



# Aplicação de Download FTP e Laboratórios de Configuração de Rede

Licenciatura em Engenharia Informática e Computação  
Redes de Computadores - 2º Trabalho Laboratorial

**Class 9 group 2**

up201906086@edu.fe.up.pt Marcelo Henriques Couto  
up201907361@edu.fe.up.pt Francisco Pinto de Oliveira

Janeiro, 2022

## Contents

<b>1</b>	<b>Sumário</b>	<b>3</b>
<b>2</b>	<b>Introdução</b>	<b>3</b>
<b>3</b>	<b>Aplicação FTP</b>	<b>4</b>
3.1	Arquitetura da aplicação . . . . .	4
3.1.1	Interpretação do URL . . . . .	4
3.1.2	Execução da transferência . . . . .	4
3.2	Exemplo de Download e Resultados . . . . .	5
<b>4</b>	<b>Configuração de uma Rede</b>	<b>6</b>
4.1	Experiência 1 . . . . .	6
4.1.1	Questões . . . . .	6
4.2	Experiência 2 . . . . .	7
4.2.1	Questões . . . . .	7
4.3	Experiência 3 . . . . .	8
4.3.1	Questões . . . . .	8
4.4	Experiência 4 . . . . .	9
4.4.1	Questões . . . . .	9
<b>5</b>	<b>Conclusão</b>	<b>12</b>
<b>6</b>	<b>Bibliografia</b>	<b>12</b>
<b>7</b>	<b>Anexo 1 - Código da aplicação FTP</b>	<b>13</b>
7.1	main . . . . .	13
7.2	ftp . . . . .	13
7.3	url_path_parser . . . . .	18
7.4	utils . . . . .	21
7.5	macros . . . . .	22
<b>8</b>	<b>Comandos de Configuração</b>	<b>23</b>
8.1	Tux52 . . . . .	23
8.2	Tux53 . . . . .	23
8.3	Tux54 . . . . .	23
8.4	Switch . . . . .	23
8.5	Router . . . . .	24
<b>9</b>	<b>Anexo 3 - Wireshark logs</b>	<b>27</b>
9.1	Experiência 1 . . . . .	27
9.2	Experiência 2 . . . . .	27
9.3	Experiência 3 . . . . .	30
9.4	Experiência 4 . . . . .	32

## 1 Sumário

Este relatório é realizado no âmbito do segundo trabalho laboratorial de Redes de Computadores e tem como objetivo a descrição do funcionamento do client FTP desenvolvido, bem como do processo de configuração de uma rede. O trabalho foi concluído com sucesso:

- A aplicação FTP (File Transfer Protocol) é capaz de realizar transferências de ficheiros de qualquer tamanho sem erros
- A rede ficou funcional e configurada corretamente

## 2 Introdução

O projeto em questão foi desenvolvido para a UC de Redes de Computadores e tem como objetivo:

- o desenvolvimento de uma aplicação FTP (File Transfer Protocol), capaz de executar um download através de um URL
- a configuração de uma rede de computadores com ligação à internet, onde a aplicação será testada

Este trabalho está, portanto, dividido em duas componentes, que por sua vez são compostas por módulos. A **organização** do relatório é então a seguinte:

### Aplicação FTP

- Arquitetura da aplicação fluxo da execução e módulos
- Documentação de um download executado com sucesso

### Configuração da Rede

- Arquitetura, Objetivos e Principais configurações
- Análise dos logs e explicação da experiência

para cada experiência, tendo sido realizadas quatro.

- **Conclusões** – síntese da informação apresentada nas secções anteriores e reflexão sobre objetivos e aprendizagens

### 3 Aplicação FTP

#### 3.1 Arquitetura da aplicação

A primeira parte do projeto desenvolvido insidia em criar uma aplicação capaz de executar transferências através de um URL utilizando o protocolo FTP (File Transfer Protocol). No desenvolvimento da aplicação foram tidas em conta as normas RFC959 RFC1738, respeitantes à comunicação com o servidor e interpretação do URL respectivamente.

##### 3.1.1 Interpretação do URL

A primeira parte da aplicação está responsável por interpretar os argumentos contidos no URL, bem como verificar a sua validade. A função `create_url_data-url_path_parser.c`:

1. Verifica se o URL corresponde a um protocolo FTP utilizando a função `is_ftp`
2. Interpreta o utilizador e a password presentes (ou não) no URL, que estarão localizadas logo a seguir ao protocolo e separados por ':' e com '@' a marcar o seu fim
3. Deteta a eventual presença de uma porta com recurso à função `has_port` e, em caso positivo, guarda o seu valor
4. Analiza o caminho e servidor presentes, separa-os, guarda o caminho e o endereço IP com recurso à função `gethostbyname - netdb.h`

Todos os dados interpretados do URL são guardados numa struct do tipo **URLPathData**.

##### 3.1.2 Execução da transferência

Depois de interpretados os dados do URL necessários para a execução da transferência é necessário comunicar o servidor e efetuar o download do ficheiro pretendido. A função `ftp_download - ftp.c`:

1. **ftp\_control\_connect:** Estabelece uma ligação ao servidor (socket) no modo de controlo à porta 21, com recurso à função `connect - sys/socket.h`
2. **ftp\_login:** Efetua autenticação no servidor com as credenciais obtidas pelo URL (comandos USER e PASS e respetivos argumentos são enviados pelo socket)
3. **ftp\_retrieve\_passive\_mode:** Envia um pedido de conexão em modo passivo (comando PASV) \*
4. **ftp\_data\_connect:** Estabelece uma ligação de dados à porta recebida pelo servidor como resposta ao pedido anterior
5. **ftp\_change\_dir\_to\_res:** Na conexão de controlo, muda de diretório para o diretório correto e envia o comando RETR para iniciar o download
6. **f\_retrieve\_fd:** Cria um cicheiro local com o mesmo nome do ficheiro remoto
7. **ftp\_dump\_data:** Realiza a transferência do ficheiro
8. **ftp\_close:** Fecha as conexões com o servidor

Cada pedido ao servidor retorna uma mensagem, da qual retiramos o código através da função `ftp.read_resp_code`. Este código é interpretado de modo a entender se o procedimento está a seguir o seu curso esperado ou se, a algum momento, houve um erro ou rejeição por parte do servidor.

\* em modo passivo, o comando enviado ao servidor é **PASV** em vez de **PORT**, sendo o servidor a retornar a porta por onde a ligação de dados deve ser estabelecida e a transferência efetuada

### 3.2 Exemplo de Download e Resultados

A aplicação foi testada com ficheiros de diferentes tamanhos, tipos e de diferentes fontes, sendo experimentado ambos o modo anónimo (sem credenciais) e com autenticação no servidor. Abaixo temos alguns exemplos de execução documentados.

```
marhcoutho at marchcoutho-TUF-FX505DD in ~/.../TP/FTPDownload/src on main*
$ ls
download  ftp.h  main.c  ubuntu-14.04.6-desktop-i386.iso.zsync  url_path_parser.h  utils.h
ftp.c      macros.h  Makefile  url_path_parser.c           utils.c
marhcoutho at marchcoutho-TUF-FX505DD in ~/.../TP/FTPDownload/src on main*
$ ./download ftp://ftp.up.pt/pub/kodi/timestamp.txt
220
user anonymous
331 Please specify the password.
pass teste
230 Login successful.
250 Directory successfully changed.
250 Directory successfully changed.
150 Opening BINARY mode data connection for timestamp.txt (11 bytes).
Starting download...
Downloaded 100.000000%
226 Transfer complete.
marhcoutho at marchcoutho-TUF-FX505DD in ~/.../TP/FTPDownload/src on main*
$ ls
download  ftp.h  main.c  timestamp.txt                  url_path_parser.c  utils.c
ftp.c      macros.h  Makefile  ubuntu-14.04.6-desktop-i386.iso.zsync  url_path_parser.h  utils.h
marhcoutho at marchcoutho-TUF-FX505DD in ~/.../TP/FTPDownload/src on main*
$ cat timestamp.txt
1643145601
```

Figure 2: Transferência de ficheiro timestamp - 11B

```
marhcoutho at marchcoutho-TUF-FX505DD in ~/.../TP/FTPDownload/src on main*
$ make
gcc -o download ftp.c main.c url_path_parser.c utils.c -Wall
marhcoutho at marchcoutho-TUF-FX505DD in ~/.../TP/FTPDownload/src on main*
$ ./download ftp://ftp.crihan.fr/releases/14.04.6/ubuntu-14.04.6-desktop-amd64.iso
220 Welcome to SYVIK FTP service.
user anonymous
331 Please specify the password.
pass teste
230 Login successful.
250 Directory successfully changed.
250 Directory successfully changed.
150 Opening BINARY mode data connection for ubuntu-14.04.6-desktop-amd64.iso (1157627904 bytes).
Starting download...
Downloaded 100.000000%
226 Transfer complete.
marhcoutho at marchcoutho-TUF-FX505DD in ~/.../TP/FTPDownload/src on main*
$ ls -l
total 1130564
-rwxrwxr-x 1 marchcoutho marchcoutho  22808 jan 26 12:31 download
-rw-rw-r-- 1 marchcoutho marchcoutho   7184 jan 26 12:30 ftp.c
-rw-rw-r-- 1 marchcoutho marchcoutho   175 jan 10 19:18 ftp.h
-rw-rw-r-- 1 marchcoutho marchcoutho    40 jan 10 19:18 macros.h
-rw-rw-r-- 1 marchcoutho marchcoutho   482 jan 26 10:52 main.c
-rw-rw-r-- 1 marchcoutho marchcoutho   119 jan 10 19:18 Makefile
-rw-r--r-- 1 marchcoutho marchcoutho 1157627904 jan 26 12:38 ubuntu-14.04.6-desktop-amd64.iso
-rw-rw-r-- 1 marchcoutho marchcoutho   2908 jan 25 21:45 url_path_parser.c
-rw-rw-r-- 1 marchcoutho marchcoutho   302 jan 10 19:18 url_path_parser.h
-rw-rw-r-- 1 marchcoutho marchcoutho   1236 jan 26 12:30 utils.c
-rw-rw-r-- 1 marchcoutho marchcoutho   195 jan 26 09:34 utils.h
```

Figure 3: Ubuntu iso - 1104 MB

## 4 Configuração de uma Rede

### 4.1 Experiência 1

Nesta experiência configuraram-se endereços IP de duas máquinas ligadas através de um switch para entender melhor como funciona a comunicação entre máquinas e o significado e funcionamento de pacotes ARP.

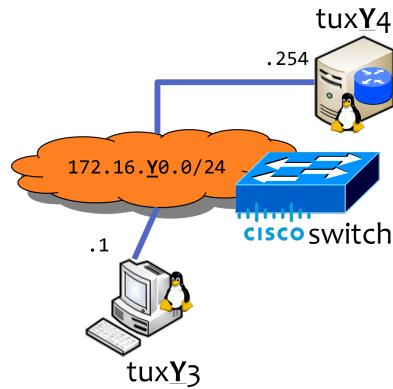


Figure 4: Estrutura da rede

#### 4.1.1 Questões

**O que são os pacotes ARP e para que são usados?** Os pacotes do protocolo ARP são utilizados para, sabendo o endereço IP de uma máquina, descobrir o seu endereço MAC.

**O que são os endereços MAC e IP dos pacotes ARP e porquê?** Os endereços MAC identificam as interfaces com a rede que uma determinada máquina utiliza para comunicar com outras máquinas. Nos pacotes ARP utilizados para pedir o endereço MAC da interface de uma máquina, o primeiro endereço é enviado para que se identifique o endereço IP da máquina cujo endereço MAC é procurado. O segundo é utilizado para identificar a máquina que pretende ser informada. Nos pacotes ARP de resposta, o primeiro endereço IP é enviado para identificar a máquina que possui o endereço MAC enviado em segundo lugar.

**Que pacotes gera o comando PING?** São gerados pacotes ICMP utilizados para mandar erros da camada 3 ou mensagens para routers. Neste caso são utilizados para testar a conectividade entre as máquinas da rede.

**Quais são os endereços IP e MAC dos pacotes de PING?** O endereço IP de origem dos pacotes enviados é 172.16.50.1, o endereço do tux53. O endereço IP de destino é 172.16.50.254, o endereço de IP do tux54. O endereço MAC de origem dos pacotes é 00:21:5a:61:2d:72 o endereço de eth0 do tux53. O endereço MAC de destino é 00:21:5a:c3:78:70, o endereço de eth0 do tux54.

**Como determinar se um frame Ethernet recebido é ARP, IP, ICMP?** Ao analisar os pacotes no Wireshark pode verificar-se o tipo de frame analisando os bytes 12-13 do pacote recebido. Os pacotes ARP têm tipo 0x0806 os pacotes IP têm tipo 0x0800. Os pacotes ICMP têm o mesmo tipo dos pacotes IP e o byte 23 é 0x01. O Wireshark faz esta análise portanto basta ver a coluna Protocol para cada pacote.

