2023/24

Computer Networks

2º Trabalho Laboratorial

Redes de Computadores

Turma 1

Licenciatura em Engenharia Informática e Computação

João Pedro Moreira Costa up202108714@.up.pt
Rafael Neves Teixeira up202108831@up.pt

Índice

1. Sumário

Este relatório foi desenvolvido no âmbito da unidade curricular RCOM (Redes de Computadores) do 1º semestre do 3º ano de Licenciatura em Engenharia Informática e Computação na FEUP. Este relatório contém como principal foco o tema abordado no 2º Trabalho Laboratorial, isto é, uma rede de computadores cujo objetivo é estabelecer e perceber os conceitos por detrás de um sistema de transferência de dados entre máquinas.

O trabalho prático ficou concluído com êxito pelo que todos os objetivos definidos foram cumpridos.

2. Introdução

A primeira parte desta atividade laboratorial consistiu na criação de uma aplicação capaz de realizar download de ficheiros da Internet utilizando o protocolo FTP com base nas especificações estabelecidas pelo guião fornecido.

A segunda parte consistiu no registo e concretização das experiências propostas e em responder às questões que nos iam colocando após realizar cada experiência.

Este relatório encontra-se dividido nas seguintes secções:

- **Introdução:** Apresentação do trabalho que nos foi proposto
- **Aplicação Download:** Explicação técnica da aplicação e demonstração de resultados e testes utilizados
- **Configuração e análise de dados:** Identificação das experiências propostas e respostas às questões colocadas
- **Anexos:** Imagens dos logs e código de implementação da aplicação

3. Aplicação Download

3.1. Arquitetura

A aplicação desenvolvida realiza o download de ficheiros através de um protocolo FTP.

Para tal, a aplicação começa por dar *parse* ao url atribuído ao programa da aplicação, dividindo-o em campos username e password (se existentes), host, path e filename. No caso de não serem especificados os campos de username e password, é atribuído um username e uma password “anonymous”.

De seguida, é retirado o endereço IP com a função **getIP()** e posteriormente todo o processo de download é iniciado dentro da função **connectionDownload()**.

Dentro dessa função, começamos por criar uma *socket*, **sockfd**, para controlo de comandos com a função **createSocket()** e entramos dentro de um ciclo while cujo objetivo é tratar da receção dos códigos e aplicar a operação mais adequada ao pedido. Mais especificamente, de acordo com cada código, fazemos o tratamento da informação da seguinte forma:

- Código 220: Escrevemos o username na socket, **write(sockfd, userLogin, strlen(userLogin));**

- Código 331: Significa que o username foi aceite. Escrevemos a password do user na socket, **write(sockfd, passwdLogin, strlen(passwdLogin));**
- Código 230: Significa que o user está autenticado. Escrevemos na socket uma solicitação para entrar no modo passivo, **write(sockfd, "pasv\r\n", 6);**
- Código 227: Fazemos o tratamento do modo passivo. Para isso retiramos o número da porta com a função **getPortNumber()**, criamos uma nova socket, **sockfd2**, para transferência de dados, e escrevemos na socket de controlo, **sockfd**, o path do url fornecido **write(sockfd, retrvPath, strlen(retrvPath));**
- Código 150: Neste momento, o ficheiro está pronto a ser aberto, por isso, abrimos um ficheiro com o filename fornecido no url para escrita.
- Código 226: Código para tratamento do download. Neste, lemos os dados recebidos pela socket responsável pelos dados, **sockfd2**, e escrevemo-los no ficheiro aberto anteriormente para receção dos dados.

Para terminar, fechamos o ficheiro que recebeu os dados e as sockets de controlo e de dados e verificamos que, após todo este processo, o ficheiro contém a informação desejada.

3.2. Testes e Resultados

Para testar a nossa aplicação, sujeitamo-la a vários testes. Estes testes englobam a transferência de ficheiros com diferentes tamanhos, a atribuição de url errados com o objetivo de deteção de erros e entrada na aplicação com o modo anónimo e não anónimo.

Diante esses testes, o programa desempenhou a sua função e concluiu todas as operações esperadas com sucesso. Para compilar e correr o programa basta escrever as seguintes linhas de código:

- gcc -o download download.c
- ./download <url>

Nas imagens seguintes, é demonstrado o output obtido após correr a aplicação com os url's: *ftp://rcom:rcom@netlab1.fe.up.pt/pipe.txt* e *ftp://rcom:rcom@netlab1.fe.up.pt/files/pic1.jpg*, respetivamente:

4. Configuração e análise de dados

4.1. Experiência 1 - Configurar rede IP

Objetivo: Dar-nos a entender o que são endereços MAC e endereços IP, o que são pacotes ARP e a sua função, que tipo de pacotes é que o comando *ping* gera e ainda como realizar a configuração das máquinas tux53 e tux54.

Quais comandos são necessários para configurar esta experiência?

- ifconfig <eth> <ipAdress>/<mask>: para atribuição de um endereço a uma máquina tux
- route -n: para verificar as routes disponíveis na máquina tux
- ping <destinationIPAdress>: para gerar o ping da máquina atual para a máquina destino
- arp -a: para verificar as ligações das tabelas ARP

O que são pacotes ARP e para que são utilizados?

ARP (Address Resolution Protocol) tem como objetivo mapear o endereço IP de uma máquina a um endereço MAC da máquina na rede local. Basicamente, quando um computador tentar enviar um pacote a outra máquina, será enviado um pacote ARP para todas as máquinas locais de forma a descobrir qual delas possui um endereço MAC idêntico ao IP do destinatário. De forma a conseguir realizar a transferência de pacotes, ao ser determinado o destinatário, este irá enviar outro pacote ARP que indica à máquina qual o seu endereço MAC.

Quais são os endereços MAC e IP de pacotes ARP e porquê?

O endereço IP da tuxY3 na nossa experiência corresponde a 172.16.50.1, é o endereço fonte e o seu endereço MAC 00:21:5a:61:2c:54.

Já o endereço IP da tuxY4 corresponde a 172.16.50.254, é o endereço destino e o seu endereço MAC a 00:22:64:19:09:5c.

No mesmo pacote ARP são enviados 2 endereços IP nomeadamente da máquina fonte (tux53: 172.16.50.1) e da máquina destino (tux54: 172.16.50.254). De seguida, o tux54 devolve como resposta outro pacote ARP com o seu endereço IP e o seu endereço MAC (00:22:64:19:09:5c).

Estes endereços podem ser verificados usando o comando ifconfig ou através dos logs do Wireshark.

Quais pacotes é que o comando ping gera?

Gera inicialmente pacotes ARP, enquanto não obtém o endereço MAC da tux destino, e pacotes ICMP (Internet Control Message Protocol), após estabelecer ligação com o endereço MAC.

