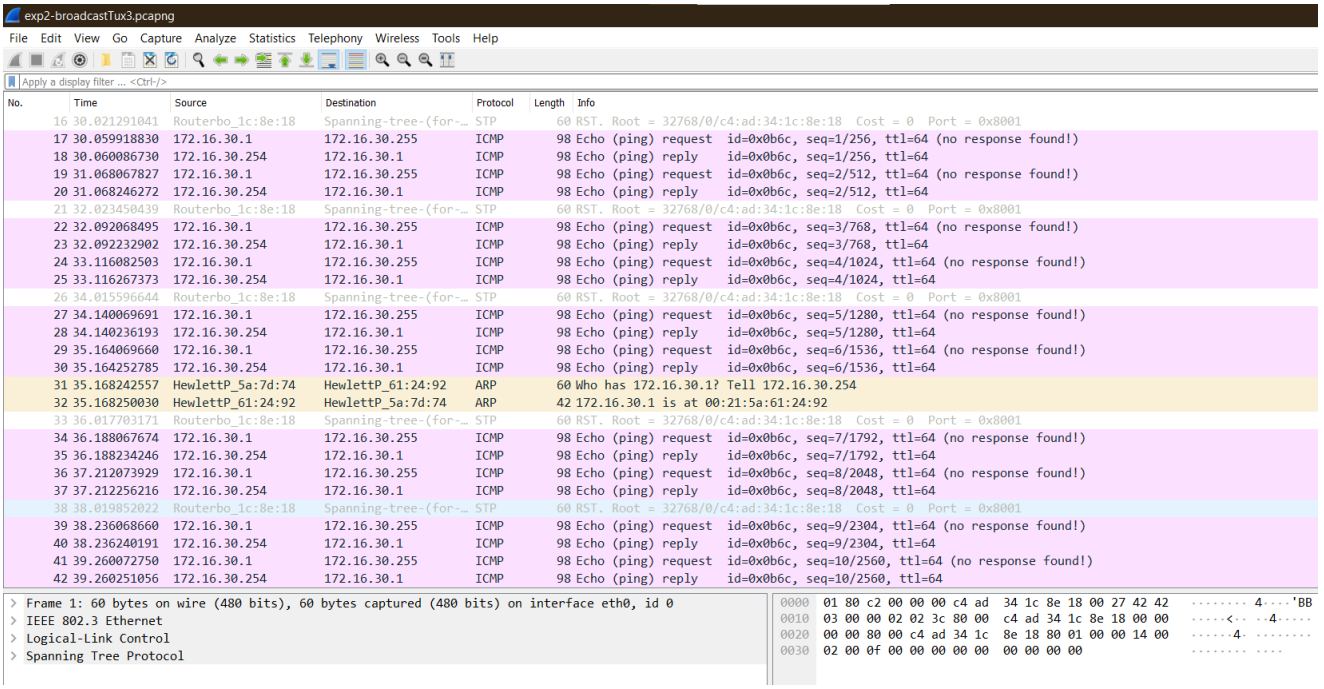


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:ea:74:f5 Cost = 10 Port = 0x8006
2	0.237815864	192.168.88.1	255.255.255.255	MNDP	157	5678 → 5678 Len=115
3	0.237849178	Routerbo_1c:8e:13	CDP/VTP/DTP/PAgP/UD…	CDP	108	Device ID: MikroTik Port ID: bridge
4	2.002174200	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:ea:74:f5 Cost = 10 Port = 0x8006
5	4.004336456	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:ea:74:f5 Cost = 10 Port = 0x8006
6	6.006518616	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:ea:74:f5 Cost = 10 Port = 0x8006
7	8.008682339	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:ea:74:f5 Cost = 10 Port = 0x8006
8	10.010781387	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:ea:74:f5 Cost = 10 Port = 0x8006
9	11.769997912	HewlettP_61:24:92	Broadcast	ARP	42	Who has 172.16.30.254? Tell 172.16.30.1
10	11.770100579	HewlettP_5a:7d:74	HewlettP_61:24:92	ARP	60	172.16.30.254 is at 00:21:5a:5a:7d:74
11	11.770109309	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x089f, seq=1/256, ttl=64 (reply in 13)
12	11.770122020	KYE_25:26:0a	HewlettP_61:24:92	ARP	60	172.16.30.254 is at 00:c0:df:25:26:0a
13	11.770196751	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x089f, seq=1/256, ttl=64 (request in 11)
14	12.012918080	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:ea:74:f5 Cost = 10 Port = 0x8006
15	12.773338904	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x089f, seq=2/512, ttl=64 (reply in 16)
16	12.773436194	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x089f, seq=2/512, ttl=64 (request in 15)
17	13.797334855	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x089f, seq=3/768, ttl=64 (reply in 18)
18	13.797434519	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x089f, seq=3/768, ttl=64 (request in 17)
19	14.015059382	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:ea:74:f5 Cost = 10 Port = 0x8006
20	14.821333460	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x089f, seq=4/1024, ttl=64 (reply in 21)
21	14.821465112	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x089f, seq=4/1024, ttl=64 (request in 20)
22	15.530001509	fe80::221:5aff:fe5a…	ff02::2	ICMPv6	70	Router Solicitation from 00:21:5a:5a:7d:74
23	15.845334510	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x089f, seq=5/1280, ttl=64 (reply in 24)
24	15.845436199	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x089f, seq=5/1280, ttl=64 (request in 23)
25	16.016965457	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:ea:74:f5 Cost = 10 Port = 0x8006
26	16.809999940	HewlettP_5a:7d:74	HewlettP_61:24:92	ARP	60	Who has 172.16.30.1? Tell 172.16.30.254
27	16.810019216	HewlettP_61:24:92	HewlettP_5a:7d:74	ARP	42	172.16.30.1 is at 00:21:5a:61:24:92

```
> Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0
> IEEE 802.3 Ethernet
> Logical-Link Control
> Spanning Tree Protocol
```

```
0000  01 80 c2 00 00 00 c4 ad  34 1c 8e 18 00 27 42 42   ........ 4....'BB
0010  03 00 00 02 02 3c 80 00  74 4d 28 ea 74 f5 00 00   .....<.. tM(.t...
0020  00 0a 80 00 c4 ad 34 1c  8e 13 80 06 01 00 14 00   ......4. ........
0030  02 00 0f 00 00 00 00 00  00 00 00 00               ........ ....
```