**Como determinar o comprimento da trama recebida?** O comprimento das tramas pode ser observado no Wireshark na coluna Length. As tramas IP possuem o seu tamanho nos bytes 16-17.

## 4.2 Experiência 2

Esta experiência teve como objetivo configurar duas VLAN's, bem como entender como funcionam domínios de broadcast.

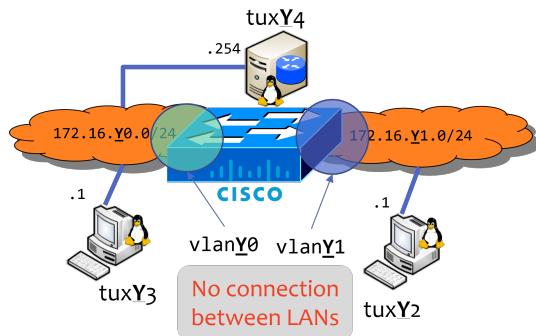


Figure 5: Estrutura da rede

### 4.2.1 Questões

**Como configurar a vlan50** Primeiro conecta-se a interface eth0 do tux53 à conexão correspondente à porta 1 do switch e o a interface eth0 do tux54 à conexão correspondente à porta 2. Em seguida configuramos os endereços IP de ambas as interfaces para que correspondam à especificação

Máquina	Interface	Endereço IP
Tux53	eth0	172.16.50.1
Tux54	eth0	172.16.50.254

Em seguida criamos conectamos a porta série do Tux53 à porta série do switch e configuramos a VLAN50 com as portas 1 e 2 associadas à VLAN pois são as portas a que estão conectados os Tux53 e Tux54.

**Quantos domínios de broadcast há? Como o podemos concluir tendo por base os logs?** Podemos concluir que há dois domínios de Broadcast. Um deles é na VLAN50, uma vez que, os pacotes do ping do tux3 (172.16.50.1) atingem o tux4 (172.16.50.254), mas não atingem tux2(172.16.51.1) como se pode ver nas figuras (Figure 10 e Figure 12). O outro é na VLAN51, no entanto, não há qualquer prova de Broadcast porque o tux2 se encontra isolado na VLAN referida havendo apenas registo dos pacotes a saírem da origem como se pode ver na figura. (Figure 14)

### 4.3 Experiência 3

Esta experiência possibilitou o entendimento do sistema de DNS bem como a configuração de NAT num router comercial.

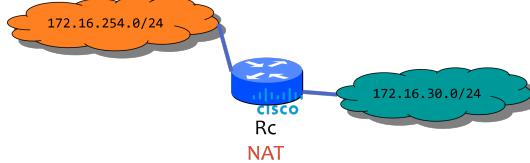


Figure 6: Estrutura da rede

#### 4.3.1 Questões

**Como configurar uma rota estática num router comercial?** Executando o seguinte comando:

```
ip route 172.16.40.0 255.255.255.0 172.16.30.2
```

O primeiro endereço refere-se ao prefixo de origem dos pacotes, o segundo endereço é a máscara de sub-rede, e o último endereço é o gateway dos pacotes.

**Como configurar NAT num router comercial?** Para configurar a NAT deve-se seguir os seguintes comandos: - Identificar a interface de rede com o comando: interface FastEthernet0/0 - Atribuir o endereço de IP que a interface vai tomar nessa NAT com: ip address 172.16.30.1 255.255.255.0 - Especificar o tipo de NAT interno ou externo (neste caso externo): ip nat inside

Após configurar a interface deve configurar-se a pool de endereços exteriores disponíveis com: ip nat pool ovrlid 172.16.254.45 172.16.254.45 prefix-length 24

Por fim, basta configurar a pool IP's internos:

- ip nat inside source list 1 pool ovrlid overload
- access-list 1 permit 172.16.40.0 0.0.0.7
- access-list 1 permit 172.16.30.0 0.0.0.7

**O que faz a NAT?** NAT é uma técnica que permite mapear uma gama de endereços IP para outra gama de endereços IP mudando o IP dos pacotes enviados enquanto estão em trânsito. É utilizada geralmente para mapear o endereço IP público fornecido por um ISP para o endereço privado da máquina do utilizador.

**Como configurar o serviço de DNS no host?** Pode-se configurar traduções específicas em /etc/hosts como fizemos com o mapeamento do endereço de youtubas. Pode-se configurar um servidor central que faz a tradução no ficheiro /etc/resolv.conf adicionando: nameserver IP<sub>DNS</sub>

**Que pacotes são trocados pelo DNS e que informação é transportada** São observados dois tipos de pacote muito semelhantes. Ambos transportam o hostname sobre o qual se quer saber o IP. A resposta aos pacotes do tipo 'A' é um endereço IPv4, já a resposta ao pacotes AAAA é um endereço IPv6.

**Que pacotes ICMP são observados e porquê?** O traceroute tenta encontrar o número mínimo de saltos para alcançar um determinado destino. Para isso gera pacotes UDP com TTL crescente partindo de 1. Quando recebe um pacote ICMP que informa que TTL foi excedido o traceroute sabe que necessita de pelo menos mais um valor de TTL para alcançar o destino. No nosso caso como o destino não aceita pedidos é ainda enviado um pacote ICMP que informa que a porta é inalcançável.

**Quais são os endereços IP e MAC associados a pacotes ICMP e porquê?** Os endereços IP de destino são sempre da nossa máquina, os de origem correspondem aos vários IP intermédios pelos quais os pacotes devem viajar até alcançar o destino (Figure 18). Cada vez que o traceroute aumenta o TTL, o endereço de origem muda, o que significa que o pacote atingiu mais um nó. O endereço MAC de origem é sempre o endereço MAC da interface virtual do computador host, o endereço de destino é o MAC da interface virtual do Guest OS.

**Quais são as rotas na sua máquina? Qual o seu significado?** A rota com origem em 0.0.0.0 e destino em 172.24.64.1 é a default gateway é indica por onde devem ser enviados os pacotes caso não possam ser encaminhados para a rede local. (Figure 19)

A rota com origem em 172.24.64.0 e gateway 0.0.0.0 é a rota inválida para pacotes com destino inválido.

#### 4.4 Experiência 4

Esta experiência é a junção do conhecimento das experiências anteriores para a formação de uma rede interna com duas VLAN's um computador que serve de ponte entre elas e a ligação à internet utilizando o sistema de NAT de um router comercial. O trabalho desenvolvido nesta experiência foi feito fora das aulas teórico-práticas na bancada 1.

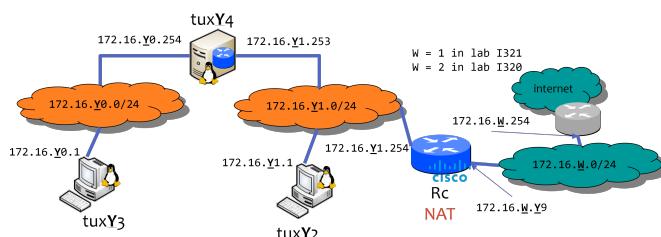


Figure 7: Estrutura da rede

##### 4.4.1 Questões

**Que rotas há nos tuxes? Qual é o seu significado?**

**Tux52**

Destino	Gateway	Máscara de subrede
0.0.0.0	172.16.11.254	0.0.0.0
172.16.11.0	0.0.0.0	255.255.255.0
172.16.10.0	172.16.11.253	255.255.255.0

- Primeira entrada: é o endereço default, indica que todos os pacotes que não tenham match em qualquer outra rota devem ser enviados para o endereço IP 172.16.11.254 neste caso o endereço do router, que encaminhará o pacote para a internet.
- Segunda entrada: significa que qualquer pacote que tenha como destino a VLAN11 deve ser tratado localmente (0.0.0.0) uma vez que já atingiu a VLAN correta e não tem de ser encaminhado.
- Terceira entrada: significa que qualquer pacote que tenha como destino a VLAN10 deve ser enviado para o endereço 172.16.11.253 o endereço da interface do tux14 na VLAN11, uma vez que este computador é a interface entre as duas VLAN e saberá encaminhar o pacote corretamente.

### Tux53

Destino	Gateway	Máscara de subrede
0.0.0.0	172.16.10.254	0.0.0.0
172.16.10.0	0.0.0.0	255.255.255.0
172.16.11.0	172.16.10.254	255.255.255.0

- Primeira entrada: é o endereço default, indica que todos os pacotes que não tenham match em qualquer outra rota devem ser enviados para o endereço IP 172.16.10.254 neste caso o endereço do tux14, que encaminhará o pacote para o router.
- Segunda entrada: significa que qualquer pacote que tenha como destino a VLAN10 deve ser tratado localmente (0.0.0.0) uma vez que já atingiu a VLAN correta e não tem de ser encaminhado.
- Terceira entrada: significa que qualquer pacote que tenha como destino a VLAN11 deve ser enviado para o endereço 172.16.10.254 o endereço da interface do tux14 na VLAN10, uma vez que este computador é a interface entre as duas VLAN e saberá encaminhar o pacote corretamente.

### Tux54

Destino	Gateway	Máscara de subrede
0.0.0.0	172.16.11.254	0.0.0.0
172.16.10.0	0.0.0.0	255.255.255.0
172.16.11.0	0.0.0.0	255.255.255.0

- Primeira entrada: é o endereço default, indica que todos os pacotes que não tenham match em qualquer outra rota devem ser enviados para o endereço IP 172.16.11.254 neste caso o endereço do router, que encaminhará o pacote para a internet.
- Segunda entrada: significa que qualquer pacote que tenha como destino a VLAN10 deve ser tratado localmente (0.0.0.0) uma vez que já atingiu a VLAN correta e não tem de ser encaminhado.
- Terceira entrada: significa que qualquer pacote que tenha como destino a VLAN11 deve ser tratado localmente (0.0.0.0) uma vez que já atingiu a VLAN correta e não tem de ser encaminhado.

**Que informação contém uma entrada da forwarding table?** Uma entrada na forwarding table contém os seguintes dados:

- Endereço de destino para uma subrede (Destination)
- Máscara de subrede(Genmask)
- Endereço do próximo salto (Gateway)
- Flags associadas com ACL
- Métrica associada à ligação. É escolhida a entrada com a métrica menor
- Número de rotas que se referem aquela entrada, este valor não é usado pelo kernel Linux (Ref)
- Coluna com número de consultas à rota (Use)
- Interface de comunicação (Iface)

**Que mensagens ARP, e endereços MAC, são observados e porquê?** Ao fazer ping do tux13 para o tux12 capturando pacotes nas duas interfaces são observados os seguintes pacotes ARP.

Primeiro é observado um pacote que pede que o endereço MAC do IP 172.16.10.254 seja enviado para 172.16.10.1, o endereço MAC observado é 00:21:5a:61:2f:24 (Figure 24). Isto acontece porque o tux13 está a encontrar a interface do tux14 que encaminhará o pacote enviado pelo ping para o tux12, o endereço MAC de resposta corresponde à interface eth0 de tux14.

Em seguida na mesma interface é observado o pacote ARP que pede que o endereço MAC de 172.16.10.1 seja enviado para 172.16.10.254 (Figure 26). Isto acontece porque o tux14 necessita do endereço MAC da interface do tux13 para encaminhar a resposta do tux12 ao ping. O endereço MAC observado é 00:21:5a:61:2d:ef o endereço de eth0 do tux13.

Já em eth1 do tux14 acontece algo semelhante, o tux14 pergunta qual o endereço MAC do tux12 (Figure 23) para lhe poder encaminhar o pacote de PING e este responde com 00:21:5a:61:2e:c3, o endereço MAC de eth0 do tux12. Em seguida o tux12 pede o endereço MAC de eth1 de tux14 (Figure 25) para lhe poder enviar a resposta a PING para que esta seja encaminhada a tux13. O tux14 responde com 00:c0:df:04:20:99 o endereço MAC de eth1 do tux14.

**Quais são os endereços IP e MAC associados aos pacotes ICMP e porquê?** Quando o tux13 envia o pacote ICMP para o tux12 o endereço IP de origem é 172.16.10.1 o endereço de tux13 e o de destino 172.16.11.1, o endereço de tux12. O endereço MAC de origem é o endereço 00:21:5a:61:2d:ef o endereço da interface de tux13 e o de destino 00:21:5a:61:2f:24, o endereço MAC de eth0 de tux14. Isto acontece porque o endereço MAC de destino é sempre o endereço da próxima interface e não o endereço MAC correspondente ao destino final. Como os pacotes têm de passar por tux4 este é o endereço MAC de destino. O mesmo acontece com os pacotes de resposta, os endereços IP são os endereços da origem neste caso o tux12 e o de destino é o de tux13. Já o endereço MAC de origem é o endereço de eth0 de tux14 e o de destino eth0 de tux13.