Quais são os endereços MAC e IP dos pacotes gerados pelo comando ping?

tux53 -> tux54:

- IP fonte: 172.16.50.1
- MAC fonte: 00:21:5a:61:2c:54
- IP destinatário: 172.16.50.254
- MAC destinatário: 00:22:64:19:09:5c

tux54 -> tux53:

- IP fonte: 172.16.50.254
- MAC fonte: 00:22:64:19:09:5c
- IP destinatário: 172.16.50.1
- MAC destinatário: 00:21:5a:61:2c:54

Como determinar se um frame recetor é ARP, IP, ICMP?

Através da visualização dos logs do Wireshark, na coluna “Protocol”, é possível verificar o tipo de frame que é recebida.

Alternativamente, analisa-se o Ethernet header da trama, isto é, caso este possua o valor 0x800 significa que a trama é do tipo IP ou ICMP porém caso possua o valor 0x806 é do tipo ARP. Para distinguir se o tipo de protocolo é IP ou ICMP, caso o IP header tome o valor 1, então podemos determinar que é do tipo ICMP.

Como determinar o comprimento de uma frame recebida?

Através dos logs do Wireshark de uma coluna com o nome “Length” que demonstra o tamanho da frame recebida em bytes.

O que é uma interface *loopback* e porque é importante?

Uma interface loopback corresponde a uma interface virtual que permite a um computador receber respostas de si mesmo para testar a configuração das ligações de rede, daí ser bastante útil e importante.

4.2. Experiência 2 - Implementação de duas bridges num switch

Objetivo: Entender como se efetua a configuração de bridges e como influenciam a troca de informação entre computadores. Nesta experiência realizamos a criação de duas bridges no switch com a atribuição dos computadores tux53 e tux54 à bridge50 e o computador tux52 à bridge51.

Como configurar bridgeY0?

No contexto desta experiência:

1. Criamos uma bridgeY0 com o comando (Y representa o número da workbench):
    - /interface bridge add name=bridgeY0
2. Removemos as portas que devemos associar à bridgeY0 (1 - tux53 e 2 - tux54) da bridge default usando os seguintes comandos, sendo X em etherX o número da porta:
    - /interface bridge port remove [find interface =ether1]
    - /interface bridge port remove [find interface =ether2]
3. Adicionamos as portas da tux53 e tux54 à bridgeY0 com o comando:
    - /interface bridge port add bridge=bridgeY0 interface=ether1
    - /interface bridge port add bridge=bridgeY0 interface=ether2

Quantos domínios broadcast existem? Como é possível concluir isso a partir dos logs?

Existem 2 domínios devido às duas bridges criadas. É possível verificar isso nos logs pois apenas as portas pertencentes à bridge são abrangidas, isto é, o ping do tux53 obtém uma resposta do tux54 mas não do tux52 pois este computador é inalcançável já que pertence a outra bridge e não possui um meio de conexão com a outra bridge existente.

4.3. Experiência 3 - Configurar um Router em Linux

Objetivo: Configurar o tuxY4 de forma a que este computador funcione como um router. Estabelecer bridges e atribuir as respectivas portas nomeadamente tuxY3 e tuxY4 à bridgeY0 e tux52 e tuxY4 à bridgeY1, permitindo a ligação de dois computadores anteriormente inalcançáveis (tux53 e tux52).

Quais são os comandos necessários para configurar esta experiência?

1. Configurar o tux54.eth1, tomando em conta que todas as entradas das experiências anteriores estão configuradas:
    - ifconfig eth1 up
    - ifconfig eth1 172.16.50.253/24
    - ifconfig eth1
2. Adicionar a porta do tux54.eth1 (porta 3 no nosso caso) à bridgeY1, tendo em conta que já foi criada na experiência 2:
    - /interface bridge port add bridge=bridgeY1 interface=ether3
3. Ativar IP forwarding e desativar ICMP echo-ignore-broadcast:
    - sysctl net.ipv4.ip_forward=1
    - sysctl net.ipv4.icmp_echo_ignore_broadcasts=0

Quais routes estão nos *tuxes*? Qual é o seu significado?

A tux53 tem uma route para a bridge50, a tux52 para a bridge51 e a tux54 para ambas. O gateway tanto do tux52 como do tux53 é o tux54 já que este é comum a ambas as bridges (bridge50 e bridge51).

Na tux53 adicionou-se uma route, **route add -net 172.16.51.0/24 gw 172.16.50.254**, de forma a quando a tux53 enviar um ping para a bridge51, ela vai utilizar como gateway o tux54. Na tux54 fez-se algo similar, isto é, adicionou-se uma route, **route add -net 172.16.50.0/24 gw 172.16.51.253**, de forma a quando enviarmos um ping para a bridge50, este será primeiro enviado para o router (172.16.51.253)

Que informação contém uma entrada da *forwarding table*?

Cada entrada possui, de componentes principais, um **destinatário**, correspondente ao IP da máquina destino, uma **gateway**, correspondente ao IP do computador para o qual vai ser enviada a mensagem e onde se irá verificar o processo de routing para o destinatário, e uma **interface** correspondente à placa de rede utilizada para o envio da mensagem.

Para além disso, possui outro tipo de dados disponíveis nomeadamente uma **Genmask**, **Flags** que fornecem informações sobre a route, **Metric** que revela o custo da route, **Ref** que demonstra o número de referências para esta route e **Use** que conta o número de pesquisas pela rota dependendo do uso de -F (número de falhas de cache) ou -C (número de sucessos)

Que mensagens ARP e endereços MAC associados são observados e porquê?

Quando o tux53 não reconhece o endereço MAC do tux52, após ping do tux53 para o tux52, é ocorrida uma troca de mensagens ARP do seguinte formato:

- Who has <IPtuxY_X1>? Tell [<IPtuxY_X2**>**]”
- <IPtuxY_X1> is at <MACtuxY_X1>”

Esta troca de mensagens ARP contém apenas os endereços MAC do tux53 e tux54 e não do tux52 devido ao facto de o tux53 não reconhecer uma ligação direta com ele, mas sim apenas o gateway que o direciona para o tux52.

Quais pacotes ICMP são observados e porquê?

Podem ser observados dois tipos de ICMP packets, nomeadamente do tipo *request* e *reply*, isto porque todos os tuxes adicionados reconhecem a ligação uns dos outros. Os pacotes transmitidos nesta experiência contêm sempre como endereço source o IP do tux53 e como endereço de destino o IP do tux52.

Quais são os endereços IP e MAC que estão associados ao pacotes ICMP e porquê?

Os endereços IP e MAC que estão associados aos pacotes ICMP correspondem aos endereços IP e MAC das máquinas que enviam/recebem os pacotes de dados.

Nesta experiência cada pacote ICMP, obtido através do ping no tux53, contém como endereço source o IP do tux53, como endereço destino o IP do tux54 e contém o endereço MAC do tux54 que estabelece a ligação entre os dois computadores pelas bridges.

4.4. Experiência 4 - Configurar um Router Comercial e Implementar NAT

Objetivo: Configurar um router comercial através da ligação deste à rede do laboratório e à bridge51. Aplicar a técnica NAT no router de forma a garantir que a conexão entre as máquinas rede IP e a internet seja possível.

Como configurar uma route estática num router comercial?