Experiência 2 - Broadcast do tux 3:



exp2-broadcastTux3.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

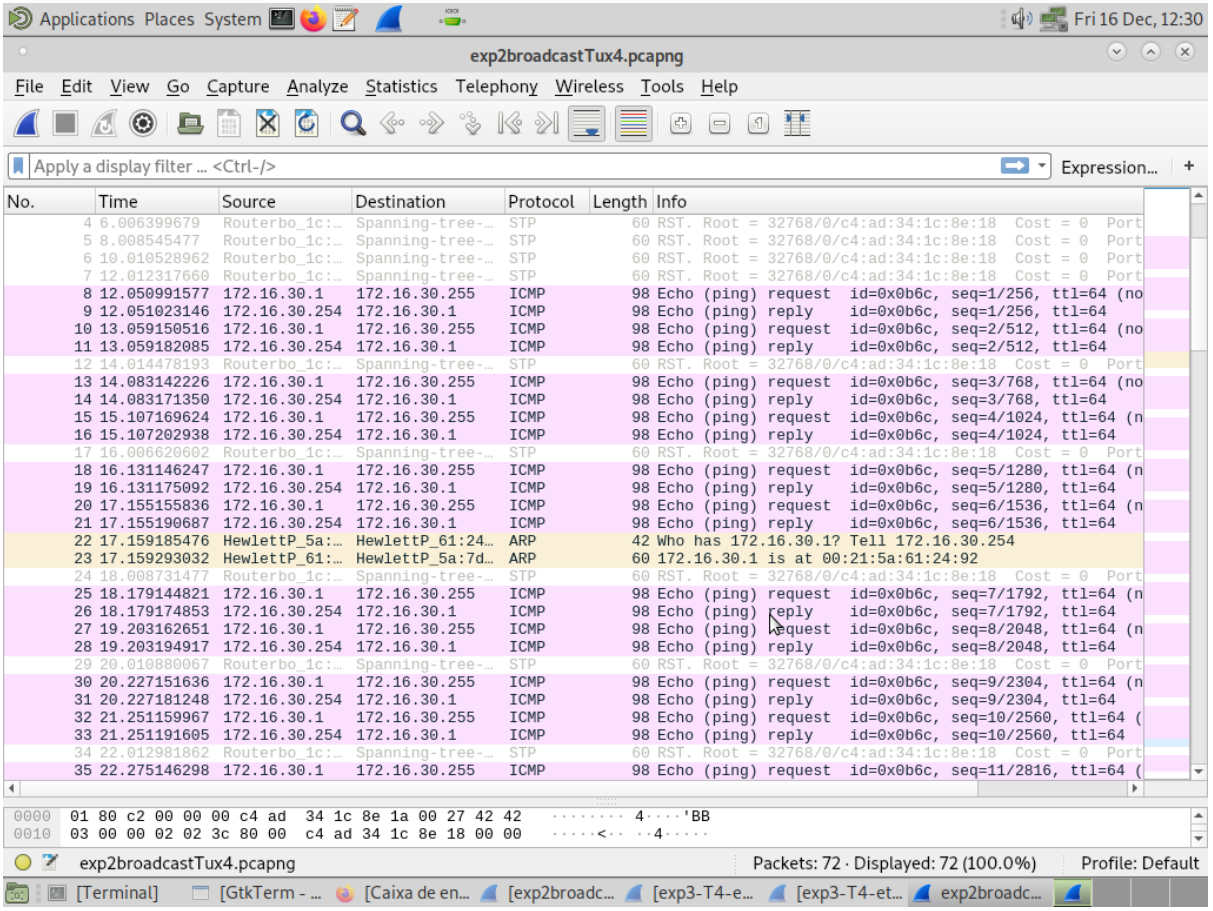
Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
16	30.021291041	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
17	30.059918830	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x0b6c, seq=1/256, ttl=64 (no response found!)
18	30.060086730	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0b6c, seq=1/256, ttl=64
19	31.068067827	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x0b6c, seq=2/512, ttl=64 (no response found!)
20	31.068246272	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0b6c, seq=2/512, ttl=64
21	32.023450439	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
22	32.092068495	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x0b6c, seq=3/768, ttl=64 (no response found!)
23	32.092232902	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0b6c, seq=3/768, ttl=64
24	33.116082503	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x0b6c, seq=4/1024, ttl=64 (no response found!)
25	33.116267373	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0b6c, seq=4/1024, ttl=64
26	34.015596644	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
27	34.140069691	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x0b6c, seq=5/1280, ttl=64 (no response found!)
28	34.140236193	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0b6c, seq=5/1280, ttl=64
29	35.164069660	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x0b6c, seq=6/1536, ttl=64 (no response found!)
30	35.164252785	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0b6c, seq=6/1536, ttl=64
31	35.168242557	HewlettP_5a:7d:74	HewlettP_61:24:92	ARP	60	Who has 172.16.30.1? Tell 172.16.30.254
32	35.168250030	HewlettP_61:24:92	HewlettP_5a:7d:74	ARP	42	172.16.30.1 is at 00:21:5a:61:24:92
33	36.017703171	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
34	36.188067674	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x0b6c, seq=7/1792, ttl=64 (no response found!)
35	36.188234246	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0b6c, seq=7/1792, ttl=64
36	37.212073929	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x0b6c, seq=8/2048, ttl=64 (no response found!)
37	37.212256216	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0b6c, seq=8/2048, ttl=64
38	38.019852022	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
39	38.236068660	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x0b6c, seq=9/2304, ttl=64 (no response found!)
40	38.236240191	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0b6c, seq=9/2304, ttl=64
41	39.260072750	172.16.30.1	172.16.30.255	ICMP	98	Echo (ping) request id=0x0b6c, seq=10/2560, ttl=64 (no response found!)
42	39.260251056	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0b6c, seq=10/2560, ttl=64

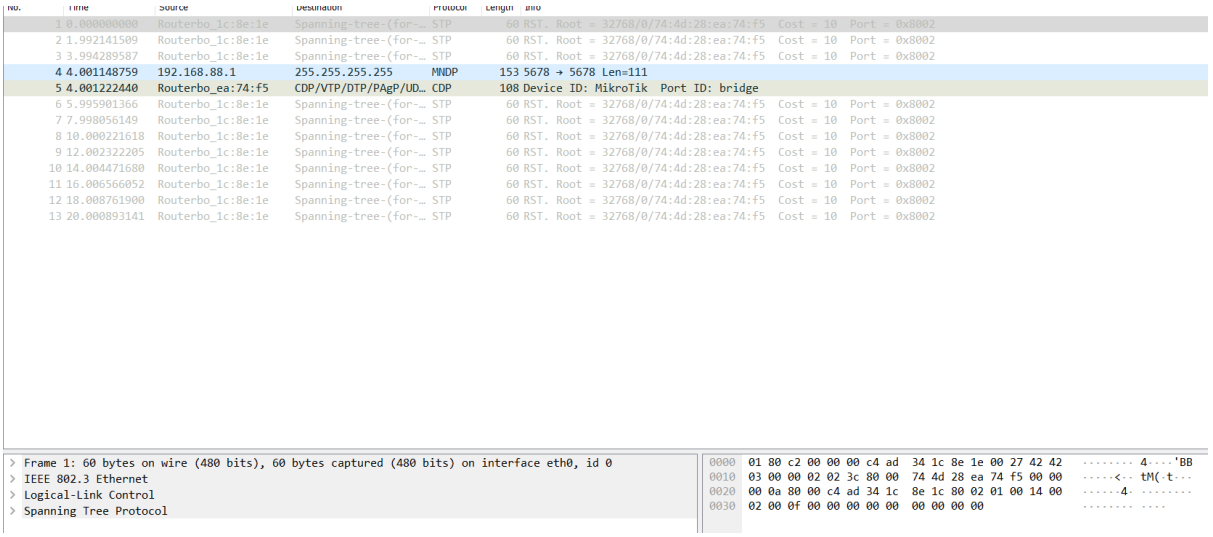
```
> Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0
> IEEE 802.3 Ethernet
> Logical-Link Control
> Spanning Tree Protocol
```

```
0000  01 80 c2 00 00 00 c4 ad  34 1c 8e 18 00 27 42 42   ........ 4....'BB
0010  03 00 00 02 02 3c 80 00  c4 ad 34 1c 8e 18 00 00   .....<.. ..4.....
0020  00 00 80 00 c4 ad 34 1c  8e 18 80 01 00 00 14 00   ......4. ........
0030  02 00 0f 00 00 00 00 00  00 00 00 00               ........ ....
```