Na interface eth1 de tux4 acontece algo semelhante. Os pacotes enviados por tux13 têm o seu endereço IP como endereço IP de origem e o endereço IP de tux12 como destino. Já o endereço MAC de origem é o de eth1 de tux14 e o de destino o de eth0 de tux12. O contrário acontece com os pacotes de resposta, estes possuem como endereço de IP de origem o endereço de tux12 e o de destino o de tux13. Mas o endereço MAC de origem é eth0 de tux12 e o de destino eth1 de tux14.

**Quais são os caminhos seguidos pelos pacotes nas experiências realizadas e porquê?**

- **Router CISCO para tux12** - O caminho seguido é router → tux12 pois estes encontram-se na mesma VLAN e podem comunicar diretamente
- **Router CISCO para tux13** - O caminho seguido é router → tux14 → tux13 pois o router encontra-se na VLAN11 e o tux13 na VLAN10 por isso os pacotes enviados têm de ser enviados primeiro para tux14
- **Router CISCO para tux14** - A comunicação é direta uma vez que ambos têm interfaces conectadas à mesma VLAN
- **Router CISCO para a 172.16.1.254** - A comunicação pode ser feita diretamente pois o router têm uma interface ligada à VLAN onde está a máquina de endereço 172.16.1.254.
- **Tux13 para 172.16.1.254** - O caminho seguido é tux13 → tux14 → router → 172.16.1.254, isto acontece porque o tux13 não tem ligação direta a 172.16.1.254 nem ao router por isso o pacote tem de ser enviado ao tux14 que o encaminhará para o router. O router por sua vez encaminha o pacote para 172.16.1.254.
- **Tux13 para 104.17.113.188** - O caminho seguido é tux13 → tux14 → router → 172.16.1.254 → ... → 104.17.113.188, a justificação para o encaminhamento até 172.16.1.254 é a mesma que na comunicação anterior, no entanto após atingir 172.16.1.254 o pacote será encaminhado para a Internet, aí fará um caminho que desconhecemos até atingir 104.17.113.188.

## 5 Conclusão

Os objetivos delineados para este projeto foram plenamente atingidos. O cliente FTP é capaz de efetuar qualquer tipo de download via protocolo FTP, dentro do que era esperado. A configuração da rede foi também um sucesso, dado que o acesso à internet foi conseguido e que todas as ligações entre os computadores que foram pedidas eram possíveis. Este projeto permitiu-nos compreender o processo por de trás de uma aplicação de transferência de ficheiros FTP, bem como alargar os conhecimentos no que toca a redes de computadores e sua configuração.

## 6 Bibliografia

route command Man Pages  
ifconfig command Man Pages  
ip command Man Pages  
Cisco Basic Router Configuration

## 7 Anexo 1 - Código da aplicação FTP

### Modules

- **ftp** - ligação ao servidor e transferência do ficheiro
- **main** - função inicial
- **utils** - funções genéricas que auxiliam o programa
- **url\_path\_parser** - interpretação dos argumentos (URL fornecido)
- **macros** - constantes

### 7.1 main

```
1 #include <stdio.h>
2 #include "ftp.h"
3 #include "url_path_parser.h"
4
5 URLPathData resource_location;
6
7 void print_usage() {
8     printf("Usage: download resource_location\n");
9 }
10
11 int main(int argc, char* argv[]) {
12     setbuf(stdout, NULL);
13     if (argc == 2) {
14         if (create_url_data(argv[1], &resource_location) != 0) {
15             return -1;
16         }
17         return ftp_download(&resource_location);
18     } else {
19         print_usage();
20         return -1;
21     }
22     return 0;
23 }
```

Listing 1: main.c

### 7.2 ftp

```
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <sys/types.h>
4 #include <sys/socket.h>
5 #include <arpa/inet.h>
6 #include <stdlib.h>
7 #include <string.h>
8 #include <errno.h>
9 #include <sys/select.h>
```

```
10 #include "macros.h"
11 #include "ftp.h"
12 #include "utils.h"
13
14 static URLPathData* resource_location;
15 static int server_socket;
16 static int data_socket;
17 static FILE* server_socket_file;
18 static char* buffer;
19 unsigned long file_size = 0;
20 int ftp_control_connect(URLPathData* path_data);
21 int ftp_data_connect(DataConData* data_con);
22 void ftp_close();
23 int ftp_dump_data(int fd);
24 int ftp_read_resp_code();
25 int ftp_login();
26 int ftp_retrieve_passive_mode(DataConData* data_con_data);
27 int ftp_change_dir_to_res(URLPathData* url_data);
28
29
30 // Main function to execute download
31 int ftp_download(URLPathData* path_data) {
32     int fd;
33     DataConData con_data;
34
35     if (ftp_control_connect(path_data) == -1) {
36         return -1;
37     }
38     if (ftp_login() == -1){
39         return -1;
40     }
41     if (ftp_retrieve_passive_mode(&con_data) == -1) {
42         return -1;
43     }
44     if (ftp_data_connect(&con_data) == -1) {
45         return -1;
46     }
47     if (ftp_change_dir_to_res(path_data) == -1) {
48         return -1;
49     }
50     if((fd = f_retrieve_fd(path_data)) == -1) {
51         return -1;
52     }
53     if (ftp_dump_data(fd) == -1) {
54         return -1;
55     }
56     ftp_close();
57     return 0;
58 }
59
60 // Connect to server
61 int ftp_control_connect(URLPathData* path_data) {
62     resource_location = path_data;
63     buffer = calloc(MAX_CTRL_SIZE, sizeof(char));
64     struct sockaddr_in server_addr = {
```

```
65     .sin_family = AF_INET,
66     .sin_addr.s_addr = inet_addr(inet_ntoa(*((struct in_addr *) resource_location->
67     sv_addr))),
68     .sin_port = htons(path_data->port),
69 };
70 if ((server_socket = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
71     perror("ftp_control_connect_socket");
72     return -1;
73 }
74 if (connect(server_socket, (struct sockaddr *) &server_addr, sizeof(server_addr)) ==
75 -1) {
76     perror("ftp_control_connect_connect");
77     return -1;
78 }
79 if ((server_socket_file = fdopen(server_socket, "r")) == NULL) {
80     perror("ftp_control_connect_fdopen");
81     return -1;
82 }
83 if (ftp_read_resp_code() != 220) {
84     fprintf(stderr, "Error: Server responded with unexpected code!\n");
85     return -1;
86 }
87
88 // Establishes data connection
89 int ftp_data_connect(DataConData* data_con) {
90     struct sockaddr_in server_addr = {
91         .sin_family = AF_INET,
92         .sin_addr.s_addr = inet_addr(data_con->ip),
93         .sin_port = htons(data_con->port),
94     };
95     if ((data_socket = socket(AF_INET, SOCK_STREAM | SOCK_NONBLOCK, 0)) == -1) {
96         perror("ftp_data_connect_socket");
97         return -1;
98     }
99     if ((connect(data_socket, (struct sockaddr *) &server_addr, sizeof(server_addr)) ==
100 -1) && (errno != EINPROGRESS)) {
101         perror("ftp_data_connect_connect");
102         return -1;
103     }
104 }
105
106 // Close connection
107 void ftp_close() {
108     close(server_socket);
109     close(data_socket);
110 }
111
112 // Download
113 int ftp_dump_data(int fd) {
114     struct timeval tv;
115     fd_set socket_set;
116     uint8_t byte_stream[1024];
```

```
117     ssize_t read_size;
118     FD_ZERO(&socket_set);
119     FD_SET(data_socket, &socket_set);
120     tv.tv_sec = 5;
121     tv.tv_usec = 0;
122
123     int select_retval = select(data_socket + 1, &socket_set, NULL, NULL, &tv);
124     if (select_retval == -1) {
125         perror("select");
126         return -1;
127     } else if (select_retval == 0) {
128         fprintf(stderr, "Error: Couldn't establish connection with data link.\n");
129         return -1;
130     }
131     printf("Starting download...\n");
132
133     unsigned long downloaded_bytes = 0;
134
135     while((read_size = read(data_socket, byte_stream, 1024)) != 0) {
136         write(fd, byte_stream, read_size);
137         downloaded_bytes += read_size;
138         print_progress(downloaded_bytes, file_size);
139     }
140     printf("\n");
141     write(fd, byte_stream, read_size);
142     if (ftp_read_resp_code() != 226) {
143         fprintf(stderr, "Error: Unexpected response after end of transfer!\n");
144         return -1;
145     }
146     return 0;
147 }
148
149 // Gets server response
150 int ftp_read_resp_code() {
151     do {
152         memset(buffer, 0, MAX_CTRL_SIZE);
153         fgets(buffer, MAX_CTRL_SIZE, server_socket_file);
154     } while (buffer[0] < '1' || buffer[0] > '5' || buffer[3] != ' ');
155     printf("%s", buffer);
156     buffer[3] = '\0';
157     int response = atoi(buffer);
158     if (response == 150) file_size = read_file_size(buffer + 4); // To print the
159     progress percentage
160     return response;
161 }
162
163 // Login
164 int ftp_login() {
165     // User
166     if (resource_location->username != NULL) {
167         strcpy(buffer, "user ");
168         strcat(buffer, resource_location->username);
169     } else {
170         strcpy(buffer, "user anonymous");
```

```
171 }
172 strcat(buffer, "\r\n");
173 write(server_socket, buffer, strlen(buffer));
174 printf("%s", buffer);
175 if (ftp_read_resp_code() != 331) {
176     fprintf(stderr, "Error: Unexpected response after username was sent!\n");
177     return -1;
178 }
179
180 // Password
181 if (resource_location->password != NULL) {
182     strcpy(buffer, "pass ");
183     strcat(buffer, resource_location->password);
184 } else {
185     strcpy(buffer, "pass teste");
186 }
187 strcat(buffer, "\r\n");
188 write(server_socket, buffer, strlen(buffer));
189 printf("%s", buffer);
190 if (ftp_read_resp_code() != 230) {
191     fprintf(stderr, "Error: Unexpected response after password was sent!\n");
192     return -1;
193 }
194 return 0;
195 }
196
197 // Request port for data connection (passive mode FTP)
198 int ftp_retrieve_passive_mode(DataConData* data_con_data) {
199     unsigned ip[4];
200     unsigned msb_port;
201     unsigned lsb_port;
202     strcpy(buffer, "pasv\r\n");
203     write(server_socket, buffer, strlen(buffer));
204     fscanf(server_socket_file, "227 Entering Passive Mode (%u,%u,%u,%u,%u,%u).",
205             &ip[0],
206             &ip[1],
207             &ip[2],
208             &ip[3],
209             &msb_port,
210             &lsb_port
211         );
212     sprintf(data_con_data->ip, "%u.%u.%u.%u", ip[0], ip[1], ip[2], ip[3]);
213     data_con_data->port = msb_port * 256 + lsb_port;
214     return 0;
215 }
216
217 // Changes the current directory to the directory of the file to download
218 int ftp_change_dir_to_res(URLPathData* url_data) {
219     size_t url_size = strlen(url_data->url) + 1;
220     char* url_cpy = calloc(url_size, sizeof(char));
221     strncpy(url_cpy, url_data->url, url_size);
222
223     char* last_token = strrchr(url_cpy, '/') + 1;
224     char* token = strtok(url_cpy, "/");
225     while(token != NULL) {
```

```

226     if (token == last_token) {
227         break;
228     }
229     strcpy(buffer, "CWD ");
230     strcat(buffer, token);
231     strcat(buffer, "\r\n");
232     write(server_socket, buffer, strlen(buffer));
233     if (ftp_read_resp_code() != 250) {
234         fprintf(stderr, "Error: Invalid path to resource!\n");
235         return -1;
236     }
237     token = strtok(NULL, "/");
238 }
239 strcpy(buffer, "RETR ");
240 strcat(buffer, last_token);
241 strcat(buffer, "\r\n");
242 write(server_socket, buffer, strlen(buffer));
243 if (ftp_read_resp_code() != 150) {
244     fprintf(stderr, "Error: Unexpected response to RETR command!\n");
245     return -1;
246 }
247 free(url_cpy);
248 return 0;
249 }
```

Listing 2: ftp.c

```

1 #pragma once
2
3 #include <stdint.h>
4 #include "url_path_parser.h"
5
6 typedef struct {
7     char ip[17];
8     uint16_t port;
9 } DataConData;
10
11 int ftp_download(URLPathData* path_data);
```