1. Dar reset às configurações iniciais do router
2. Adicionar a porta do router ligado ao switch à bridgeY1
3. Atribuir IP's interno e externo no GTKTerm

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

Sem a ligação do tuxY2 ao tuxY4, os pacotes de dados sofrem uma operação de ICMP redirects, isto é, são encaminhados até ao endereço IP de destino através do router, ou seja, o tuxY2, com utilização do **Rc**, consegue enviar os pacotes de informação para o tuxY4.

Com a ligação, como existe uma conexão direta entre estes, não foi necessário qualquer tipo de redirecionamento.

Como configurar NAT num router comercial?

Utilizando os comandos:

- /ip firewall nat disable 0: Para desativar o default nat
- /ip firewall nat enable 0: Para ativar o default nat

O que é que o NAT faz?

NAT (Network Address Translation) corresponde a um protocolo que procura associar um IP de um endereço noutro endereço de IP, assegurando a privacidade e segurança das máquinas numa subrede privada local que comunicam com máquinas externas já que esta técnica esconde o endereço origem/destino original dos pacotes

enviados. Esta técnica permite que redes locais se conectem com redes públicas e permite com que máquinas inseridas numa subrede privada local, que se conectam com máquinas exteriores, sejam reconhecidas por um IP único que engloba todos os dispositivos da mesma.

Entrando no contexto da experiência, caso o tux53 (máquina de rede local) tente enviar um pacote para um endereço de uma rede pública, ex: 172.16.2.254, o pacote é enviado para o NAT que modifica o seu endereço origem para o seu endereço exterior (172.16.1.59), escondendo a operação de que é o tux53 quem realiza o envio. Assim, o pacote é enviado para 172.16.2.254 e o pacote de resposta possui o endereço destino de 172.16.1.59, que precisa ser alterado para permitir ao sistema saber qual a verdadeira identidade do computador a que se quer enviar este pacote. Por isso, o NAT, ao receber este pacote, de forma a permitir a comunicação entre computadores, antes de enviar o pacote de volta para o tux53, altera o destinatário do pacote para o seu verdadeiro endereço.

4.5. Experiência 5 - DNS

Objetivo: Analisar os diferentes comportamentos das máquinas ao conectá-las a um servidor DNS que realiza traduções de hostnames para endereços IP

Como configurar o serviço DNS num host?

Adicionando a linha de comando “nameserver <ipServer>” no file /etc/resolv.conf em cada tux.

Quais pacotes são trocados pelo DNS e que informação é transportada?

A informação transportada corresponde a pacotes do tipo DNS. O router identifica o IP de destino destes pacotes e traduz-lo.

4.6. Experiência 6 - Conexões TCP

Objetivo: Fornecer uma compreensão prática dos conceitos relacionados ao protocolo TCP, controlo de erros, controlo de congestionamento e como o tráfego de rede é afetado por múltiplas conexões simultâneas.

Quantas conexões TCP são abertas pela aplicação FTP?

Uma para o envio de comandos e outra para a receção do ficheiro, ou seja, duas.

Em qual conexão é transportada a informação de controlo FTP?

Na conexão TCP responsável pelo envio de comandos. É nesta que ocorre a transferência de comandos de controlo e recepção de mensagens do servidor.

Quais são as fases de uma conexão TCP?

1. DNS
2. [SYN, ACK] (sincronização)
3. Configuração
4. Transferência do ficheiro

5. [FIN, ACK] (finalização)

Como é que o mecanismo ARQ TCP funciona? Quais são os campos TCP relevantes? Qual informação relevante pode ser observada nos logs?

O mecanismo ARQ (Automatic Repeat reQuest) no TCP (Transmission Control Protocol) é responsável por garantir uma comunicação segura entre dois pontos finais de uma rede. Este permite que o receptor detete a perda de pacotes e solicite a retransmissão desses pacotes ao remetente.

É possível ver os números de sequência, usados para ordenar os pacotes e detetar pacotes perdidos, e os ACKs associados a cada pacote, simbolizando a confirmação da receção dos dados. Se o remetente não receber um ACK durante um determinado *timeout*, temporizador para determinar quando considerar um pacote como perdido, o TCP assume que o segmento foi perdido e o retransmite.

Como é que o mecanismo de controlo de congestionamento funciona? Quais são os campos relevantes? Como é que a conexão de dados evolui ao longo do tempo? Esta está de acordo com o mecanismo de controlo de congestionamento?

O mecanismo *WindowSize* no cabeçalho TCP representa a quantidade de dados que um remetente pode enviar antes de receber uma confirmação (ACK) do receptor, sendo o tamanho desta janela ajustado dinamicamente para controlar o fluxo de dados.

Na conexão, um parâmetro chamado *CongestionWindow* determina a capacidade de comunicação do emissor, influenciando a decisão de enviar mais ou menos pacotes. Com isto, quando o nível de congestionamento da rede diminui, a *CongestionWindow* aumenta, e vice-versa.

As conexões de dados TCP são perturbadas pela aparência de uma segunda conexão TCP? Como?

A introdução de uma segunda conexão TCP afeta as conexões de dados TCP, pois, ao estabelecer múltiplas conexões TCP, a banda de largura é dividida entre elas, o que acaba por reduzir a velocidade de cada uma.