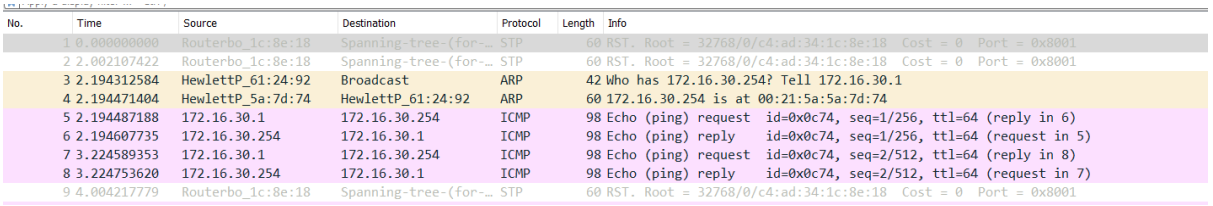
Experiência 2 - Broadcast do tux 4:



Experiência 2 - Broadcast do tux 2:

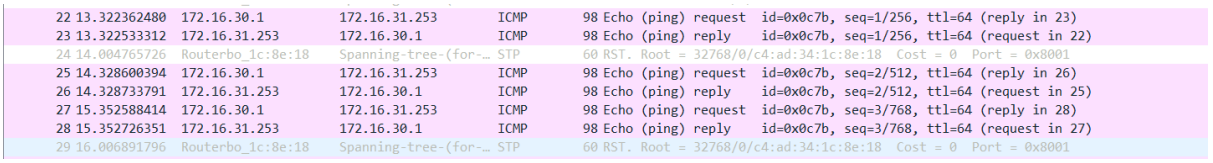


Experiência 3 - Tux 3 para Tux 4 eth0:



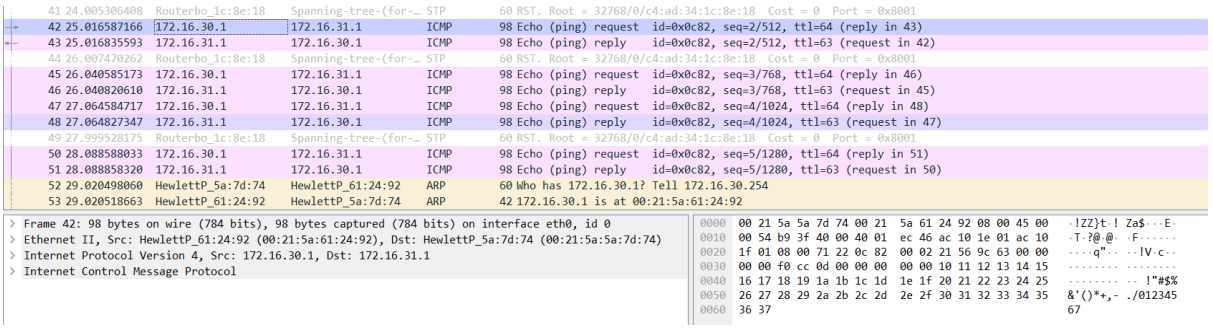
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
2	2.002107422	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
3	2.194312584	HewlettP_61:24:92	Broadcast	ARP	42	Who has 172.16.30.254? Tell 172.16.30.1
4	2.194471404	HewlettP_5a:7d:74	HewlettP_61:24:92	ARP	60	172.16.30.254 is at 00:21:5a:5a:7d:74
5	2.194487188	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x0c74, seq=1/256, ttl=64 (reply in 6)
6	2.194607735	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0c74, seq=1/256, ttl=64 (request in 5)
7	3.224589353	172.16.30.1	172.16.30.254	ICMP	98	Echo (ping) request id=0x0c74, seq=2/512, ttl=64 (reply in 8)
8	3.224753620	172.16.30.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0c74, seq=2/512, ttl=64 (request in 7)
9	4.004217779	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001

Experiência 3 - Tux 3 para Tux 4 eth1:



22	13.322362480	172.16.30.1	172.16.31.253	ICMP	98	Echo (ping) request id=0x0c7b, seq=1/256, ttl=64 (reply in 23)
23	13.322533312	172.16.31.253	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0c7b, seq=1/256, ttl=64 (request in 22)
24	14.004765726	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
25	14.328600394	172.16.30.1	172.16.31.253	ICMP	98	Echo (ping) request id=0x0c7b, seq=2/512, ttl=64 (reply in 26)
26	14.328733791	172.16.31.253	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0c7b, seq=2/512, ttl=64 (request in 25)
27	15.352588414	172.16.30.1	172.16.31.253	ICMP	98	Echo (ping) request id=0x0c7b, seq=3/768, ttl=64 (reply in 28)
28	15.352726351	172.16.31.253	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0c7b, seq=3/768, ttl=64 (request in 27)
29	16.006891796	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001

Experiência 3 - Tux 3 para Tux 2:

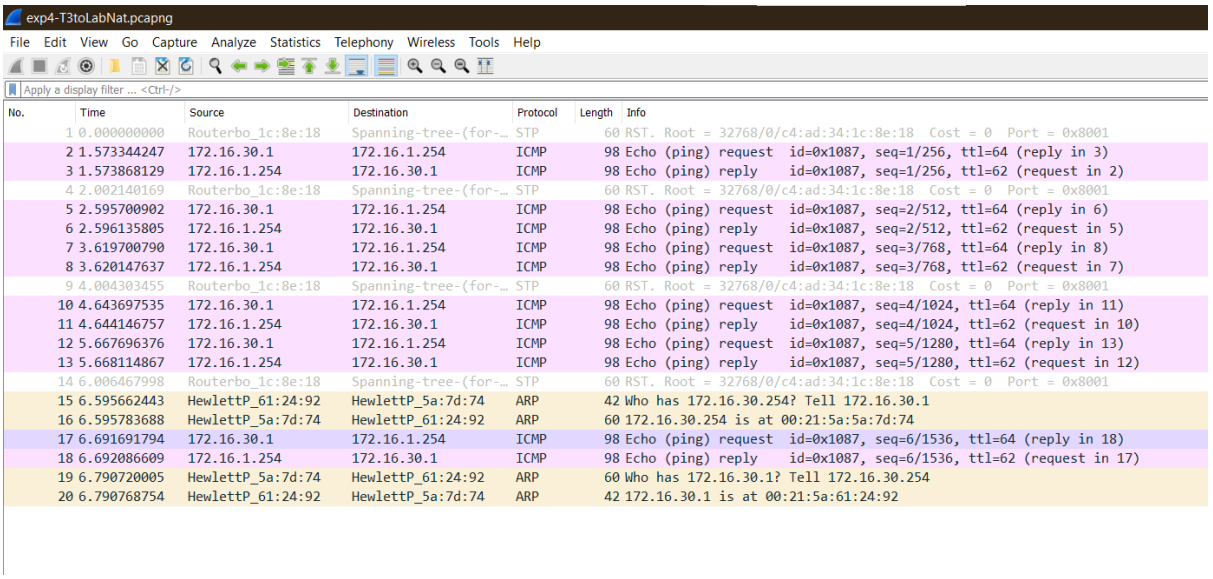


41	24.005306408	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
42	25.016587166	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) request id=0x0c82, seq=2/512, ttl=64 (reply in 43)
43	25.016835593	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0c82, seq=2/512, ttl=63 (request in 42)
44	26.007470262	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
45	26.040585173	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) request id=0x0c82, seq=3/768, ttl=64 (reply in 46)
46	26.040820610	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0c82, seq=3/768, ttl=63 (request in 45)
47	27.064584717	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) request id=0x0c82, seq=4/1024, ttl=64 (reply in 48)
48	27.064827347	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0c82, seq=4/1024, ttl=63 (request in 47)
49	27.999528175	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
50	28.088588033	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) request id=0x0c82, seq=5/1280, ttl=64 (reply in 51)
51	28.088858320	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0c82, seq=5/1280, ttl=63 (request in 50)
52	29.020498060	HewlettP_5a:7d:74	HewlettP_61:24:92	ARP	60	Who has 172.16.30.1? Tell 172.16.30.254
53	29.020518663	HewlettP_61:24:92	HewlettP_5a:7d:74	ARP	42	172.16.30.1 is at 00:21:5a:61:24:92

```
> Frame 42: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth0, id 0
> Ethernet II, Src: HewlettP_61:24:92 (00:21:5a:61:24:92), Dst: HewlettP_5a:7d:74 (00:21:5a:5a:7d:74)
> Internet Protocol Version 4, Src: 172.16.30.1, Dst: 172.16.31.1
> Internet Control Message Protocol
```

```
0000  00 21 5a 5a 7d 74 00 21  5a 61 24 92 08 00 45 00   ·!ZZ}t.! Za$···E·
0010  00 54 b9 3f 40 00 40 01  ec 46 ac 10 1e 01 ac 10   ·T·?@·@· ·F······
0020  1f 01 08 00 71 22 0c 82  00 02 21 56 9c 63 00 00   ····q"·· ··!V·c··
0030  00 00 f0 cc 0d 00 00 00  00 00 10 11 12 13 14 15   ········ ········
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ········ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                             67
```

Experiência 4 - Tux 3 para o lab, NAT enable:

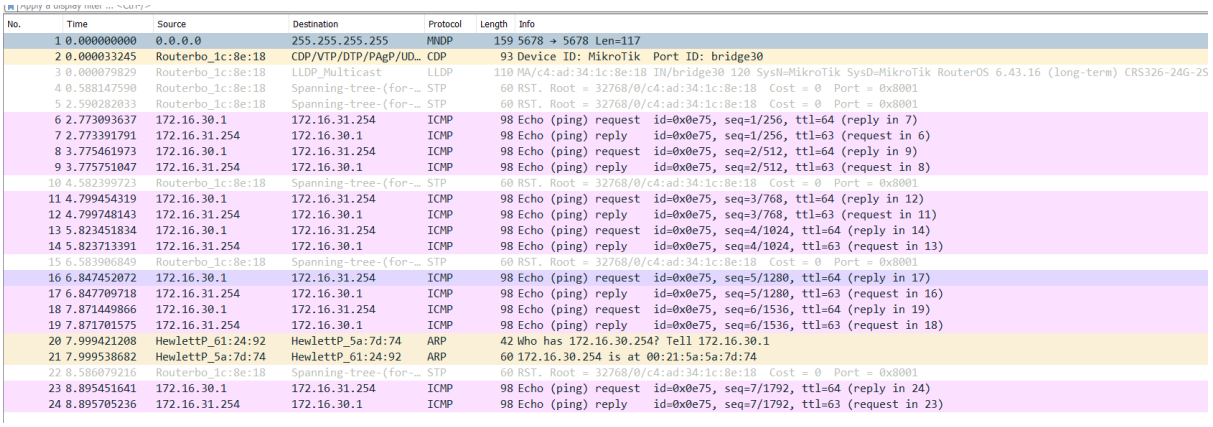


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
2	1.573344247	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x1087, seq=1/256, ttl=64 (reply in 3)
3	1.573868129	172.16.1.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x1087, seq=1/256, ttl=62 (request in 2)
4	2.002140169	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
5	2.595700902	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x1087, seq=2/512, ttl=64 (reply in 6)
6	2.596135805	172.16.1.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x1087, seq=2/512, ttl=62 (request in 5)
7	3.619700790	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x1087, seq=3/768, ttl=64 (reply in 8)
8	3.620147637	172.16.1.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x1087, seq=3/768, ttl=62 (request in 7)
9	4.004303455	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
10	4.643697535	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x1087, seq=4/1024, ttl=64 (reply in 11)
11	4.644146757	172.16.1.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x1087, seq=4/1024, ttl=62 (request in 10)
12	5.667696376	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x1087, seq=5/1280, ttl=64 (reply in 13)
13	5.668114867	172.16.1.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x1087, seq=5/1280, ttl=62 (request in 12)
14	6.006467998	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
15	6.595662443	HewlettP_61:24:92	HewlettP_5a:7d:74	ARP	42	Who has 172.16.30.254? Tell 172.16.30.1
16	6.595783688	HewlettP_5a:7d:74	HewlettP_61:24:92	ARP	60	172.16.30.254 is at 00:21:5a:5a:7d:74
17	6.691691794	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x1087, seq=6/1536, ttl=64 (reply in 18)
18	6.692086609	172.16.1.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x1087, seq=6/1536, ttl=62 (request in 17)
19	6.790720005	HewlettP_5a:7d:74	HewlettP_61:24:92	ARP	60	Who has 172.16.30.1? Tell 172.16.30.254
20	6.790768754	HewlettP_61:24:92	HewlettP_5a:7d:74	ARP	42	172.16.30.1 is at 00:21:5a:61:24:92