Listing 3: ftp.h

### 7.3 url\_path\_parser

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdbool.h>
4 #include <string.h>
5 #include <netdb.h>
6 #include "url_path_parser.h"
7
8 // Prints error message
9 void print_url_format_error() {
10     fprintf(stderr, "Invalid URL format. It should respect the following convention: ftp
11 ://<user>:<password>@<host>:<port>/<url-path>\n");
```

```
12 // Checks if url is FTP
13 bool is_ftp(char* url_path) {
14     if (strncmp(url_path, "ftp://", 6) == 0) {
15         return true;
16     }
17     return false;
18 }
19
20
21 // Checks if path has port
22 bool has_port(char* url_path) {
23     return (strchr(url_path, ':') != NULL);
24 }
25
26 char* strptrncpy(char* start, char* end) {
27     size_t len;
28     if (end == NULL) {
29         len = strlen(start);
30     } else {
31         len = (size_t) (end - start);
32     }
33     char* dest = calloc(len + 1, sizeof(char));
34     for(size_t i = 0; i < len && start != end; i++) {
35         dest[i] = *start;
36         start++;
37     }
38     dest[len] = '\0';
39     return dest;
40 }
41
42 // Parses the information from the url to a struct
43 int create_url_data(char* url_path, URLPathData* url_data) {
44     char* server_name;
45     struct hostent* sv_resesolv;
46     url_data->port = 21;
47
48     // Check if ftp
49     if (!is_ftp(url_path)) {
50         print_url_format_error();
51         return -1;
52     }
53
54     // User and password
55     url_path += strlen("ftp://"); //Removes protocol from url path
56     char* at_place = strchr(url_path, '@');
57     if (at_place != NULL) {
58         char* user_pass_sep = strchr(url_path, ':');
59         if (user_pass_sep == NULL || user_pass_sep > at_place) {
60             print_url_format_error();
61             return -1;
62         }
63         url_data->username = strptrncpy(url_path, user_pass_sep);
64         url_data->password = strptrncpy(user_pass_sep + 1, at_place);
65         url_path = at_place + 1;
66     }
```

```
67
68     char* resource_slash = strchr(url_path, '/');
69     if(resource_slash == NULL) {
70         print_url_format_error();
71         return -1;
72     }
73
74     // Port
75     if (has_port(url_path)) {
76         char* port_tok = strchr(url_path, ':');
77         if (port_tok == NULL) {
78             print_url_format_error();
79             return -1;
80         }
81         server_name = strptrncpy(url_path, port_tok);
82         char* port_str = strptrncpy(port Tok + 1, resource_slash);
83         url_data->port = (uint16_t)atoi(port_str);
84         free(port_str);
85     } else {
86         server_name = strptrncpy(url_path, resource_slash);
87     }
88
89
90     // Url
91     url_data->url = strptrncpy(resource_slash + 1, NULL);
92     if ((sv_resesolv = gethostbyname(server_name)) == NULL) {
93         perror("create_url_data");
94         return -1;
95     }
96
97     // Server
98     url_data->sv_addr = (char*) calloc(strlen(sv_resesolv->h_addr) + 1, sizeof(char));
99     strcpy(url_data->sv_addr, sv_resesolv->h_addr);
100    free(server_name);
101    return 0;
102}
103
104 void destroy_url_data(URLPathData data) {
105     free(data.username);
106     free(data.password);
107     free(data.url);
108     free(data.sv_addr);
109 }
```

Listing 4: url\_path\_parser.c

```
1 #pragma once
2
3 #include <stdint.h>
4
5 typedef struct {
6     char* username;
7     char* password;
8     char* url;
9     char* sv_addr;
10    uint16_t port;
```

```
11 } URLPathData;
12
13 int create_url_data(char* url_path, URLPathData* url_data);
14 char* strptrcpy(char* start, char* end);
15 void destroy_url_data(URLPathData data);
```

Listing 5: url\_path\_parser.h

## 7.4 utils

```
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <string.h>
4 #include <sys/stat.h>
5 #include <sys/types.h>
6 #include <stdlib.h>
7 #include <fcntl.h>
8 #include "utils.h"
9
10
11
12 int f_retrieve_fd(URLPathData* url_data) {
13     if (url_data == NULL) {
14         fprintf(stderr, "Error: Invalid path!\n");
15         return -1;
16     }
17     char* file_name = strrchr(url_data->url, '/') + 1 ?: url_data->url;
18     if (access(file_name, F_OK) != -1) {
19         fprintf(stderr, "Missing permissions or file already exists\n");
20         return -1;
21     }
22     return creat(file_name, S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH);
23 }
24
25
26 // Reads file size from server message
27 unsigned long read_file_size(char* buffer) {
28     int i = 0;
29     char* file_size_str;
30     while(buffer[i] != 40) i++;
31
32     i++;
33     file_size_str = buffer + i;
34     while(buffer[i] != 32) i++;
35
36     buffer[i] = '\0';
37
38     return atol(file_size_str);
39 }
40
41 // Prints progress message
42 void print_progress(unsigned long progress, unsigned long total) {
43     if (total == 0) {
44         fprintf(stderr, "\rDownload file size not defined\n");
45         return;
```

```
46     }
47     double percentage = ((double) progress) / ((double) total) * 100;
48     printf("\rDownloaded %lf%%\n", percentage);
49     sleep(1);
50 }
```

Listing 6: utils.c

```
1 #pragma once
2
3 #include "url_path_parser.h"
4
5 int f_retrieve_fd(URLPathData* url_data);
6 unsigned long read_file_size(char* buffer);
7 void print_progress(unsigned long progress, unsigned long total);
```

Listing 7: utils.h

## 7.5 macros

```
1 #pragma once
2
3 #define MAX_CTRL_SIZE 1024
```

Listing 8: macros.h

## 8 Comandos de Configuração

### 8.1 Tux52

```
1 #!/bin/bash
2 ifconfig eth0 172.16.51.1/24
3 route add -net 172.16.50.0/24 gw 172.16.51.253
4 route add default gw 172.16.51.254
```

Listing 9: tux52\_configuration

### 8.2 Tux53

```
1 #!/bin/bash
2 ifconfig eth0 172.16.50.1/24
3 route add -net 172.16.51.0/24 gw 172.16.50.254
4 route add default gw 172.16.50.254
```

Listing 10: tux53\_configuration

### 8.3 Tux54

```
1 #!/bin/bash
2 ifconfig eth0 172.16.50.254/24
3 ifconfig eth1 172.16.51.253/24
4 route add default gw 172.16.51.254
5 echo 1 > /proc/sys/net/ipv4/ip_forward
6 echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

Listing 11: tux54\_configuration

### 8.4 Switch

```
1 configure terminal
2 vlan 50
3 end
4
5 configure terminal
6 vlan 51
7 end
8
9 #Adiciona porta 1 (Tux53)
10 configure terminal
11 interface fastethernet 0/1
12 switchport mode access
13 switchport access vlan 50
14 end
15
16 #Adiciona porta 2 (Tux54)
17 configure terminal
18 interface fastethernet 0/2
19 switchport mode access
```

```
20 switchport access vlan 50
21 end
22
23 #Adiciona porta 3 (Tux52)
24 configure terminal
25 interface fastethernet 0/3
26 switchport mode access
27 switchport access vlan 51
28 end
29
30 #Adicionar porta 4 (Tux54)
31 configure terminal
32 interface fastethernet 0/4
33 switchport mode access
34 switchport access vlan 51
35 end
36
37 #Adicionar porta 5 (Router)
38 configure terminal
39 interface fastethernet 0/5
40 switchport mode access
41 switchport access vlan 51
42 end
```

Listing 12: switch\_configuration

## 8.5 Router

```
1 !
2 ! Last configuration change at 16:26:26 UTC Wed Jan 19 2022 by root
3 !
4 version 15.0
5 service timestamps debug datetime msec
6 service timestamps log datetime msec
7 service password-encryption
8 !
9 hostname lab-tux5
10 !
11 boot-start-marker
12 boot-end-marker
13 !
14 logging buffered 51200 warnings
15 enable secret 5 $1$u53Q$vBawpP8.1YpCT6ypap1zX.
16 !
17 no aaa new-model
18 !
19 no ipv6 cef
20 ip source-route
21 ip cef
22 !
23 !
24 !
25 !
26 no ip domain lookup
27 multilink bundle-name authenticated
```

```
28 !
29 !
30 !
31 !
32 license udi pid CISCO2901/K9 sn FHK134976HR
33 !
34 !
35 archive
36 log config
37 hidekeys
38 username root privilege 15 secret 5 $1$8AFR$bNAYevxPFjXFExpnZI2fj .
39 username cisco password 7 02050D480809
40 !
41 !
42 !
43 !
44 !
45 !
46 interface GigabitEthernet0/0
47 description $ETH-LAN$$ETH-SW-LAUNCH$$INTF-INFO-FE 0$
48 ip address 172.16.51.254 255.255.255.0
49 ip nat inside
50 ip virtual-reassembly
51 duplex auto
52 speed auto
53 !
54 interface GigabitEthernet0/1
55 ip address 172.16.1.59 255.255.255.0
56 ip nat outside
57 ip virtual-reassembly
58 duplex auto
59 speed auto
60 !
61 interface Serial0/0/0
62 no ip address
63 shutdown
64 clock rate 2000000
65 !
66 interface Serial0/0/1
67 no ip address
68 shutdown
69 clock rate 2000000
70 !
71 ip forward-protocol nd
72 !
73 ip http server
74 ip http access-class 23
75 ip http authentication local
76 ip http secure-server
77 ip http timeout-policy idle 60 life 86400 requests 10000
78 !
79 ip nat pool ovrlid 172.16.1.59 172.16.1.59 prefix-length 24
80 ip nat inside source list 1 pool ovrlid overload
81 ip route 0.0.0.0 0.0.0.0 172.16.1.254
82 ip route 172.16.50.0 255.255.255.0 172.16.51.253
```

```
83 !
84 access-list 1 permit 172.16.50.0 0.0.0.7
85 access-list 1 permit 172.16.51.0 0.0.0.7
86 access-list 23 permit 10.10.10.0 0.0.0.7
87 access-list 23 permit 172.16.0.0 0.0.1.255
88 !
89 !
90 !
91 control-plane
92 !
93 !
94 line con 0
95 login local
96 line aux 0
97 line vty 0 4
98 access-class 23 in
99 privilege level 15
100 login local
101 transport input telnet ssh
102 line vty 5 15
103 access-class 23 in
104 privilege level 15
105 login local
106 transport input telnet ssh
107 !
108 scheduler allocate 20000 1000
109 end
```