5. Anexos

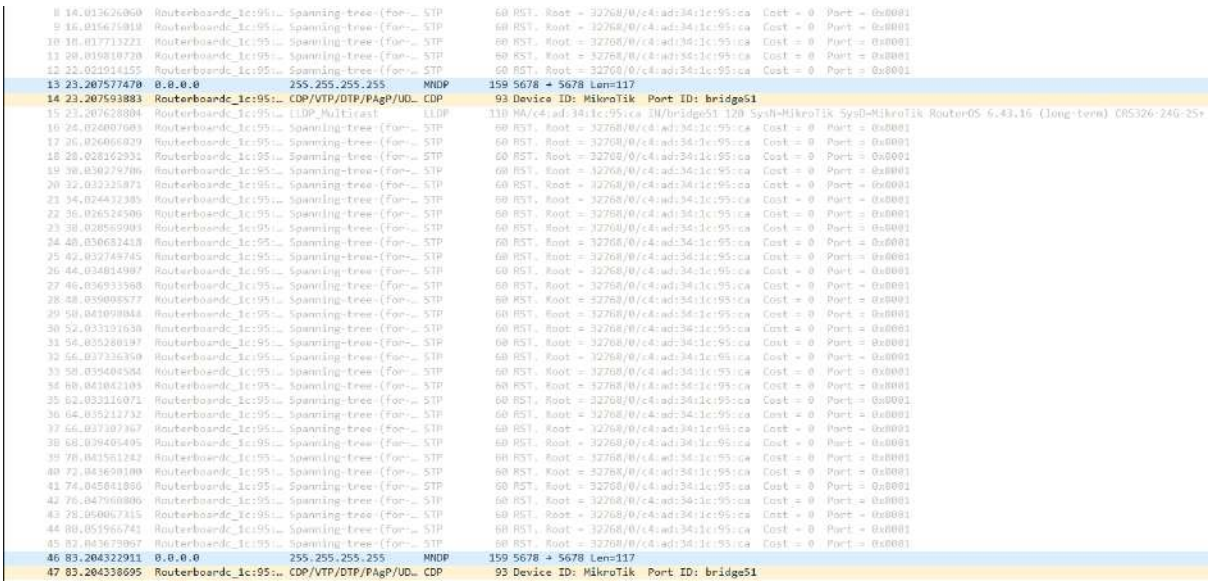
5.1. Logs/Imagens

1	0.000000000	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
2	2.002105684	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
3	4.004186783	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
4	6.005924825	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
5	8.008031975	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
6	10.010120827	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
7	11.051331201	HewlettPacka_61:2c:…	Broadcast	ARP	42 Who has 172.16.50.254? Tell 172.16.50.1
8	11.051431981	HewlettPacka_19:09:…	HewlettPacka_61:2c:…	ARP	60 172.16.50.254 is at 00:22:64:19:09:5c
9	11.051439664	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x370e, seq=1/256, ttl=64 (reply in 10)
10	11.051527523	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x370e, seq=1/256, ttl=64 (request in 9)
11	12.012200110	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
12	12.058858282	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x370e, seq=2/512, ttl=64 (reply in 13)
13	12.058960110	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x370e, seq=2/512, ttl=64 (request in 12)
14	13.082827485	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x370e, seq=3/768, ttl=64 (reply in 15)
15	13.082930640	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x370e, seq=3/768, ttl=64 (request in 14)
16	14.014310194	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
17	14.106820155	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x370e, seq=4/1024, ttl=64 (reply in 18)
18	14.106949640	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x370e, seq=4/1024, ttl=64 (request in 17)
19	15.130810660	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x370e, seq=5/1280, ttl=64 (reply in 20)
20	15.130909415	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x370e, seq=5/1280, ttl=64 (request in 19)
21	16.016055359	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
22	16.134083320	HewlettPacka_19:09:…	HewlettPacka_61:2c:…	ARP	60 Who has 172.16.50.1? Tell 172.16.50.254
23	16.134098197	HewlettPacka_61:2c:…	HewlettPacka_19:09:…	ARP	42 172.16.50.1 is at 00:21:5a:61:2c:54
24	16.154810104	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x370e, seq=6/1536, ttl=64 (reply in 25)
25	16.154901246	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x370e, seq=6/1536, ttl=64 (request in 24)
26	17.178822679	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x370e, seq=7/1792, ttl=64 (reply in 27)
27	17.178926532	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x370e, seq=7/1792, ttl=64 (request in 26)
28	18.018147424	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
29	18.202822752	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x370e, seq=8/2048, ttl=64 (reply in 30)
30	18.202924789	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x370e, seq=8/2048, ttl=64 (request in 29)

Tabela 1: Experiência 1 step 7, exemplo de log do Wireshark

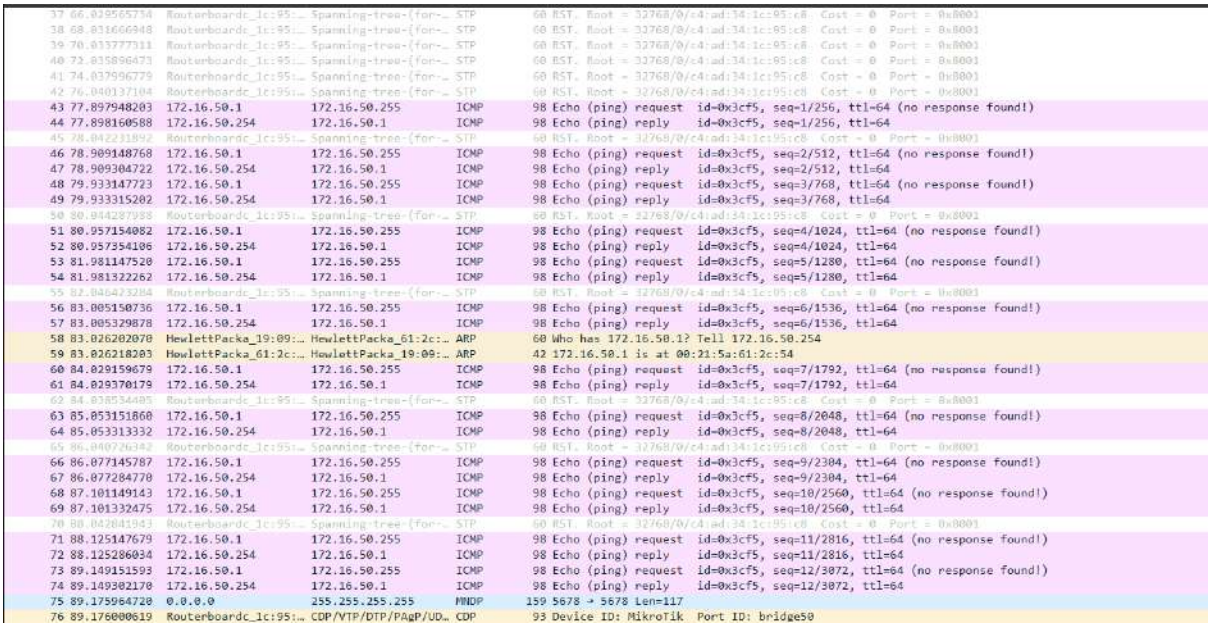
8	13.914373086	fe80::221:5aff:fe61…	ff02::fb	MDNS	180 Standard query 0x0000 PTR _nfs._tcp.local, "QM" question PTR _ftp._tcp.local, "QM" question PTR _webdav._tcp.local, "QM" questi…
15	25.081324278	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x3b51, seq=1/256, ttl=64 (reply in 16)
16	25.081499299	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3b51, seq=1/256, ttl=64 (request in 15)
18	26.088962336	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x3b51, seq=2/512, ttl=64 (reply in 19)
19	26.089098595	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3b51, seq=2/512, ttl=64 (request in 18)
20	27.112951933	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x3b51, seq=3/768, ttl=64 (reply in 21)
21	27.113089170	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3b51, seq=3/768, ttl=64 (request in 20)
23	28.136958012	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x3b51, seq=4/1024, ttl=64 (reply in 24)
24	28.137086379	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3b51, seq=4/1024, ttl=64 (request in 23)
25	29.160955431	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x3b51, seq=5/1280, ttl=64 (reply in 26)
26	29.161122700	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3b51, seq=5/1280, ttl=64 (request in 25)
28	30.107966326	HewlettPacka_19:09:…	HewlettPacka_61:2c:…	ARP	60 Who has 172.16.50.1? Tell 172.16.50.254
29	30.107981831	HewlettPacka_61:2c:…	HewlettPacka_19:09:…	ARP	42 172.16.50.1 is at 00:21:5a:61:2c:54
30	30.184947472	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x3b51, seq=6/1536, ttl=64 (reply in 31)
31	30.185076049	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3b51, seq=6/1536, ttl=64 (request in 30)
32	30.312919432	HewlettPacka_61:2c:…	HewlettPacka_19:09:…	ARP	42 Who has 172.16.50.254? Tell 172.16.50.1
33	30.313037043	HewlettPacka_19:09:…	HewlettPacka_61:2c:…	ARP	60 172.16.50.254 is at 00:22:64:19:09:5c
34	31.208954390	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x3b51, seq=7/1792, ttl=64 (reply in 35)
35	31.209090509	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3b51, seq=7/1792, ttl=64 (request in 34)
37	32.232962774	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x3b51, seq=8/2048, ttl=64 (reply in 38)
38	32.233089814	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3b51, seq=8/2048, ttl=64 (request in 37)
39	33.256959145	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x3b51, seq=9/2304, ttl=64 (reply in 40)
40	33.257141779	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3b51, seq=9/2304, ttl=64 (request in 39)
42	34.280950907	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x3b51, seq=10/2560, ttl=64 (reply in 43)
43	34.281103789	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3b51, seq=10/2560, ttl=64 (request in 42)