Experiência 4 - Tux 3 para o lab, NAT disable:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
2	2.002096867	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
3	4.004201207	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
4	6.006315465	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
5	8.008475119	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
6	10.010646088	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
7	11.392375545	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x0fec, seq=1/256, ttl=64 (no response found!)
8	12.012794987	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
9	12.400280644	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x0fec, seq=2/512, ttl=64 (no response found!)
10	13.424289542	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x0fec, seq=3/768, ttl=64 (no response found!)
11	14.014964001	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
12	14.448261284	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x0fec, seq=4/1024, ttl=64 (no response found!)
13	15.472261313	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x0fec, seq=5/1280, ttl=64 (no response found!)
14	16.017112201	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
15	16.496244858	HewlettP_61:24:92	HewlettP_5a:7d:74	ARP	42	Who has 172.16.30.254? Tell 172.16.30.1
16	16.496285576	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x0fec, seq=6/1536, ttl=64 (no response found!)
17	16.496384611	HewlettP_5a:7d:74	HewlettP_61:24:92	ARP	60	172.16.30.254 is at 00:21:5a:5a:7d:74
18	17.520260461	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x0fec, seq=7/1792, ttl=64 (no response found!)
19	18.019269551	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
20	18.544282838	172.16.30.1	172.16.1.254	ICMP	98	Echo (ping) request id=0x0fec, seq=8/2048, ttl=64 (no response found!)
21	20.021413561	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
22	22.023578663	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001

Experiência 4 - Tux 3 para Rc:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	0.0.0.0	255.255.255.255	MNDP	159	5678 → 5678 Len=117
2	0.000033245	Routerbo_1c:8e:18	CDP/VTP/DTP/PAgP/UD…	CDP	93	Device ID: MikroTik Port ID: bridge30
3	0.000079829	Routerbo_1c:8e:18	LLDP_Multicast	LLDP	110	MA/c4:ad:34:1c:8e:18 IN/bridge30 120 SysN=MikroTik SysD=MikroTik RouterOS 6.43.16 (long-term) CRS326-24G-2S
4	0.588147590	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
5	2.590282033	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
6	2.773093637	172.16.30.1	172.16.31.254	ICMP	98	Echo (ping) request id=0x0e75, seq=1/256, ttl=64 (reply in 7)
7	2.773391791	172.16.31.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0e75, seq=1/256, ttl=63 (request in 6)
8	3.775461973	172.16.30.1	172.16.31.254	ICMP	98	Echo (ping) request id=0x0e75, seq=2/512, ttl=64 (reply in 9)
9	3.775751047	172.16.31.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0e75, seq=2/512, ttl=63 (request in 8)
10	4.582399723	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
11	4.799454319	172.16.30.1	172.16.31.254	ICMP	98	Echo (ping) request id=0x0e75, seq=3/768, ttl=64 (reply in 12)
12	4.799748143	172.16.31.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0e75, seq=3/768, ttl=63 (request in 11)
13	5.823451834	172.16.30.1	172.16.31.254	ICMP	98	Echo (ping) request id=0x0e75, seq=4/1024, ttl=64 (reply in 14)
14	5.823713391	172.16.31.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0e75, seq=4/1024, ttl=63 (request in 13)
15	6.583906849	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
16	6.847452072	172.16.30.1	172.16.31.254	ICMP	98	Echo (ping) request id=0x0e75, seq=5/1280, ttl=64 (reply in 17)
17	6.847709718	172.16.31.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0e75, seq=5/1280, ttl=63 (request in 16)
18	7.871449866	172.16.30.1	172.16.31.254	ICMP	98	Echo (ping) request id=0x0e75, seq=6/1536, ttl=64 (reply in 19)
19	7.871701575	172.16.31.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0e75, seq=6/1536, ttl=63 (request in 18)
20	7.999421208	HewlettP_61:24:92	HewlettP_5a:7d:74	ARP	42	Who has 172.16.30.254? Tell 172.16.30.1
21	7.999538682	HewlettP_5a:7d:74	HewlettP_61:24:92	ARP	60	172.16.30.254 is at 00:21:5a:5a:7d:74
22	8.586079216	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
23	8.895451641	172.16.30.1	172.16.31.254	ICMP	98	Echo (ping) request id=0x0e75, seq=7/1792, ttl=64 (reply in 24)
24	8.895705236	172.16.31.254	172.16.30.1	ICMP	98	Echo (ping) reply id=0x0e75, seq=7/1792, ttl=63 (request in 23)