Listing 13: router\_configuration

## 9 Anexo 3 - Wireshark logs

### 9.1 Experiência 1

No.	Time	Source	Destination	Protocol	Length	Info
4	2.000967...	Cisco_3a:f6:03	Spanning-tree-(...)	STP	60	Conf. Root = 32768/1/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
5	2.541939...	172.16.50.254	172.16.1.1	DNS	97	Standard query 0x5fc0 A 1.debian.pool.ntp.org.netlab.fe.up.pt
6	2.541940...	172.16.50.254	172.16.1.1	DNS	97	Standard query 0x5fc0 AAAA 1.debian.pool.ntp.org.netlab.fe.up.pt
7	2.616077...	HeuletP_c3:78:70	HeuletP_61:2d:72 ARP	ARP	60	Who has 172.16.50.17 tell 172.16.50.254
8	2.616096...	HeuletP_61:2d:72	HeuletP_c3:78:70 ARP	ARP	60	172.16.50.15 is at 00:21:5a:61:2d:72
9	4.000811...	Cisco_3a:f6:03	Spanning-tree-(...)	STP	60	Conf. Root = 32768/1/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
10	4.074408...	Cisco_3a:f6:03	CDP/VT/DPF/Pag...	DPF	60	Dynamic Trunk Protocol
11	4.074409...	Cisco_3a:f6:03	CDP/VT/DPF/Pag...	DPF	60	Dynamic Trunk Protocol
12	5.785440...	172.16.50.1	172.16.1.1	DNS	63	Standard query 0x0d62 A 2.debian.pool.ntp.org
13	5.785452...	172.16.50.1	172.16.1.1	DNS	63	Standard query 0x0e5e AAAA 2.debian.pool.ntp.org
14	5.793292...	172.16.50.254	ICMP	98	Echo (ping) request id=0x056e, seq=1/256, ttl=64 (reply in 15)	
15	5.794797...	172.16.50.254	ICMP	98	Echo (ping) reply id=0x056e, seq=1/256, ttl=64 (request in 14)	
16	5.988291...	HeuletP_61:2d:72	HeuletP_c3:78:70 ARP	ARP	42	Who has 172.16.50.254 Tell 172.16.50.1
17	5.988406...	HeuletP_c3:78:70	HeuletP_61:2d:72 ARP	ARP	60	172.16.50.254 is at 00:21:5a:c3:78:70 Cost = 0 Port = 0x0003
18	6.000000...	172.16.50.254	172.16.50.254	ICMP	60	Request 0x0d62 172.16.50.254
19	6.000372...	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x056e, seq=2/12, ttl=64 (reply in 21)
20	6.000459...	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x056e, seq=2/12, ttl=64 (request in 20)
22	7.547035...	172.16.50.254	193.136.28.18	DNS	97	Standard query 0x5fc0 A 1.debian.pool.ntp.org.netlab.fe.up.pt
23	7.547052...	172.16.50.254	193.136.28.18	DNS	97	Standard query 0x0d62 AAA 1.debian.pool.ntp.org.netlab.fe.up.pt
24	8.000465...	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x056e, seq=3/768, ttl=64 (reply in 25)
25	8.000466...	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x056e, seq=3/768, ttl=64 (request in 24)
26	8.000467...	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x056e, seq=4/1536, ttl=64 (request in 26)
27	9.028327...	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x056e, seq=4/1536, ttl=64 (reply in 28)
28	9.028456...	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x056e, seq=4/1536, ttl=64 (request in 27)
29	16.024296...	Cisco_3a:f6:03	Spanning-tree-(...)	STP	60	Conf. Root = 32768/1/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
30	16.024300...	172.16.50.254	172.16.50.254	ICMP	98	Echo (ping) request id=0x056e, seq=1/256, ttl=64 (reply in 31)
31	16.024704...	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x056e, seq=1/256, ttl=64 (request in 30)
32	16.799057...	172.16.50.1	193.136.28.18	DNS	83	Standard query 0x0d62 A 2.debian.pool.ntp.org
33	16.799058...	172.16.50.1	193.136.28.18	DNS	83	Standard query 0x0e5e AAA 2.debian.pool.ntp.org
34	11.07633...	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x056e, seq=4/1536, ttl=64 (reply in 35)
35	11.07647...	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x056e, seq=4/1536, ttl=64 (request in 34)
36	11.076478...	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x056e, seq=5/1536, ttl=64 (request in 36)
37	12.10002...	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x056e, seq=5/1536, ttl=64 (reply in 38)
38	12.100452...	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x056e, seq=5/1536, ttl=64 (request in 37)
39	12.55271...	172.16.50.254	172.16.1.1	DNS	83	Standard query 0x210b A 2.debian.pool.ntp.org
40	12.55272...	172.16.50.254	172.16.1.1	DNS	83	Standard query 0x0d61 AAA 2.debian.pool.ntp.org
41	13.12433...	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x056e, seq=6/2048, ttl=64 (reply in 42)
42	13.12446...	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x056e, seq=6/2048, ttl=64 (request in 41)
43	13.124468...	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x056e, seq=7/2048, ttl=64 (request in 43)
44	14.14933...	172.16.50.1	172.16.50.254	ICMP	98	Echo (ping) request id=0x056e, seq=8/2048, ttl=64 (reply in 45)
45	14.14946...	172.16.50.254	172.16.50.1	ICMP	98	Echo (ping) reply id=0x056e, seq=8/2048, ttl=64 (request in 44)

Figure 8: Experiência 1 - Resultado do ping de tux3 para tux4

### 9.2 Experiência 2

No.	Time	Source	Destination	Protocol	Length	Info
2	2.000488...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
3	4.000763...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
4	5.885945...	Cisco_3a:f6:05	Loop	60	Reply	
5	6.014777...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
6	8.000489...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
7	8.000490...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
8	12.02937...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
9	14.03416...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
10	15.88846...	Cisco_3a:f6:05	Loop	60	Reply	
11	16.03913...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
12	18.04899...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
13	20.05000...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
14	22.05374...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
15	24.06367...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
16	25.08606...	Cisco_3a:f6:05	Loop	60	Reply	
17	26.06354...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
18	26.06364...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
19	26.06365...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
20	26.06366...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
21	34.08303...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
22	35.09380...	Cisco_3a:f6:05	Loop	60	Reply	
23	36.09301...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
24	38.09286...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
25	40.09305...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
26	42.10780...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
27	44.10747...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
28	45.09732...	Cisco_3a:f6:05	Loop	60	Reply	
29	46.11233...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
30	48.12505...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
31	51.08330...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
32	51.68310...	Cisco_3a:f6:05	CDP/VT/DPF/9AG...	DPF	459	Device ID: tux05 Port ID: Fastetherthernet0/3
33	52.12701...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
34	54.13687...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
35	55.91059...	Cisco_3a:f6:05	Loop	60	Reply	
36	56.13679...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
37	58.13616...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
38	60.13653...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
39	62.15156...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
40	64.16134...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/51/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005
41	65.92329...	Cisco_3a:f6:05	Loop	60	Reply	
42	65.92330...	Cisco_3a:f6:05	Spanning-tree-(...)	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005

Figure 9: Experiência 2 - Resultado do ping broadcast de tux3 em tux2

24	34.27057...	172.16.50.1	172.16.50.255	ICMP	98	Echo (ping) request id=0x1424, seq=2/512, ttl=64 (no response found!)
25	35.29457...	172.16.50.1	172.16.50.255	ICMP	98	Echo (ping) request id=0x1424, seq=3/768, ttl=64 (no response found!)
26	36.12048...	Cisco_3a:f6:03	Spanning-tree-(...)	STP	60	Conf. Root = 32768/50/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0005

Figure 10: Experiência 2 - tux3 ping broadcast tux3

Redes de Computadores - 1º Trabalho Laboratorial

No.	Time	Source	Destination	Protocol	Length	Info
1	20.084550...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
2	20.084550...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
3	20.084550...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
4	4.068068...	Cisco_3a:f6:03	Cisco_3a:f6:03	LOOP	00	Reply
5	6.022363...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
6	6.022393...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
7	10.05996...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
8	10.060000...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
9	10.060000...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
10	14.07551...	Cisco_3a:f6:03	Cisco_3a:f6:03	LOOP	00	Reply
11	16.07193...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
12	18.07709...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
13	20.08160...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
14	22.08200...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
15	24.08334...	Cisco_3a:f6:03	Cisco_3a:f6:03	LOOP	00	Reply
16	26.08455...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
17	26.08916...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
18	28.10115...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
19	30.10604...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
20	32.11080...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
21	33.26128...	172.16.50.1	172.16.50.255	ICMP	98	Echo (ping) request 1d=0x1424, seq=1/256, ttl=64 (no response found!)
22	34.08652...	Cisco_3a:f6:03	Cisco_3a:f6:03	LOOP	00	Reply
23	36.26128...	172.16.50.1	172.16.50.255	ICMP	98	Echo (ping) request 1d=0x1424, seq=2/256, ttl=64 (no response found!)
24	34.27057...	172.16.50.1	172.16.50.255	ICMP	98	Echo (ping) request 1d=0x1424, seq=2/512, ttl=64 (no response found!)
25	35.29457...	172.16.50.1	172.16.50.255	ICMP	98	Echo (ping) request 1d=0x1424, seq=3/768, ttl=64 (no response found!)
26	36.12048...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
27	36.31857...	172.16.50.1	172.16.50.255	ICMP	98	Echo (ping) request 1d=0x1424, seq=4/1024, ttl=64 (no response found!)
27	37.34257...	172.16.50.1	172.16.50.255	ICMP	98	Echo (ping) request 1d=0x1424, seq=5/1280, ttl=64 (no response found!)
28	38.36657...	172.16.50.1	172.16.50.255	ICMP	98	Echo (ping) request 1d=0x1424, seq=6/1536, ttl=64 (no response found!)
29	39.39987...	172.16.50.1	172.16.50.255	ICMP	98	Echo (ping) request 1d=0x1424, seq=7/1792, ttl=64 (no response found!)
30	39.48621...	Cisco_3a:f6:03	CDP/TP/DP/PAg	CDP	949	Device ID: tux-sws Port ID: FastEthernet0/1
31	40.13933...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
34	40.41454...	172.16.50.1	172.16.50.255	ICMP	98	Echo (ping) request 1d=0x1424, seq=8/2048, ttl=64 (no response found!)
35	41.49857...	172.16.50.1	172.16.50.255	ICMP	98	Echo (ping) request 1d=0x1424, seq=9/2304, ttl=64 (no response found!)
36	42.49857...	172.16.50.1	172.16.50.255	ICMP	98	Echo (ping) request 1d=0x1424, seq=10/2560, ttl=64 (no response found!)
37	42.46257...	172.16.50.1	172.16.50.255	ICMP	98	Echo (ping) request 1d=0x1424, seq=11/2816, ttl=64 (no response found!)
38	43.49857...	172.16.50.1	172.16.50.255	ICMP	98	Echo (ping) request 1d=0x1424, seq=12/3072, ttl=64 (no response found!)
39	44.08965...	Cisco_3a:f6:03	Cisco_3a:f6:03	LOOP	00	Reply
40	44.13993...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
41	46.14480...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003
42	46.14961...	Cisco_3a:f6:03	Spanning-tree(...)	STP	00	Conf, Root = 32768/59/fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x0003

Figure 11: Experiência 2 - Resultado do ping broadcast de tux3 em tux3

26	38.28003...	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x1424, seq=2/512, ttl=64 (no response found!)
27	39.30403...	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x1424, seq=3/768, ttl=64 (no response found!)
28	40.13002...	Cisco 3a:f6:04	Spanning-tree-...	STP	60 Conf. Root = 32768/50:fc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8004

Figure 12: Experiência 2 - tux3 ping broadcast tux4

No.	Time	Source	Destination	Protocol	Length	Info
11.18	07:46:47	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000
12.18.005909	Cisco_3a:f6:04	Cisco_3a:f6:04	Cisco_3a:f6:04	ICMP	88	Echo (ping) request id=0x1424, seq=1/256, ttl=64 (no response found!)
13.20.00161	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
14.22.005669	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
15.24.009161	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
16.26.009680	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
17.28.009269	Cisco_3a:f6:04	Cisco_3a:f6:04	LOOP	88	Reply	
18.30.009269	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
19.36.105772	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
20.32.110782	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
21.34.115959	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
22.36.120396	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
23.37.27975	Cisco_3a:f6:04	ICMP	98	Echo (ping) request id=0x1424, seq=1/256, ttl=64 (no response found!)		
24.38.009680	Cisco_3a:f6:04	Cisco_3a:f6:04	LOOP	88	Reply	
25.40.009680	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
26.38.28003	Cisco_3a:f6:04	ICMP	98	Echo (ping) request id=0x1424, seq=2/252, ttl=64 (no response found!)		
27.39.30403	Cisco_3a:f6:04	ICMP	98	Echo (ping) request id=0x1424, seq=3/268, ttl=64 (no response found!)		
28.40.139802	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
29.46.32802	Cisco_3a:f6:04	ICMP	98	Echo (ping) request id=0x1424, seq=4/182, ttl=64 (no response found!)		
30.41.35202	Cisco_3a:f6:04	ICMP	98	Echo (ping) request id=0x1424, seq=5/120, ttl=64 (no response found!)		
31.42.37602	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
32.42.37602	Cisco_3a:f6:04	ICMP	98	Echo (ping) request id=0x1424, seq=6/155, ttl=64 (no response found!)		
33.43.40802	Cisco_3a:f6:04	ICMP	98	Echo (ping) request id=0x1424, seq=7/179, ttl=64 (no response found!)		
34.44.16932	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
35.44.42402	Cisco_3a:f6:04	ICMP	98	Echo (ping) request id=0x1424, seq=8/204, ttl=64 (no response found!)		
36.45.44802	Cisco_3a:f6:04	ICMP	98	Echo (ping) request id=0x1424, seq=9/234, ttl=64 (no response found!)		
37.45.88463	Cisco_3a:f6:04	CDP/TP/DTP/Pagoda	CDP	459	Device ID: tux-sw9 Port ID: FastEthernet0/2	
38.46.10000	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
39.46.47200	Cisco_3a:f6:04	ICMP	98	Echo (ping) request id=0x1424, seq=10/256, ttl=64 (no response found!)		
40.47.49601	Cisco_3a:f6:04	ICMP	98	Echo (ping) request id=0x1424, seq=11/2016, ttl=64 (no response found!)		
41.48.009193	Cisco_3a:f6:04	Cisco_3a:f6:04	LOOP	88	Reply	
42.48.148980	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
43.50.154607	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
44.52.15912	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
45.54.16985	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
46.56.17594	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
47.58.11166	Cisco_3a:f6:04	ICMP	88	Reply		
48.59.17594	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
49.60.17854	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
50.62.18855	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
51.64.18852	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	
52.66.19326	Cisco_3a:f6:04	Spanning-tree-1	STP	0	0x0000	