Tabela 2: Experiência 2 step 5 ping 54, exemplo de log do Wireshark



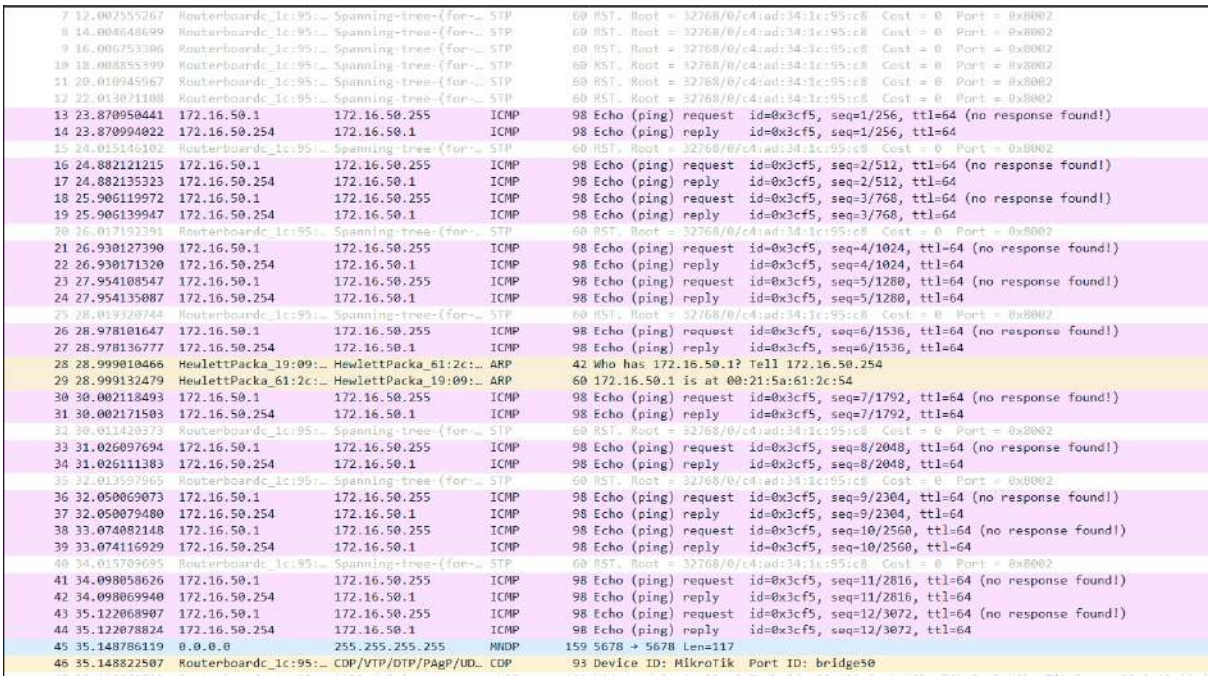
13	23.207577470	0.0.0.0	255.255.255.255	MNDP	159 5678 → 5678 Len=117
14	23.207593883	Routerboardc_1c:95:…	CDP/VTP/DTP/PAgP/UD…	CDP	93 Device ID: MikroTik Port ID: bridge51
46	83.204322911	0.0.0.0	255.255.255.255	MNDP	159 5678 → 5678 Len=117
47	83.204338695	Routerboardc_1c:95:…	CDP/VTP/DTP/PAgP/UD…	CDP	93 Device ID: MikroTik Port ID: bridge51

Tabela 3: Experiência 2 step 8 ping 52, exemplo de log do Wireshark



43	77.897948203	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=1/256, ttl=64 (no response found!)
44	77.898160588	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=1/256, ttl=64
46	78.909148768	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=2/512, ttl=64 (no response found!)
47	78.909304722	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=2/512, ttl=64
48	79.933147723	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=3/768, ttl=64 (no response found!)
49	79.933315202	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=3/768, ttl=64
51	80.957154082	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=4/1024, ttl=64 (no response found!)
52	80.957354106	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=4/1024, ttl=64
53	81.981147520	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=5/1280, ttl=64 (no response found!)
54	81.981322262	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=5/1280, ttl=64
56	83.005150736	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=6/1536, ttl=64 (no response found!)
57	83.005329878	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=6/1536, ttl=64
58	83.026202070	HewlettPacka_19:09:…	HewlettPacka_61:2c:…	ARP	60 Who has 172.16.50.1? Tell 172.16.50.254
59	83.026218203	HewlettPacka_61:2c:…	HewlettPacka_19:09:…	ARP	42 172.16.50.1 is at 00:21:5a:61:2c:54
60	84.029159679	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=7/1792, ttl=64 (no response found!)
61	84.029370179	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=7/1792, ttl=64
63	85.053151860	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=8/2048, ttl=64 (no response found!)
64	85.053313332	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=8/2048, ttl=64
66	86.077145787	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=9/2304, ttl=64 (no response found!)
67	86.077284770	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=9/2304, ttl=64
68	87.101149143	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=10/2560, ttl=64 (no response found!)
69	87.101332475	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=10/2560, ttl=64
71	88.125147679	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=11/2816, ttl=64 (no response found!)
72	88.125286034	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=11/2816, ttl=64
73	89.149151593	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=12/3072, ttl=64 (no response found!)
74	89.149302170	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=12/3072, ttl=64
75	89.175964720	0.0.0.0	255.255.255.255	MNDP	159 5678 → 5678 Len=117
76	89.176000619	Routerboardc_1c:95:…	CDP/VTP/DTP/PAgP/UD…	CDP	93 Device ID: MikroTik Port ID: bridge50