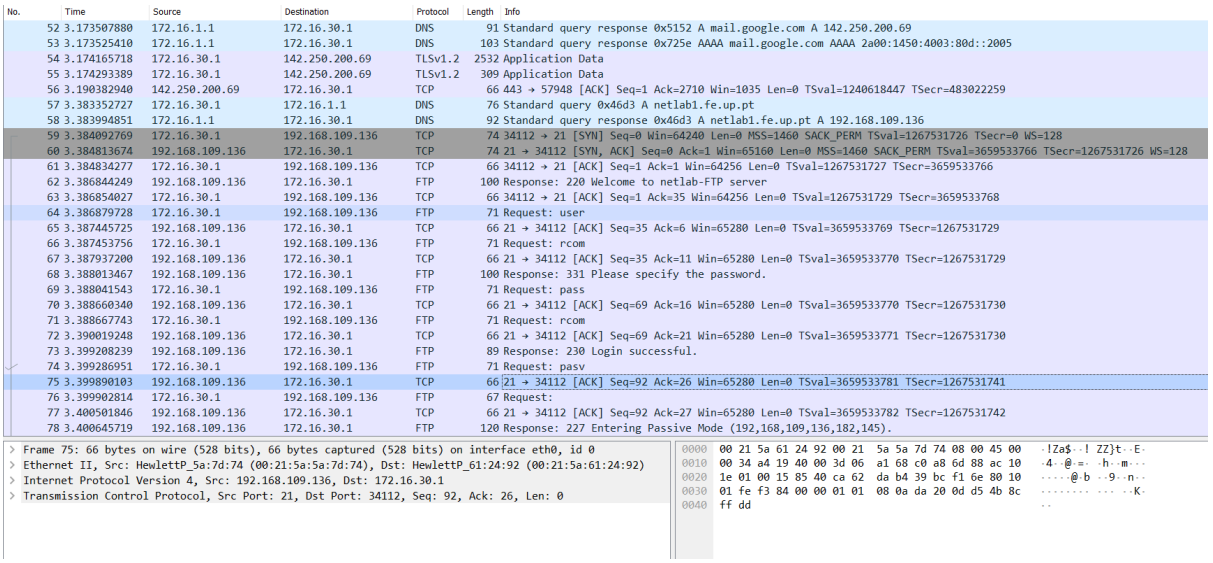
Experiência 4 - Tux 2 para Tux 3 com redirect:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	Routerbo_1c:8e:1e	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:ea:74:f5 Cost = 10 Port = 0x8002
2	2.001270388	Routerbo_1c:8e:1e	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:ea:74:f5 Cost = 10 Port = 0x8002
3	4.003429905	Routerbo_1c:8e:1e	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:ea:74:f5 Cost = 10 Port = 0x8002
4	6.005592215	Routerbo_1c:8e:1e	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:ea:74:f5 Cost = 10 Port = 0x8002
5	6.088661019	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x7a1e, seq=1/256, ttl=64 (reply in 6)
6	6.089079778	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) reply id=0x7a1e, seq=1/256, ttl=63 (request in 5)
7	7.114202606	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x7a1e, seq=2/512, ttl=64 (reply in 9)
8	7.116069423	172.16.31.254	172.16.31.1	ICMP	126	Redirect (Redirect for host)
9	7.116307367	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) reply id=0x7a1e, seq=2/512, ttl=63 (request in 7)
10	8.007756132	Routerbo_1c:8e:1e	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:ea:74:f5 Cost = 10 Port = 0x8002
11	8.115379826	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x7a1e, seq=3/768, ttl=64 (reply in 13)
12	8.115531728	172.16.31.254	172.16.31.1	ICMP	126	Redirect (Redirect for host)
13	8.115737196	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) reply id=0x7a1e, seq=3/768, ttl=63 (request in 11)
14	9.130202617	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x7a1e, seq=4/1024, ttl=64 (reply in 16)
15	9.130357382	172.16.31.254	172.16.31.1	ICMP	126	Redirect (Redirect for host)
16	9.130562152	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) reply id=0x7a1e, seq=4/1024, ttl=63 (request in 14)
17	10.009894139	Routerbo_1c:8e:1e	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:ea:74:f5 Cost = 10 Port = 0x8002
18	10.154198162	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x7a1e, seq=5/1280, ttl=64 (reply in 20)
19	10.154341822	172.16.31.254	172.16.31.1	ICMP	126	Redirect (Redirect for host)
20	10.154558814	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) reply id=0x7a1e, seq=5/1280, ttl=63 (request in 18)
21	11.178196709	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x7a1e, seq=6/1536, ttl=64 (reply in 23)
22	11.178337925	172.16.31.254	172.16.31.1	ICMP	126	Redirect (Redirect for host)
23	11.178552892	172.16.30.1	172.16.31.1	ICMP	98	Echo (ping) reply id=0x7a1e, seq=6/1536, ttl=63 (request in 21)
24	11.306165563	HewlettP_61:30:63	Routerbo_ea:74:f5	ARP	42	Who has 172.16.31.254? Tell 172.16.31.1
25	11.306260126	Routerbo_ea:74:f5	HewlettP_61:30:63	ARP	60	172.16.31.254 is at 74:4d:28:ea:74:f5
26	12.011997016	Routerbo_1c:8e:1e	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:ea:74:f5 Cost = 10 Port = 0x8002
27	12.202199308	172.16.31.1	172.16.30.1	ICMP	98	Echo (ping) request id=0x7a1e, seq=7/1792, ttl=64 (reply in 28)

Experiência 5 - Tux 3 para Google:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
2	2.002123826	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
3	2.422471217	172.16.30.1	172.16.1.1	DNS	70	Standard query 0x5d1b A google.com
4	2.422481694	172.16.30.1	172.16.1.1	DNS	70	Standard query 0x6a24 AAAA google.com
5	2.423145887	172.16.1.1	172.16.30.1	DNS	86	Standard query response 0x5d1b A google.com A 142.250.200.78
6	2.469630787	172.16.1.1	172.16.30.1	DNS	98	Standard query response 0x6a24 AAAA google.com AAAA 2a00:1450:4003:80d::200e
7	2.480329055	172.16.30.1	142.250.200.78	ICMP	98	Echo (ping) request id=0x111d, seq=1/256, ttl=64 (reply in 8)
8	2.497941014	142.250.200.78	172.16.30.1	ICMP	98	Echo (ping) reply id=0x111d, seq=1/256, ttl=112 (request in 7)
9	2.498037256	172.16.30.1	172.16.1.1	DNS	87	Standard query 0x3710 PTR 78.200.250.142.in-addr.arpa
10	2.498882829	172.16.1.1	172.16.30.1	DNS	126	Standard query response 0x3710 PTR 78.200.250.142.in-addr.arpa PTR mad07s24-in-f14.1e100.net
11	2.576122202	172.16.30.1	172.16.1.1	DNS	86	Standard query 0x8cab PTR 82.221.107.34.in-addr.arpa
12	2.576997807	172.16.1.1	172.16.30.1	DNS	138	Standard query response 0x8cab PTR 82.221.107.34.in-addr.arpa PTR 82.221.107.34.bc.googleusercontent.com
13	2.577196785	172.16.30.1	172.16.1.1	DNS	86	Standard query 0x160f PTR 82.221.107.34.in-addr.arpa
14	2.577850992	172.16.1.1	172.16.30.1	DNS	138	Standard query response 0x160f PTR 82.221.107.34.in-addr.arpa PTR 82.221.107.34.bc.googleusercontent.com
15	2.577964973	172.16.30.1	172.16.1.1	DNS	86	Standard query 0x9221 PTR 26.112.82.140.in-addr.arpa
16	2.578415731	172.16.1.1	172.16.30.1	DNS	131	Standard query response 0x9221 PTR 26.112.82.140.in-addr.arpa PTR lb-140-82-112-26-iad.github.com
17	2.578542075	172.16.30.1	172.16.1.1	DNS	85	Standard query 0x092b PTR 4.121.82.140.in-addr.arpa
18	2.578951137	172.16.1.1	172.16.30.1	DNS	129	Standard query response 0x092b PTR 4.121.82.140.in-addr.arpa PTR lb-140-82-121-4-fra.github.com
19	3.481963914	172.16.30.1	142.250.200.78	ICMP	98	Echo (ping) request id=0x111d, seq=2/512, ttl=64 (reply in 20)
20	3.498179460	142.250.200.78	172.16.30.1	ICMP	98	Echo (ping) reply id=0x111d, seq=2/512, ttl=112 (request in 19)
21	3.887138774	172.16.30.1	140.82.121.4	TLSv1.2	105	Application Data
22	3.934055575	140.82.121.4	172.16.30.1	TCP	66	443 → 39458 [ACK] Seq=1 Ack=40 Win=72 Len=0 TSval=217065680 TSecr=791846841
23	3.934071918	140.82.121.4	172.16.30.1	TLSv1.2	105	Application Data
24	3.934080439	172.16.30.1	140.82.121.4	TCP	66	39458 → 443 [ACK] Seq=40 Ack=40 Win=501 Len=0 TSval=791846888 TSecr=217065680
25	4.004187447	Routerbo_1c:8e:18	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/c4:ad:34:1c:8e:18 Cost = 0 Port = 0x8001
26	4.483271285	172.16.30.1	142.250.200.78	ICMP	98	Echo (ping) request id=0x111d, seq=3/768, ttl=64 (reply in 27)