Figure 13: Experiência 2 - Resultado do ping broadcast de tux3 em tux4

19	21.10153...	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x1249, seq=4/1024, ttl=64 (no response found!)
20	22.12552...	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x1249, seq=5/1280, ttl=64 (no response found!)
21	22.60705...	Cisco 3a-f6-05	Snapping-tree-1	STP	60 Conf Root = 30768/51/fe8:f1-3a-f6-00 Cost = 0 Port = 0xa005

Figure 14: Experiência 2 - tux2 ping broadcast

## Redes de Computadores - 1º Trabalho Laboratorial

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000	Cisco_3a:f6:05	Cisco_3a:f6:05	LOOP	60	60 Reply
2	0.553335	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
3	2.558215	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
4	4.568215	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
5	5.696249	Cisco_3a:f6:05	CDP/VT/PD/PAG-	CDP	459	Device ID: tux2w5 Port ID: Fastethernet0/3
6	8.572871	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
7	8.572871	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
8	10.00778	Cisco_3a:f6:05	Cisco_3a:f6:05	LOOP	60	60 Reply
9	10.58277	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
10	12.58277	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
11	14.59277	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
12	16.59277	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
13	18.03646	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=1/256, ttl=64 (no response found!)
14	18.59730	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
15	19.03535	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=2/252, ttl=64 (no response found!)
16	20.02959	Cisco_3a:f6:05	Cisco_3a:f6:05	LOOP	60	60 Reply
17	20.07754	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=3/256, ttl=64 (no response found!)
18	21.10153	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=4/252, ttl=64 (no response found!)
19	22.12552	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=5/280, ttl=64 (no response found!)
20	22.66709	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
21	23.14954	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=6/256, ttl=64 (no response found!)
22	23.17352	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=7/252, ttl=64 (no response found!)
23	24.17352	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=8/256, ttl=64 (no response found!)
24	24.17352	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=9/252, ttl=64 (no response found!)
25	25.19754	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=10/256, ttl=64 (no response found!)
26	26.22153	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=11/252, ttl=64 (no response found!)
27	26.62187	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
28	27.24553	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=12/256, ttl=64 (no response found!)
29	28.26953	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=13/281, ttl=64 (no response found!)
30	28.62174	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005

Figure 15: Experiência 2 - Resultado do ping broadcast de tux2 em tux2

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000	Cisco_3a:f6:05	Cisco_3a:f6:05	LOOP	60	60 Reply
2	0.553335	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
3	2.558215	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
4	4.568215	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
5	5.696249	Cisco_3a:f6:05	CDP/VT/PD/PAG-	CDP	459	Device ID: tux2w5 Port ID: Fastethernet0/3
6	8.572871	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
7	8.572871	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
8	10.00778	Cisco_3a:f6:05	Cisco_3a:f6:05	LOOP	60	60 Reply
9	10.58277	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
10	12.58277	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
11	14.59277	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
12	16.59277	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
13	18.03646	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=1/256, ttl=64 (no response found!)
14	18.59730	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
15	19.03535	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=2/252, ttl=64 (no response found!)
16	20.02959	Cisco_3a:f6:05	Cisco_3a:f6:05	LOOP	60	60 Reply
17	20.07754	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=3/256, ttl=64 (no response found!)
18	21.10153	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=4/252, ttl=64 (no response found!)
19	22.12552	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=5/280, ttl=64 (no response found!)
20	22.66709	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
21	23.14954	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=6/256, ttl=64 (no response found!)
22	23.17352	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=7/252, ttl=64 (no response found!)
23	24.17352	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=8/256, ttl=64 (no response found!)
24	24.17352	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=9/252, ttl=64 (no response found!)
25	25.19754	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=10/256, ttl=64 (no response found!)
26	26.22153	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=11/252, ttl=64 (no response found!)
27	26.62187	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005
28	27.24553	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=12/256, ttl=64 (no response found!)
29	28.26953	172.16.51.1	172.16.51.255	ICMP	98	Echo (ping) request id=0x1249, seq=13/281, ttl=64 (no response found!)
30	28.62174	Cisco_3a:f6:05	Spanning-tree-(~)	STP	60	60 Conf. Root = 32768/51/fcc:fb:fb:3a:f6:00 Cost = 0 Port = 0x8005

Figure 16: Experiência 2 - Resultado do ping broadcast de tux2 em tux3

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
2	2.010940	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
3	3.653796	Cisco_3a:f6:04	Cisco_3a:f6:04	LOOP	60	Reply
4	4.000000	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
5	6.014604	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
6	8.024880	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
7	10.024430	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
8	12.029600	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
9	13.638800	Cisco_3a:f6:04	Cisco_3a:f6:04	LOOP	60	Reply
10	14.000000	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
11	16.038924	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
12	18.043650	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
13	20.053550	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
14	21.340004	Cisco_3a:f6:04	CDP/VT/DPF/Pag.- CDP	450	Device ID: tux-sb Port ID: Fastethermet#0/2	
15	22.051424	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
16	23.658500	Cisco_3a:f6:04	Cisco_3a:f6:04	LOOP	60	Reply
17	24.000000	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
18	26.068244	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
19	28.067914	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
20	30.073084	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
21	32.077600	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
22	33.051104	Cisco_3a:f6:04	Cisco_3a:f6:04	LOOP	60	Reply
23	34.000000	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
24	36.087404	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
25	38.092104	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
26	40.097114	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
27	42.101984	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
28	43.658304	Cisco_3a:f6:04	Cisco_3a:f6:04	LOOP	60	Reply
29	44.000000	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
30	46.111604	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
31	48.116404	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
32	50.121254	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004
33	52.128224	Cisco_3a:f6:04	Spanning-tree(-)	STP	60	Conf., Root = 32768/59/fccfbfb3a:f6:00 Cost = 0 Port = 0x8004

Figure 17: Experiência 2 - Resultado do ping broadcast de tux2 em tux4

### 9.3 Experiência 3

```
traceroute to 104.17.113.188 (104.17.113.188), 30 hops max, 60 byte packets
 1  172.24.64.1  0.788 ms  0.682 ms  0.672 ms
 2  192.168.1.1  25.139 ms  41.210 ms  41.168 ms
 3  10.4.191.254  50.343 ms  53.117 ms  53.069 ms
 4  10.137.213.113  62.294 ms  65.350 ms  58.389 ms
 5  10.255.48.82  75.504 ms  71.210 ms  75.466 ms
 6  195.23.124.121  77.039 ms  76.854 ms  73.701 ms
 7  104.17.113.188  67.021 ms  44.657 ms  37.766 ms
```

Figure 18: Experiência 3 - Rota percorrida pelo pacote até atingir 104.17.113.188

Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	172.24.64.1	0.0.0.0	UG	0	0	0	eth0
172.24.64.0	0.0.0.0	255.255.240.0	U	0	0	0	eth0
<b>francisco@DESKTOP-GGJ0BB6:~\$</b>							

Figure 19: Experiência 3 - Rotas presentes no computador

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.24.64.244	104.17.113.188	UDP	74	36626 → 33434 Len=32
2	0.000008	172.24.64.244	104.17.113.188	UDP	74	34060 → 33435 Len=32
3	0.000008	172.24.64.244	104.17.113.188	UDP	74	57310 → 33436 Len=32
4	0.000072	172.24.64.1	172.24.64.244	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
5	0.000072	172.24.64.1	172.24.64.244	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
6	0.000128	172.24.64.1	172.24.64.244	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
7	0.000174	172.24.64.244	104.17.113.188	UDP	74	55409 → 33438 Len=32
8	0.000218	172.24.64.244	104.17.113.188	UDP	74	55489 → 33437 Len=32
9	0.000218	172.24.64.244	104.17.113.188	UDP	74	59470 → 33440 Len=32
10	0.000218	172.24.64.244	104.17.113.188	UDP	74	43529 → 33444 Len=32
11	0.000218	172.24.64.244	104.17.113.188	UDP	74	40573 → 33448 Len=32
12	0.000218	172.24.64.244	104.17.113.188	UDP	74	40056 → 33449 Len=32
13	0.000256	172.24.64.244	104.17.113.188	UDP	74	51923 → 33441 Len=32
14	0.000256	172.24.64.244	104.17.113.188	UDP	74	49476 → 33445 Len=32
15	0.000256	172.24.64.244	104.17.113.188	UDP	74	55109 → 33446 Len=32
16	0.000298	172.24.64.244	104.17.113.188	UDP	74	57813 → 33439 Len=32
17	0.000298	172.24.64.244	104.17.113.188	UDP	74	50196 → 33442 Len=32
18	0.000298	172.24.64.244	104.17.113.188	UDP	74	49345 → 33443 Len=32
19	0.000298	172.24.64.244	104.17.113.188	UDP	74	39616 → 33447 Len=32
20	0.000557	172.24.64.244	104.17.113.188	UDP	74	52111 → 33450 Len=32
21	0.000560	172.24.64.244	104.17.113.188	UDP	74	40444 → 33451 Len=32
22	0.000562	172.24.64.244	104.17.113.188	UDP	74	54915 → 33452 Len=32
23	0.020981	192.168.1.1	172.24.64.244	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
24	0.021261	172.24.64.244	104.17.113.188	UDP	74	52110 → 33453 Len=32
25	0.039833	192.168.1.1	172.24.64.244	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
26	0.039872	192.168.1.1	172.24.64.244	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
27	0.040258	172.24.64.244	104.17.113.188	UDP	74	45979 → 33454 Len=32
28	0.040258	172.24.64.244	104.17.113.188	UDP	74	52721 → 33455 Len=32
29	0.044374	10.4.191.254	172.24.64.244	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
30	0.044811	172.24.64.244	104.17.113.188	UDP	74	45588 → 33456 Len=32
31	0.049357	10.137.213.113	172.24.64.244	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
32	0.049739	172.24.64.244	104.17.113.188	UDP	74	56571 → 33457 Len=32
33	0.053497	10.4.191.254	172.24.64.244	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)

▶ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF\_{D34CCB4F-46C8-49F8-9258-1C3820597217}, id 0  
 ▶ Ethernet II, Src: Microsoft\_2a:97:14 (00:15:5d:2a:97:14), Dst: Microsoft\_78:2d:c3 (00:15:5d:78:2d:c3)  
 ▶ Internet Protocol Version 4, Src: 172.24.64.244, Dst: 104.17.113.188  
 ▶ User Datagram Protocol, Src Port: 36626, Dst Port: 33434  
 ▶ Data (32 bytes)

Figure 20: Experiência 3 - Resultado traceroute

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.24.73.31	142.250.200.142	ICMP	98	Echo (ping) request id=0x0751, seq=1/256, ttl=64 (reply in 2)
2	0.045260	142.250.200.142	172.24.73.31	ICMP	98	Echo (ping) reply id=0x0751, seq=1/256, ttl=114 (request i...)
3	1.002610	172.24.73.31	142.250.200.142	ICMP	98	Echo (ping) request id=0x0751, seq=2/512, ttl=64 (reply in 4)
4	1.054563	142.250.200.142	172.24.73.31	ICMP	98	Echo (ping) reply id=0x0751, seq=2/512, ttl=114 (request i...)
5	2.004728	172.24.73.31	142.250.200.142	ICMP	98	Echo (ping) request id=0x0751, seq=3/768, ttl=64 (reply in 6)
6	2.411352	142.250.200.142	172.24.73.31	ICMP	98	Echo (ping) reply id=0x0751, seq=3/768, ttl=114 (request i...)
7	3.0006536	172.24.73.31	142.250.200.142	ICMP	98	Echo (ping) request id=0x0751, seq=4/1024, ttl=64 (reply in ...)
8	3.057269	142.250.200.142	172.24.73.31	ICMP	98	Echo (ping) reply id=0x0751, seq=4/1024, ttl=114 (request ...)
9	4.008653	172.24.73.31	142.250.200.142	ICMP	98	Echo (ping) request id=0x0751, seq=5/1280, ttl=64 (reply in ...)
10	4.063119	142.250.200.142	172.24.73.31	ICMP	98	Echo (ping) reply id=0x0751, seq=5/1280, ttl=114 (request ...)
11	5.010340	172.24.73.31	142.250.200.142	ICMP	98	Echo (ping) request id=0x0751, seq=6/1536, ttl=64 (reply in ...)
12	5.086391	142.250.200.142	172.24.73.31	ICMP	98	Echo (ping) reply id=0x0751, seq=6/1536, ttl=114 (request ...)
13	6.012626	172.24.73.31	142.250.200.142	ICMP	98	Echo (ping) request id=0x0751, seq=7/1792, ttl=64 (reply in ...)
14	6.100988	142.250.200.142	172.24.73.31	ICMP	98	Echo (ping) reply id=0x0751, seq=7/1792, ttl=114 (request ...)
15	7.015199	172.24.73.31	142.250.200.142	ICMP	98	Echo (ping) request id=0x0751, seq=8/2048, ttl=64 (reply in ...)
16	7.326285	142.250.200.142	172.24.73.31	ICMP	98	Echo (ping) reply id=0x0751, seq=8/2048, ttl=114 (request ...)
17	8.017345	172.24.73.31	142.250.200.142	ICMP	98	Echo (ping) request id=0x0751, seq=9/2304, ttl=64 (reply in ...)
18	8.057834	142.250.200.142	172.24.73.31	ICMP	98	Echo (ping) reply id=0x0751, seq=9/2304, ttl=114 (request ...)
19	9.019135	172.24.73.31	142.250.200.142	ICMP	98	Echo (ping) request id=0x0751, seq=10/2560, ttl=64 (reply in ...)
20	9.065969	142.250.200.142	172.24.73.31	ICMP	98	Echo (ping) reply id=0x0751, seq=10/2560, ttl=114 (request ...)
21	10.020936	172.24.73.31	142.250.200.142	ICMP	98	Echo (ping) request id=0x0751, seq=11/2816, ttl=64 (reply in ...)
22	10.069241	142.250.200.142	172.24.73.31	ICMP	98	Echo (ping) reply id=0x0751, seq=11/2816, ttl=114 (request ...)
23	11.022970	172.24.73.31	142.250.200.142	ICMP	98	Echo (ping) request id=0x0751, seq=12/3072, ttl=64 (reply in ...)
24	11.075311	142.250.200.142	172.24.73.31	ICMP	98	Echo (ping) reply id=0x0751, seq=12/3072, ttl=114 (request ...)