Tabela 4: Experiência 2 step 8 ping 53, exemplo de log do Wireshark

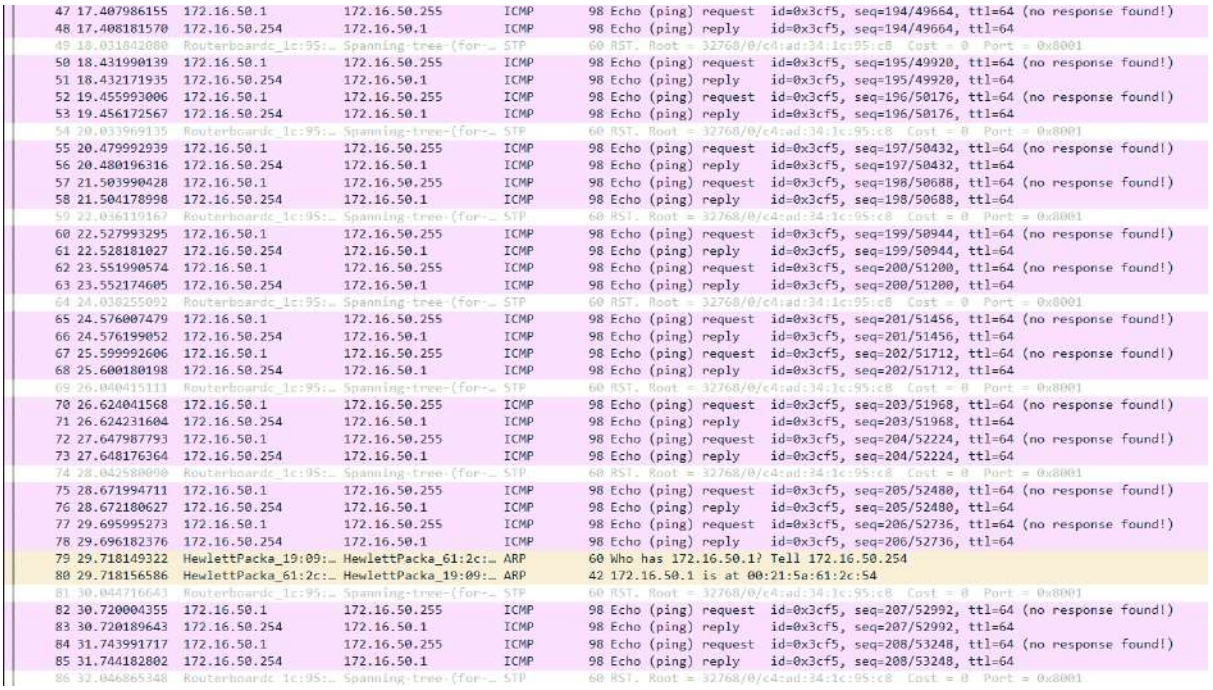


7	12.002555267	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
8	14.004648699	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
9	16.006753306	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
10	18.008855399	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
11	20.010945967	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
12	22.013071108	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
13	23.870950441	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=1/256, ttl=64 (no response found!)
14	23.870994022	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=1/256, ttl=64
15	24.015146102	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
16	24.882121215	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=2/512, ttl=64 (no response found!)
17	24.882135323	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=2/512, ttl=64
18	25.906119972	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=3/768, ttl=64 (no response found!)
19	25.906139947	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=3/768, ttl=64
20	26.017152391	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
21	26.930127390	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=4/1024, ttl=64 (no response found!)
22	26.930171320	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=4/1024, ttl=64
23	27.954108547	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=5/1280, ttl=64 (no response found!)
24	27.954135087	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=5/1280, ttl=64
25	28.019320744	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
26	28.978101647	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=6/1536, ttl=64 (no response found!)
27	28.978136777	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=6/1536, ttl=64
28	28.999010466	HewlettPacka_19:09:…	HewlettPacka_61:2c:…	ARP	42 Who has 172.16.50.1? Tell 172.16.50.254
29	28.999132479	HewlettPacka_61:2c:…	HewlettPacka_19:09:…	ARP	60 172.16.50.1 is at 00:21:5a:61:2c:54
30	30.002118493	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=7/1792, ttl=64 (no response found!)
31	30.002171503	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=7/1792, ttl=64
32	30.011420373	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
33	31.026097694	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=8/2048, ttl=64 (no response found!)
34	31.026111383	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=8/2048, ttl=64
35	32.013597965	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
36	32.050069073	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=9/2304, ttl=64 (no response found!)
37	32.050079480	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=9/2304, ttl=64
38	33.074082148	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=10/2560, ttl=64 (no response found!)
39	33.074116929	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=10/2560, ttl=64
40	34.015709695	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
41	34.098058626	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=11/2816, ttl=64 (no response found!)
42	34.098069940	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=11/2816, ttl=64
43	35.122068907	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=12/3072, ttl=64 (no response found!)
44	35.122078824	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=12/3072, ttl=64
45	35.148786119	0.0.0.0	255.255.255.255	MNDP	159 5678 → 5678 Len=117
46	35.148822507	Routerboardc_1c:95:…	CDP/VTP/DTP/PAgP/UD…	CDP	93 Device ID: MikroTik Port ID: bridge50