Experiência 6 - Download

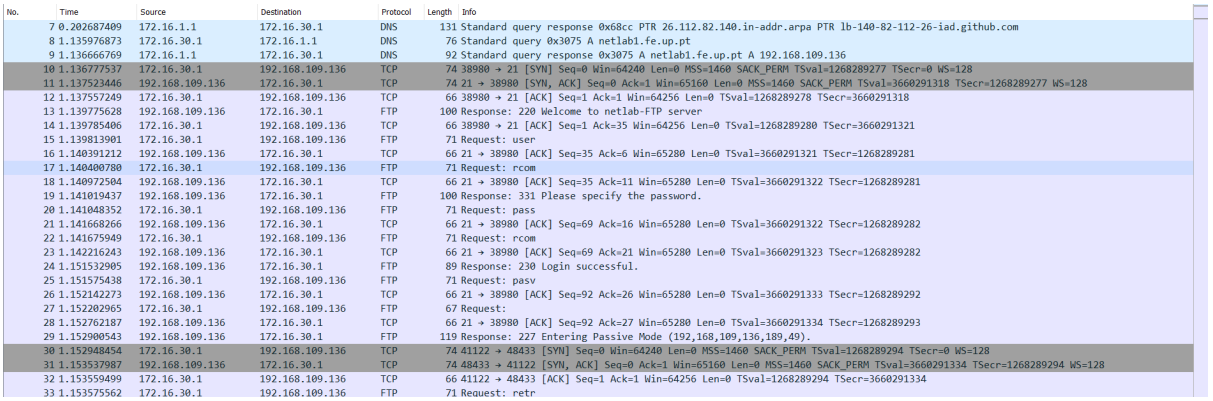


No.	Time	Source	Destination	Protocol	Length	Info
52	3.173507880	172.16.1.1	172.16.30.1	DNS	91	Standard query response 0x5152 A mail.google.com A 142.250.200.69
53	3.173525410	172.16.1.1	172.16.30.1	DNS	103	Standard query response 0x725e AAAA mail.google.com AAAA 2a00:1450:4003:80d::2005
54	3.174165718	172.16.30.1	142.250.200.69	TLSv1.2	2532	Application Data
55	3.174293389	172.16.30.1	142.250.200.69	TLSv1.2	309	Application Data
56	3.190382940	142.250.200.69	172.16.30.1	TCP	66	443 → 57948 [ACK] Seq=1 Ack=2710 Win=1035 Len=0 TSval=1240618447 TSecr=483022259
57	3.383352727	172.16.30.1	172.16.1.1	DNS	76	Standard query 0x46d3 A netlab1.fe.up.pt
58	3.383994851	172.16.1.1	172.16.30.1	DNS	92	Standard query response 0x46d3 A netlab1.fe.up.pt A 192.168.109.136
59	3.384092769	172.16.30.1	192.168.109.136	TCP	74	34112 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1267531726 TSecr=0 WS=128
60	3.384813674	192.168.109.136	172.16.30.1	TCP	74	21 → 34112 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3659533766 TSecr=1267531726 WS=128
61	3.384834277	172.16.30.1	192.168.109.136	TCP	66	34112 → 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1267531727 TSecr=3659533766
62	3.386844249	192.168.109.136	172.16.30.1	FTP	100	Response: 220 Welcome to netlab-FTP server
63	3.386854027	172.16.30.1	192.168.109.136	TCP	66	34112 → 21 [ACK] Seq=1 Ack=35 Win=64256 Len=0 TSval=1267531729 TSecr=3659533768
64	3.386879728	172.16.30.1	192.168.109.136	FTP	71	Request: user
65	3.387445725	192.168.109.136	172.16.30.1	TCP	66	21 → 34112 [ACK] Seq=35 Ack=6 Win=65280 Len=0 TSval=3659533769 TSecr=1267531729
66	3.387453756	172.16.30.1	192.168.109.136	FTP	71	Request: rcom
67	3.387937200	192.168.109.136	172.16.30.1	TCP	66	21 → 34112 [ACK] Seq=35 Ack=11 Win=65280 Len=0 TSval=3659533770 TSecr=1267531729
68	3.388013467	192.168.109.136	172.16.30.1	FTP	100	Response: 331 Please specify the password.
69	3.388041543	172.16.30.1	192.168.109.136	FTP	71	Request: pass
70	3.388660340	192.168.109.136	172.16.30.1	TCP	66	21 → 34112 [ACK] Seq=69 Ack=16 Win=65280 Len=0 TSval=3659533770 TSecr=1267531730
71	3.388667743	172.16.30.1	192.168.109.136	FTP	71	Request: rcom
72	3.390019248	192.168.109.136	172.16.30.1	TCP	66	21 → 34112 [ACK] Seq=69 Ack=21 Win=65280 Len=0 TSval=3659533771 TSecr=1267531730
73	3.399208239	192.168.109.136	172.16.30.1	FTP	89	Response: 230 Login successful.
74	3.399286951	172.16.30.1	192.168.109.136	FTP	71	Request: pasv
75	3.399890103	192.168.109.136	172.16.30.1	TCP	66	21 → 34112 [ACK] Seq=92 Ack=26 Win=65280 Len=0 TSval=3659533781 TSecr=1267531741
76	3.399902814	172.16.30.1	192.168.109.136	FTP	67	Request:
77	3.400501846	192.168.109.136	172.16.30.1	TCP	66	21 → 34112 [ACK] Seq=92 Ack=27 Win=65280 Len=0 TSval=3659533782 TSecr=1267531742
78	3.400645719	192.168.109.136	172.16.30.1	FTP	120	Response: 227 Entering Passive Mode (192,168,109,136,182,145).

```
> Frame 75: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth0, id 0
> Ethernet II, Src: HewlettP_5a:7d:74 (00:21:5a:5a:7d:74), Dst: HewlettP_61:24:92 (00:21:5a:61:24:92)
> Internet Protocol Version 4, Src: 192.168.109.136, Dst: 172.16.30.1
> Transmission Control Protocol, Src Port: 21, Dst Port: 34112, Seq: 92, Ack: 26, Len: 0
```

```
0000  00 21 5a 61 24 92 00 21  5a 5a 7d 74 08 00 45 00   ·!Za$··! ZZ}t··E·
0010  00 34 a4 19 40 00 3d 06  a1 68 c0 a8 6d 88 ac 10   ·4··@·=· ·h··m···
0020  1e 01 00 15 85 40 ca 62  da b4 39 bc f1 6e 80 10   ·····@·b ··9··n··
0030  01 fe f3 84 00 00 01 01  08 0a da 20 0d d5 4b 8c   ········ ··· ··K·
0040  ff dd                                              ··
```