▶ Frame 13: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface \Device\NPF\_{D34CCB4F-46C8-49F8-9258-1C3820597217}, id 0  
 ▶ Ethernet II, Src: Microsoft\_2a:93:23 (00:15:5d:2a:93:23), Dst: Microsoft\_2a:9f:9c (00:15:5d:2a:9f:9c)  
 ▶ Internet Protocol Version 4, Src: 172.24.73.31, Dst: 142.250.200.142  
 ▶ Internet Control Message Protocol

Figure 21: Experiência 3 - Resultado ping youtubas

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.24.73.31	172.24.64.1	DNS	75	Standard query 0x917c A enisa.europa.eu
2	0.000000	172.24.73.31	172.24.64.1	DNS	75	Standard query 0x7746 AAAA enisa.europa.eu
3	0.908024	172.24.64.1	172.24.73.31	DNS	106	Standard query response 0x917c A enisa.europa.eu A 212.146.10...
4	1.112141	172.24.64.1	172.24.73.31	DNS	118	Standard query response 0x7746 AAAA enisa.europa.eu AAAA 2001...
5	1.119729	172.24.73.31	212.146.105.104	ICMP	98	Echo (ping) request id=0x9b4b, seq=1/256, ttl=64 (reply in 6)
6	1.317111	212.146.105.104	172.24.73.31	ICMP	98	Echo (ping) reply id=0x9b4b, seq=1/256, ttl=48 (request in ...)
7	1.319060	172.24.73.31	172.24.64.1	DNS	88	Standard query 0x2925 PTR 104.105.146.212.in-addr.arpa
8	1.367505	172.24.64.1	224.0.0.251	MDNS	94	Standard query 0x0000 PTR 104.105.146.212.in-addr.arpa.local,...
9	1.368034	fe80::5114:7b1d:aa3.. ff02::fb		MDNS	114	Standard query 0x0000 PTR 104.105.146.212.in-addr.arpa.local,...
10	1.727612	172.24.64.1	172.24.73.31	DNS	210	Standard query response 0x2925 PTR 104.105.146.212.in-addr.ar...
11	2.119406	172.24.73.31	212.146.105.104	ICMP	98	Echo (ping) request id=0x9b4b, seq=2/512, ttl=64 (reply in 1...
12	2.228502	212.146.105.104	172.24.73.31	ICMP	98	Echo (ping) reply id=0x9b4b, seq=2/512, ttl=48 (request in ...)
13	2.229353	172.24.73.31	172.24.64.1	DNS	88	Standard query 0xb442 PTR 104.105.146.212.in-addr.arpa
14	2.230874	172.24.64.1	172.24.73.31	DNS	210	Standard query response 0xb442 PTR 104.105.146.212.in-addr.ar...
15	3.120877	172.24.73.31	212.146.105.104	ICMP	98	Echo (ping) request id=0x9b4b, seq=3/768, ttl=64 (reply in 1...
16	3.261534	212.146.105.104	172.24.73.31	ICMP	98	Echo (ping) reply id=0x9b4b, seq=3/768, ttl=48 (request in ...)
17	3.264053	172.24.73.31	172.24.64.1	DNS	88	Standard query 0x6682 PTR 104.105.146.212.in-addr.arpa
18	3.309608	172.24.64.1	172.24.73.31	DNS	210	Standard query response 0x6682 PTR 104.105.146.212.in-addr.ar...
19	4.122893	172.24.73.31	212.146.105.104	ICMP	98	Echo (ping) request id=0x9b4b, seq=4/1024, ttl=64 (reply in ...)
20	4.247308	212.146.105.104	172.24.73.31	ICMP	98	Echo (ping) reply id=0x9b4b, seq=4/1024, ttl=48 (request in ...)
21	4.248104	172.24.73.31	172.24.64.1	DNS	88	Standard query 0xd60b PTR 104.105.146.212.in-addr.arpa
22	4.249455	172.24.64.1	172.24.73.31	DNS	210	Standard query response 0xd60b PTR 104.105.146.212.in-addr.ar...
23	5.006199	Microsof_2a:93:23	Microsof_2a:9f:9c	ARP	42	Who has 172.24.64.1? Tell 172.24.73.31
24	5.006239	Microsof_2a:9f:9c	Microsof_2a:93:23	ARP	42	172.24.64.1 is at 00:15:5d:2a:9f:9c
25	5.124620	172.24.73.31	212.146.105.104	ICMP	98	Echo (ping) request id=0x9b4b, seq=5/1280, ttl=64 (reply in ...)
26	5.238750	212.146.105.104	172.24.73.31	ICMP	98	Echo (ping) reply id=0x9b4b, seq=5/1280, ttl=48 (request in ...)
27	5.239833	172.24.73.31	172.24.64.1	DNS	88	Standard query 0xfdf1 PTR 104.105.146.212.in-addr.arpa
28	5.241490	172.24.64.1	172.24.73.31	DNS	210	Standard query response 0xfdf1 PTR 104.105.146.212.in-addr.ar...
29	5.818371	Microsof_2a:9f:9c	Microsof_2a:93:23	ARP	42	Who has 172.24.73.31? Tell 172.24.64.1
30	5.818896	Microsof_2a:93:23	Microsof_2a:9f:9c	ARP	42	172.24.73.31 is at 00:15:5d:2a:93:23
▶ Frame 1: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface \Device\NPF_{D34CCB4F-46C8-49F8-9258-1C3820597217}, id 0						
▶ Ethernet II, Src: Microsof_2a:93:23 (00:15:5d:2a:93:23), Dst: Microsof_2a:9f:9c (00:15:5d:2a:9f:9c)						
▶ Internet Protocol Version 4, Src: 172.24.73.31, Dst: 172.24.64.1						
▶ User Datagram Protocol, Src Port: 43238, Dst Port: 53						
▶ Domain Name System (query)						

Figure 22: Experiência 3 - Resultado ping enisa.europa.eu

## 9.4 Experiência 4

81	128.4179.. KYE_04:20:99	Broadcast	ARP	42 Who has 172.16.11.1? Tell 172.16.11.253
82	128.4180.. HewlettP_61:2e:c3	KYE_04:20:99	ARP	60 172.16.11.1 is at 00:21:5a:61:2e:c3
83	128.4180.. 172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=1/256, ttl=63 (reply in 84)

Figure 23: Experiência 4 - tux4 pede MAC a tux2

84	129.1141.. HewlettP_61:2d:ef Broadcast	ARP	60 Who has 172.16.10.254? Tell 172.16.10.1
85	129.1141.. HewlettP_61:2f:24 HewlettP_61:2d:ef	ARP	42 172.16.10.254 is at 00:21:5a:61:2f:24
86	129.1142.. 172.16.10.1	172.16.11.1	ICMP

Figure 24: Experiência 4 - tux3 pede MAC a tux4

96	133.5166.. HewlettP_61:2e:c3 KYE_04:20:99	ARP	60 Who has 172.16.11.253? Tell 172.16.11.1
97	133.5166.. KYE_04:20:99	HewlettP_61:2e:c3 ARP	42 172.16.11.253 is at 00:c0:df:04:20:99
98	133.5205.. 172.16.10.1	172.16.11.1	ICMP

Figure 25: Experiência 4 - tux2 pede MAC a tux4

101	134.2352.. HewlettP_61:2f:24 HewlettP_61:2d:ef	ARP	42 Who has 172.16.10.1? Tell 172.16.10.254
102	134.2353.. HewlettP_61:2d:ef HewlettP_61:2f:24	ARP	60 172.16.10.1 is at 00:21:5a:61:2d:ef
103	134.2356.. Cisco_21:3a:84	Snapping-tree/ STP	60 Root = 32768/10/a0:cf:54:a1:3a:84 Cost =

Figure 26: Experiência 4 - tux4 pede MAC a tux3

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	Cisco_a1:3a:83	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...
2	2.000573257	Cisco_a1:3a:83	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...
3	2.516699295	Cisco_a1:3a:83	Cisco_a1:3a:83	LOOP	60	Reply
4	4.005471646	Cisco_a1:3a:83	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...
5	4.833157415	172.16.10.1	172.16.11.253	ICMP	98	Echo (ping) request id=0x090e, seq=1/256, ttl=64 (reply in 6)
6	4.833316234	172.16.11.253	172.16.10.1	ICMP	98	Echo (ping) reply id=0x090e, seq=1/256, ttl=64 (request in ...)
7	5.856099578	172.16.10.1	172.16.11.253	ICMP	98	Echo (ping) request id=0x090e, seq=2/512, ttl=64 (reply in 8)
8	5.856256931	172.16.11.253	172.16.10.1	ICMP	98	Echo (ping) reply id=0x090e, seq=2/512, ttl=64 (request in ...)
9	6.010413755	Cisco_a1:3a:83	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...
10	6.880073786	172.16.10.1	172.16.11.253	ICMP	98	Echo (ping) request id=0x090e, seq=3/768, ttl=64 (reply in 1...)
11	6.880203551	172.16.11.253	172.16.10.1	ICMP	98	Echo (ping) reply id=0x090e, seq=3/768, ttl=64 (request in ...)
12	7.904027297	172.16.10.1	172.16.11.253	ICMP	98	Echo (ping) request id=0x090e, seq=4/1024, ttl=64 (reply in ...)
13	7.904201435	172.16.11.253	172.16.10.1	ICMP	98	Echo (ping) reply id=0x090e, seq=4/1024, ttl=64 (request in ...)
14	8.015291470	Cisco_a1:3a:83	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...
15	8.928086945	172.16.10.1	172.16.11.253	ICMP	98	Echo (ping) request id=0x090e, seq=5/1280, ttl=64 (reply in ...)
16	8.928214475	172.16.11.253	172.16.10.1	ICMP	98	Echo (ping) reply id=0x090e, seq=5/1280, ttl=64 (request in ...)
17	9.901259232	HewlettP_61:2f:24	HewlettP_61:2d:ef	ARP	60	Who has 172.16.10.1? Tell 172.16.10.254
18	9.901267543	HewlettP_61:2d:ef	HewlettP_61:2f:24	ARP	42	172.16.10.1 is at 00:21:5a:61:2d:ef
19	9.952086645	172.16.10.1	172.16.11.253	ICMP	98	Echo (ping) request id=0x090e, seq=6/1536, ttl=64 (reply in ...)
20	9.952210613	172.16.11.253	172.16.10.1	ICMP	98	Echo (ping) reply id=0x090e, seq=6/1536, ttl=64 (request in ...)
21	10.024428665	Cisco_a1:3a:83	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...
22	10.048049826	HewlettP_61:2d:ef	HewlettP_61:2f:24	ARP	42	Who has 172.16.10.254? Tell 172.16.10.1
23	10.048156543	HewlettP_61:2f:24	HewlettP_61:2d:ef	ARP	60	172.16.10.254 is at 00:21:5a:61:2f:24
24	10.596503718	fe80::221:5aff:fe61...	ff02::fb	MDNS	180	Standard query 0x0000 PTR _ftp._tcp.local, "QM" question PTR ...
25	10.596588994	172.16.10.1	224.0.0.251	MDNS	160	Standard query 0x0000 PTR _ftp._tcp.local, "QM" question PTR ...
26	10.976089208	172.16.10.1	172.16.11.253	ICMP	98	Echo (ping) request id=0x090e, seq=7/1792, ttl=64 (replay in ...)