Tabela 5: Experiência 2 step 8 ping 54, exemplo de log do Wireshark

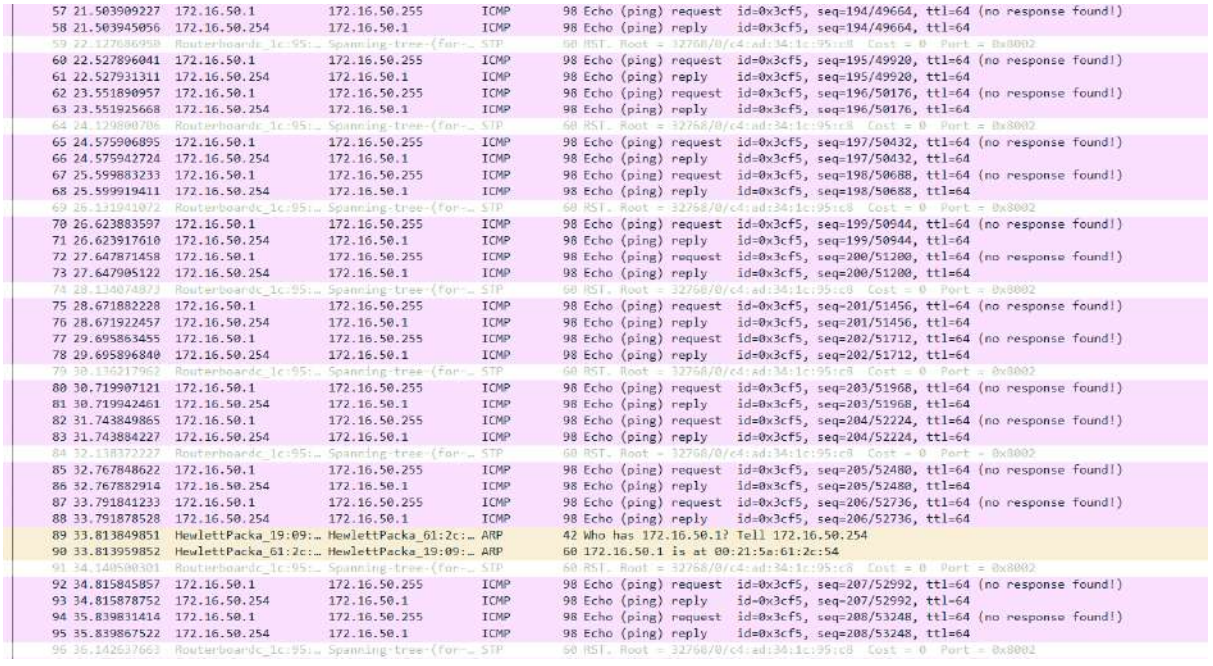
7	6.006376726	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:ca Cost = 0 Port = 0x8001
8	7.998515717	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:ca Cost = 0 Port = 0x8001
9	10.000618105	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:ca Cost = 0 Port = 0x8001
10	12.002744868	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:ca Cost = 0 Port = 0x8001
11	14.004854589	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:ca Cost = 0 Port = 0x8001
12	15.933340317	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=1/256, ttl=64 (no response found!)
13	16.006991294	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:ca Cost = 0 Port = 0x8001
14	16.939252871	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=2/512, ttl=64 (no response found!)
15	17.963252786	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=3/768, ttl=64 (no response found!)
16	18.009137797	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:ca Cost = 0 Port = 0x8001
17	18.987256962	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=4/1024, ttl=64 (no response found!)
18	20.011253664	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:ca Cost = 0 Port = 0x8001
19	20.011255969	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=5/1280, ttl=64 (no response found!)
20	21.035254069	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=6/1536, ttl=64 (no response found!)
21	22.013374700	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:ca Cost = 0 Port = 0x8001
22	22.059249723	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=7/1792, ttl=64 (no response found!)
23	23.083254109	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=8/2048, ttl=64 (no response found!)
24	24.015499437	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:ca Cost = 0 Port = 0x8001
25	24.107248367	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=9/2304, ttl=64 (no response found!)
26	25.131256593	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=10/2560, ttl=64 (no response found!)
27	26.017626759	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:ca Cost = 0 Port = 0x8001
28	26.155260699	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=11/2816, ttl=64 (no response found!)
29	27.179254818	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=12/3072, ttl=64 (no response found!)
30	28.019779921	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:ca Cost = 0 Port = 0x8001
31	28.203253196	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=13/3328, ttl=64 (no response found!)
32	29.227252902	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=14/3584, ttl=64 (no response found!)
33	30.021851718	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:ca Cost = 0 Port = 0x8001
34	30.251254075	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=15/3840, ttl=64 (no response found!)
35	31.275255457	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=16/4096, ttl=64 (no response found!)
36	32.023921560	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:ca Cost = 0 Port = 0x8001
37	32.299252019	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=17/4352, ttl=64 (no response found!)
38	33.323254868	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=18/4608, ttl=64 (no response found!)
39	34.025965280	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:ca Cost = 0 Port = 0x8001
40	34.347253037	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=19/4864, ttl=64 (no response found!)
41	35.371251696	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=20/5120, ttl=64 (no response found!)
42	36.028044690	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:ca Cost = 0 Port = 0x8001
43	36.395253566	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=21/5376, ttl=64 (no response found!)
44	37.419273876	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=22/5632, ttl=64 (no response found!)
45	38.030127731	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:ca Cost = 0 Port = 0x8001
46	38.443249486	172.16.51.1	172.16.51.255	ICMP	98 Echo (ping) request id=0x33f0, seq=23/5888, ttl=64 (no response found!)

Tabela 6: Experiência 2 step 10 ping 52, exemplo de log do Wireshark



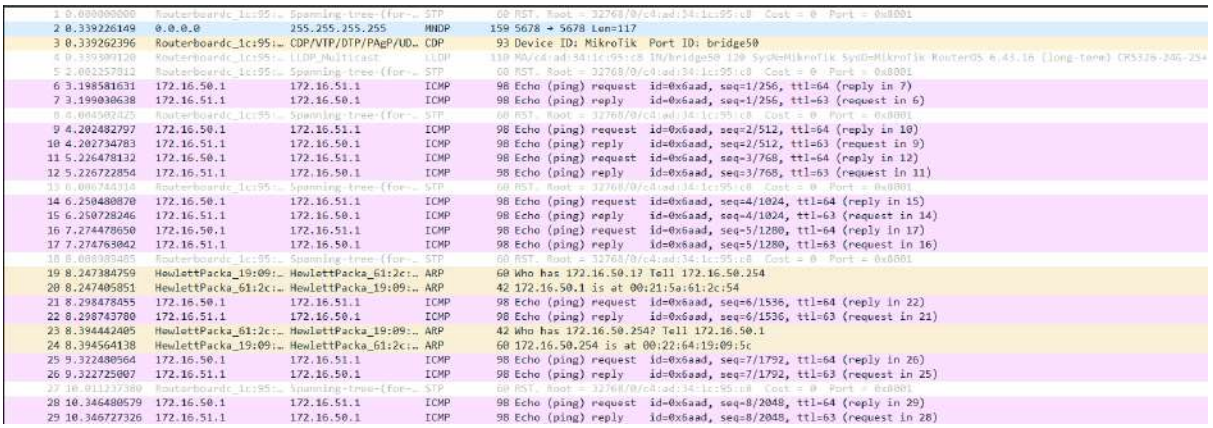
47	17.407986155	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=194/49664, ttl=64 (no response found!)
48	17.408181570	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=194/49664, ttl=64
49	18.031842080	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
50	18.431990139	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=195/49920, ttl=64 (no response found!)
51	18.432171935	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=195/49920, ttl=64
52	19.455993006	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=196/50176, ttl=64 (no response found!)
53	19.456172567	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=196/50176, ttl=64
54	20.033969135	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
55	20.479992939	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=197/50432, ttl=64 (no response found!)
56	20.480196316	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=197/50432, ttl=64
57	21.503990428	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=198/50688, ttl=64 (no response found!)
58	21.504178998	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=198/50688, ttl=64
59	22.036119167	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
60	22.527993295	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=199/50944, ttl=64 (no response found!)
61	22.528181027	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=199/50944, ttl=64
62	23.551990574	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=200/51200, ttl=64 (no response found!)
63	23.552174605	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=200/51200, ttl=64
64	24.038255092	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
65	24.576007479	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=201/51456, ttl=64 (no response found!)
66	24.576199052	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=201/51456, ttl=64
67	25.599992606	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=202/51712, ttl=64 (no response found!)
68	25.600180198	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=202/51712, ttl=64
69	26.040415111	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
70	26.624041568	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=203/51968, ttl=64 (no response found!)
71	26.624231604	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=203/51968, ttl=64
72	27.647987793	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=204/52224, ttl=64 (no response found!)
73	27.648176364	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=204/52224, ttl=64
74	28.042580000	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
75	28.671994711	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=205/52480, ttl=64 (no response found!)
76	28.672180627	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=205/52480, ttl=64
77	29.695995273	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=206/52736, ttl=64 (no response found!)
78	29.696182376	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=206/52736, ttl=64
79	29.718149322	HewlettPacka_19:09:…	HewlettPacka_61:2c:…	ARP	60 Who has 172.16.50.1? Tell 172.16.50.254
80	29.718156586	HewlettPacka_61:2c:…	HewlettPacka_19:09:…	ARP	42 172.16.50.1 is at 00:21:5a:61:2c:54
81	30.044716643	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
82	30.720004355	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=207/52992, ttl=64 (no response found!)
83	30.720189643	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=207/52992, ttl=64
84	31.743991717	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=208/53248, ttl=64 (no response found!)
85	31.744182802	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=208/53248, ttl=64
86	32.046865348	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001