Experiência 6 - Dois downloads



No.	Time	Source	Destination	Protocol	Length	Info
7	0.202687409	172.16.1.1	172.16.30.1	DNS	131	Standard query response 0x68cc PTR 26.112.82.140.in-addr.arpa PTR lb-140-82-112-26-iad.github.com
8	1.135976873	172.16.30.1	172.16.1.1	DNS	76	Standard query 0x3075 A netlab1.fe.up.pt
9	1.136666769	172.16.1.1	172.16.30.1	DNS	92	Standard query response 0x3075 A netlab1.fe.up.pt A 192.168.109.136
10	1.136777537	172.16.30.1	192.168.109.136	TCP	74	38980 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1268289277 TSecr=0 WS=128
11	1.137523446	192.168.109.136	172.16.30.1	TCP	74	21 → 38980 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3660291318 TSecr=1268289277 WS=128
12	1.137557249	172.16.30.1	192.168.109.136	TCP	66	38980 → 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1268289278 TSecr=3660291318
13	1.139775628	192.168.109.136	172.16.30.1	FTP	100	Response: 220 Welcome to netlab-FTP server
14	1.139785406	172.16.30.1	192.168.109.136	TCP	66	38980 → 21 [ACK] Seq=1 Ack=35 Win=64256 Len=0 TSval=1268289280 TSecr=3660291321
15	1.139813901	172.16.30.1	192.168.109.136	FTP	71	Request: user
16	1.140391212	192.168.109.136	172.16.30.1	TCP	66	21 → 38980 [ACK] Seq=35 Ack=6 Win=65280 Len=0 TSval=3660291321 TSecr=1268289281
17	1.140400780	172.16.30.1	192.168.109.136	FTP	71	Request: rcom
18	1.140972504	192.168.109.136	172.16.30.1	TCP	66	21 → 38980 [ACK] Seq=35 Ack=11 Win=65280 Len=0 TSval=3660291322 TSecr=1268289281
19	1.141019437	192.168.109.136	172.16.30.1	FTP	100	Response: 331 Please specify the password.
20	1.141048352	172.16.30.1	192.168.109.136	FTP	71	Request: pass
21	1.141668266	192.168.109.136	172.16.30.1	TCP	66	21 → 38980 [ACK] Seq=69 Ack=16 Win=65280 Len=0 TSval=3660291322 TSecr=1268289282
22	1.141675949	172.16.30.1	192.168.109.136	FTP	71	Request: rcom
23	1.142216243	192.168.109.136	172.16.30.1	TCP	66	21 → 38980 [ACK] Seq=69 Ack=21 Win=65280 Len=0 TSval=3660291323 TSecr=1268289282
24	1.151532905	192.168.109.136	172.16.30.1	FTP	89	Response: 230 Login successful.
25	1.151575438	172.16.30.1	192.168.109.136	FTP	71	Request: pasv
26	1.152142273	192.168.109.136	172.16.30.1	TCP	66	21 → 38980 [ACK] Seq=92 Ack=26 Win=65280 Len=0 TSval=3660291333 TSecr=1268289292
27	1.152202965	172.16.30.1	192.168.109.136	FTP	67	Request:
28	1.152762187	192.168.109.136	172.16.30.1	TCP	66	21 → 38980 [ACK] Seq=92 Ack=27 Win=65280 Len=0 TSval=3660291334 TSecr=1268289293
29	1.152900543	192.168.109.136	172.16.30.1	FTP	119	Response: 227 Entering Passive Mode (192,168,109,136,189,49).
30	1.152948454	172.16.30.1	192.168.109.136	TCP	74	41122 → 48433 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1268289294 TSecr=0 WS=128
31	1.153537987	192.168.109.136	172.16.30.1	TCP	74	48433 → 41122 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3660291334 TSecr=1268289294 WS=128
32	1.153559499	172.16.30.1	192.168.109.136	TCP	66	41122 → 48433 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1268289294 TSecr=3660291334
33	1.153575562	172.16.30.1	192.168.109.136	FTP	71	Request: retr

Comandos

Experiência 1

tux 2- ifconfig eth0 up 172.16.Y1.1/24
tux 3- ifconfig eth0 up 172.16.Y0.1/24
tux 4- ifconfig eth0 up 172.16.Y0.254/24

Experiência 2

switch:
/interface bridge add name=bridgeY0

/interface bridge add name=bridgeY1
/interface bridge port remove [find interface=ether2]
/interface bridge port remove [find interface=ether4]
/interface bridge port remove [find interface=ether12]
/interface bridge port add bridge=bridgeY0 interface=ether2 - tux 3
/interface bridge port add bridge=bridgeY0 interface=ether4 - tux 4
/interface bridge port add bridge=bridgeY1 interface=ether12 - tux 2

Experiência 3

ifconfig eth1 up 172.16.Y1.253/24

switch:
/interface bridge port remove [find interface=ether12]
/interface bridge port add bridge=bridgeY1 interface=ether12

Enable IP Forwarding :
echo 1 > /proc/sys/net/ipv4/ip_forward
Disable ICMP echo-ignore-broadcast:
echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

tux 3: route add -net 172.16.Y1.0/24 gw 172.16.Y0.254
tux 2: route add -net 172.16.Y0.0/24 gw 172.16.Y1.253

Experience 4

switch:
/interface bridge port remove [find interface=ether14]
/interface bridge port add bridge=bridgeY1 interface=ether14

router:
system reset-configuration
/ip address add address=172.16.Z.Y9/24 interface=ether1 (Z na I321 = 1, Z na I320 = 2)
/ip address add address=172.16.Y1.254/24 interface=ether2
/ip route add dst-address=172.16.Y0.0/24 gateway=172.16.Y1.253
/ip route add dst-address = 0.0.0.0/0 gateway=172.16.Z.254 (Z na I321 = 1, Z na I320 = 2)

tux 3: route add default gw 172.16.Y0.254
tux 2 e 4: route add default gw 172.16.Y1.254

tux 2:
echo 0 > /proc/sys/net/ipv4/conf/eth0/accept_redirects
echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects
route delete -net 172.16.Y0.0/24 gw 172.16.Y1.253

Experiência 5
tuxs: nano /etc/resolv.conf
nameserver 172.16.1.1