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

Figure 27: Experiência 4 - Resultado de tux3 ping tux4 via 253

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	Cisco_a1:3a:83	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...
2	0.646286584	Cisco_a1:3a:83	Cisco_a1:3a:83	LOOP	60	Reply
3	2.009127418	Cisco_a1:3a:83	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...
4	2.260310896	172.16.10.1	172.16.10.254	ICMP	98	Echo (ping) request id=0x08c1, seq=1/256, ttl=64 (reply in 5)
5	2.260472858	172.16.10.254	172.16.10.1	ICMP	98	Echo (ping) reply id=0x08c1, seq=1/256, ttl=64 (request in ...)
6	3.274948162	172.16.10.1	172.16.10.254	ICMP	98	Echo (ping) request id=0x08c1, seq=2/512, ttl=64 (reply in 7)
7	3.275082886	172.16.10.254	172.16.10.1	ICMP	98	Echo (ping) reply id=0x08c1, seq=2/512, ttl=64 (request in ...)
8	4.009789164	Cisco_a1:3a:83	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...
9	4.298950446	172.16.10.1	172.16.10.254	ICMP	98	Echo (ping) request id=0x08c1, seq=3/768, ttl=64 (reply in 1...)
10	4.299104306	172.16.10.254	172.16.10.1	ICMP	98	Echo (ping) reply id=0x08c1, seq=3/768, ttl=64 (request in ...)
11	5.322952171	172.16.10.1	172.16.10.254	ICMP	98	Echo (ping) request id=0x08c1, seq=4/1024, ttl=64 (reply in ...)
12	5.323087943	172.16.10.254	172.16.10.1	ICMP	98	Echo (ping) reply id=0x08c1, seq=4/1024, ttl=64 (request in ...)
13	6.014700682	Cisco_a1:3a:83	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...
14	6.346952640	172.16.10.1	172.16.10.254	ICMP	98	Echo (ping) request id=0x08c1, seq=5/1280, ttl=64 (reply in ...)
15	6.347079332	172.16.10.254	172.16.10.1	ICMP	98	Echo (ping) reply id=0x08c1, seq=5/1280, ttl=64 (request in ...)
16	6.921108823	Cisco_a1:3a:83	CDP/VTP/DTP/PAgP/UD...	CDP	448	Device ID: tux-sw1 Port ID: FastEthernet0/1
17	7.306912429	HewlettP_61:2d:ef	HewlettP_61:2f:24	ARP	42	Who has 172.16.10.254? Tell 172.16.10.1
18	7.307029826	HewlettP_61:2f:24	HewlettP_61:2d:ef	ARP	60	172.16.10.254 is at 00:21:5a:61:2f:24
19	7.370942422	172.16.10.1	172.16.10.254	ICMP	98	Echo (ping) request id=0x08c1, seq=6/1536, ttl=64 (reply in ...)
20	7.371066600	172.16.10.254	172.16.10.1	ICMP	98	Echo (ping) reply id=0x08c1, seq=6/1536, ttl=64 (request in ...)
21	7.414697009	HewlettP_61:2f:24	HewlettP_61:2d:ef	ARP	60	Who has 172.16.10.1? Tell 172.16.10.254
22	7.414702037	HewlettP_61:2d:ef	HewlettP_61:2f:24	ARP	42	172.16.10.1 is at 00:21:5a:61:2d:ef
23	8.019583356	Cisco_a1:3a:83	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...

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

Figure 28: Experiência 4 - Resultado de tux3 ping tux4 via 254

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	Cisco_a1:3a:83	Spanning-tree-(for... STP	60 Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...		
2	2.000522203	Cisco_a1:3a:83	Spanning-tree-(for... STP	60 Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...		
3	2.276434105	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x092c, seq=1/256, ttl=64 (reply in 4)	
4	2.276737426	172.16.11.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x092c, seq=1/256, ttl=63 (request in ...	
5	3.282975510	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x092c, seq=2/512, ttl=64 (reply in 6)	
6	3.283218418	172.16.11.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x092c, seq=2/512, ttl=63 (request in ...	
7	4.005427017	Cisco_a1:3a:83	Spanning-tree-(for... STP	60 Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...		
8	4.306978143	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x092c, seq=3/768, ttl=64 (reply in 9)	
9	4.307257019	172.16.11.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x092c, seq=3/768, ttl=63 (request in ...	
10	4.446064266	Cisco_a1:3a:83	Cisco_a1:3a:83	LOOP	60 Reply	
11	5.330960383	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x092c, seq=4/1024, ttl=64 (reply in ...	
12	5.331201405	172.16.11.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x092c, seq=4/1024, ttl=63 (request in ...	
13	6.010327780	Cisco_a1:3a:83	Spanning-tree-(for... STP	60 Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...		
14	6.354962807	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x092c, seq=5/1280, ttl=64 (reply in ...	
15	6.355202572	172.16.11.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x092c, seq=5/1280, ttl=63 (request in ...	
16	8.015253337	Cisco_a1:3a:83	Spanning-tree-(for... STP	60 Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...		

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

Figure 29: Experiência 4 - Resultado de tux3 ping tux2

No.	Time	Source	Destination	Protocol	Length	Info
79	122.283647177	Cisco_a1:3a:84	Cisco_a1:3a:84	LOOP	60 Reply	
80	122.296358569	Cisco_a1:3a:84	Spanning-tree-(for... STP	60 Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...		
81	124.301167631	Cisco_a1:3a:84	Spanning-tree-(for... STP	60 Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...		
82	126.310673379	Cisco_a1:3a:84	Spanning-tree-(for... STP	60 Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...		
83	128.311473690	Cisco_a1:3a:84	Spanning-tree-(for... STP	60 Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...		
84	129.114110185	HewlettP_61:2d:ef	Broadcast	ARP	60 Who has 172.16.10.254? Tell 172.16.10.1	
85	129.114137702	HewlettP_61:2d:ef	HewlettP_61:2d:ef	ARP	42 172.16.10.254 is at 00:21:5a:61:2d:24	
86	129.114259222	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=1/256, ttl=64 (reply in 8...	
87	129.114514356	172.16.11.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x09bd, seq=1/256, ttl=63 (request in ...	
88	130.120960078	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=2/512, ttl=64 (reply in 8...	
89	130.121110865	172.16.11.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x09bd, seq=2/512, ttl=63 (request in ...	
90	130.319938314	Cisco_a1:3a:84	Spanning-tree-(for... STP	60 Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...		
91	131.144940781	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=3/768, ttl=64 (reply in 9...	
92	131.145086330	172.16.11.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x09bd, seq=3/768, ttl=63 (request in ...	
93	132.168917364	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=4/1024, ttl=64 (reply in ...	
94	132.169094830	172.16.11.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x09bd, seq=4/1024, ttl=63 (request in ...	
95	132.278876947	Cisco_a1:3a:84	Cisco_a1:3a:84	LOOP	60 Reply	
96	132.320576175	Cisco_a1:3a:84	Spanning-tree-(for... STP	60 Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...		
97	133.192896740	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=5/1280, ttl=64 (reply in ...	
98	133.193044035	172.16.11.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x09bd, seq=5/1280, ttl=63 (request in ...	
99	134.216878700	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=6/1536, ttl=64 (reply in ...	
100	134.217017265	172.16.11.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x09bd, seq=6/1536, ttl=63 (request in ...	
101	134.235244164	HewlettP_61:2d:ef	HewlettP_61:2d:ef	ARP	42 Who has 172.16.10.1? Tell 172.16.10.254	
102	134.235362196	HewlettP_61:2d:ef	HewlettP_61:2d:ef	ARP	60 172.16.10.1 is at 00:21:5a:61:2d:ef	
103	134.325695261	Cisco_a1:3a:84	Spanning-tree-(for... STP	60 Conf. Root = 32768/10:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...		
104	135.240859124	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=7/1792, ttl=64 (reply in ...	

Frame 84: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0  
 Ethernet II, Src: HewlettP\_61:2d:ef (00:21:5a:61:2d:ef), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
 Address Resolution Protocol (request)

Figure 30: Experiência 4 - Resultado em tux4 eth0 de tux3 ping tux2

No.	Time	Source	Destination	Protocol	Length	Info
75	120.291437203	Cisco_a1:3a:87	Spanning-tree-(for... STP	60 Conf.	Root = 32768/11:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...	
76	121.587065243	Cisco_a1:3a:87	Cisco_a1:3a:87	LOOP	60 Reply	
77	122.300434367	Cisco_a1:3a:87	Spanning-tree-(for... STP	60 Conf.	Root = 32768/11:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...	
78	124.301055117	Cisco_a1:3a:87	Spanning-tree-(for... STP	60 Conf.	Root = 32768/11:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...	
79	126.305796641	Cisco_a1:3a:87	Spanning-tree-(for... STP	60 Conf.	Root = 32768/11:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...	
80	128.310594109	Cisco_a1:3a:87	Spanning-tree-(for... STP	60 Conf.	Root = 32768/11:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...	
81	128.417939743	Kye_04:20:99	Broadcast	ARP	42 Who has 172.16.11.1? Tell 172.16.11.253	
82	128.418056029	HewlettP_61:2e:c3	Kye_04:20:99	ARP	60 172.16.11.1 is at 00:21:5a:61:2e:c3	
83	128.418069438	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=1/256, ttl=63 (reply in 8...	
84	128.418167425	172.16.10.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x09bd, seq=1/256, ttl=64 (request in...	
85	129.424645414	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=2/512, ttl=63 (reply in 8...	
86	129.424756741	172.16.10.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x09bd, seq=2/512, ttl=64 (request in...	
87	130.315439208	Cisco_a1:3a:87	Spanning-tree-(for... STP	60 Conf.	Root = 32768/11:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...	
88	130.448627235	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=3/768, ttl=63 (reply in 8...	
89	130.448737095	172.16.10.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x09bd, seq=3/768, ttl=64 (request in...	
90	131.472603398	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=4/1024, ttl=63 (reply in ...	
91	131.472740985	172.16.10.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x09bd, seq=4/1024, ttl=64 (request in...	
92	131.582837179	Cisco_a1:3a:87	Cisco_a1:3a:87	LOOP	60 Reply	
93	132.320831864	Cisco_a1:3a:87	Spanning-tree-(for... STP	60 Conf.	Root = 32768/11:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...	
94	132.496582635	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=5/1280, ttl=63 (reply in ...	
95	132.496693403	172.16.10.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x09bd, seq=5/1280, ttl=64 (request i...	
96	133.516625898	HewlettP_61:2e:c3	Kye_04:20:99	ARP	60 Who has 172.16.11.253? Tell 172.16.11.1	
97	133.516642588	Kye_04:20:99	HewlettP_61:2e:c3	ARP	42 172.16.11.253 is at 00:c0:df:04:20:99	
98	133.520565433	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=6/1536, ttl=63 (reply in ...	
99	133.520673128	172.16.10.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x09bd, seq=6/1536, ttl=64 (request i...	
100	134.325150498	Cisco_a1:3a:87	Spanning-tree-(for... STP	60 Conf.	Root = 32768/11:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...	
101	134.544544250	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=7/1792, ttl=63 (reply in ...	
102	134.544653692	172.16.10.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x09bd, seq=7/1792, ttl=64 (request i...	
103	135.568521741	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=8/2048, ttl=63 (reply in ...	
104	135.568634116	172.16.10.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x09bd, seq=8/2048, ttl=64 (request i...	
105	136.334172457	Cisco_a1:3a:87	Spanning-tree-(for... STP	60 Conf.	Root = 32768/11:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...	
106	136.592500698	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=9/2304, ttl=63 (reply in ...	
107	136.592634724	172.16.10.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x09bd, seq=9/2304, ttl=64 (request i...	
108	137.616489773	172.16.10.1	172.16.11.1	ICMP	98 Echo (ping) request id=0x09bd, seq=10/2560, ttl=63 (reply in ...	
109	137.616594265	172.16.10.1	172.16.10.1	ICMP	98 Echo (ping) reply id=0x09bd, seq=10/2560, ttl=64 (request i...	
110	138.335079555	Cisco_a1:3a:87	Spanning-tree-(for... STP	60 Conf.	Root = 32768/11:a0:cf:5b:a1:3a:80 Cost = 0 Port = 0x8...	

▶ Frame 82: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth1, id 0  
 ▶ Ethernet II, Src: HewlettP\_61:2e:c3 (00:21:5a:61:2e:c3), Dst: Kye\_04:20:99 (00:c0:df:04:20:99)  
 ▶ Address Resolution Protocol (reply)

Figure 31: Experiência 4 - Resultado em tux4 eth1 de tux3 ping tux2