Tabela 7: Experiência 2 step 10 ping 53, exemplo de log do Wireshark



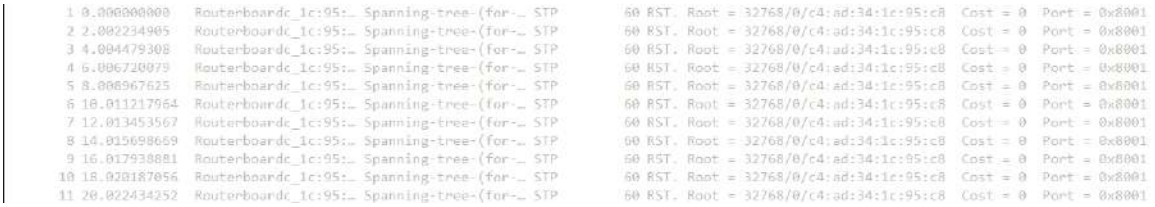
57	21.503909227	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=194/49664, ttl=64 (no response found!)
58	21.503945056	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=194/49664, ttl=64
59	22.127686950	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
60	22.527896041	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=195/49920, ttl=64 (no response found!)
61	22.527931311	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=195/49920, ttl=64
62	23.551890957	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=196/50176, ttl=64 (no response found!)
63	23.551925668	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=196/50176, ttl=64
64	24.129800706	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
65	24.575906895	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=197/50432, ttl=64 (no response found!)
66	24.575942724	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=197/50432, ttl=64
67	25.599883233	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=198/50688, ttl=64 (no response found!)
68	25.599919411	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=198/50688, ttl=64
69	26.131941072	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
70	26.623883597	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=199/50944, ttl=64 (no response found!)
71	26.623917610	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=199/50944, ttl=64
72	27.647871458	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=200/51200, ttl=64 (no response found!)
73	27.647905122	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=200/51200, ttl=64
74	28.134074873	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
75	28.671882228	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=201/51456, ttl=64 (no response found!)
76	28.671922457	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=201/51456, ttl=64
77	29.695863455	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=202/51712, ttl=64 (no response found!)
78	29.695896840	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=202/51712, ttl=64
79	30.136217962	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
80	30.719907121	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=203/51968, ttl=64 (no response found!)
81	30.719942461	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=203/51968, ttl=64
82	31.743849865	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=204/52224, ttl=64 (no response found!)
83	31.743884227	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=204/52224, ttl=64
84	32.138372227	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
85	32.767848622	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=205/52480, ttl=64 (no response found!)
86	32.767882914	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=205/52480, ttl=64
87	33.791841233	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=206/52736, ttl=64 (no response found!)
88	33.791878528	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=206/52736, ttl=64
89	33.813849851	HewlettPacka_19:09:…	HewlettPacka_61:2c:…	ARP	42 Who has 172.16.50.1? Tell 172.16.50.254
90	33.813959852	HewlettPacka_61:2c:…	HewlettPacka_19:09:…	ARP	60 172.16.50.1 is at 00:21:5a:61:2c:54
91	34.140500301	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
92	34.815845857	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=207/52992, ttl=64 (no response found!)
93	34.815878752	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=207/52992, ttl=64
94	35.839831414	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=208/53248, ttl=64 (no response found!)
95	35.839867522	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=208/53248, ttl=64
96	36.142637663	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002

Tabela 8: Experiência 2 step 10 ping 54, exemplo de log do Wireshark



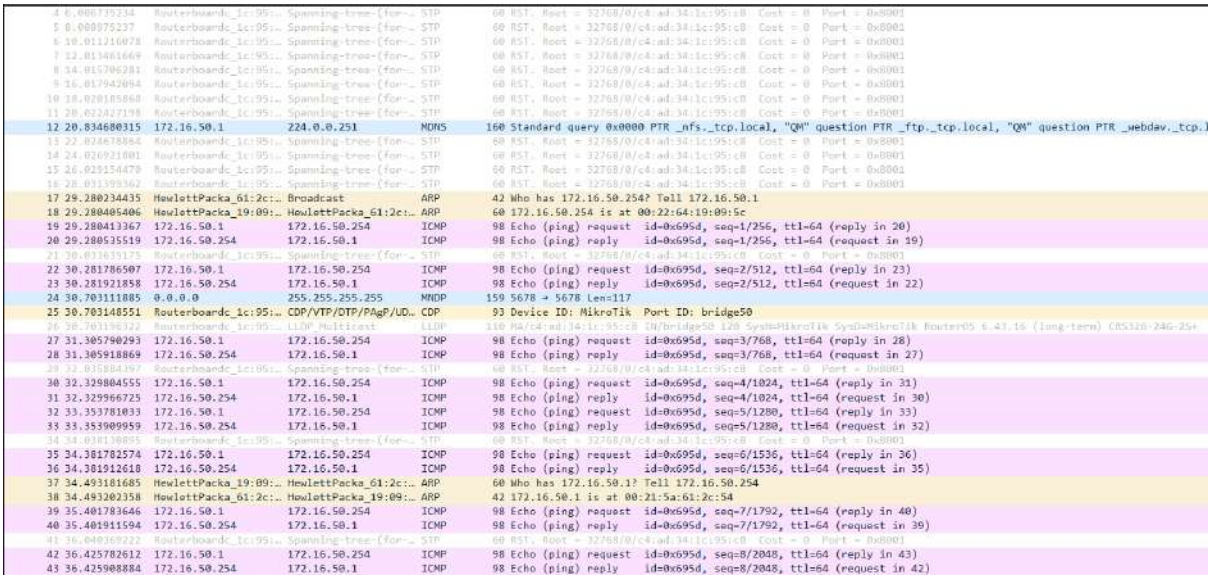
2	0.339226149	0.0.0.0	255.255.255.255	MNDP	159 5678 → 5678 Len=117
3	0.339262396	Routerboardc_1c:95:…	CDP/VTP/DTP/PAgP/UD…	CDP	93 Device ID: MikroTik Port ID: bridge50
6	3.198581631	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6aad, seq=1/256, ttl=64 (reply in 7)
7	3.199030638	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6aad, seq=1/256, ttl=63 (request in 6)
9	4.202482797	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6aad, seq=2/512, ttl=64 (reply in 10)
10	4.202734783	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6aad, seq=2/512, ttl=63 (request in 9)
11	5.226478132	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6aad, seq=3/768, ttl=64 (reply in 12)
12	5.226722854	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6aad, seq=3/768, ttl=63 (request in 11)
14	6.250480870	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6aad, seq=4/1024, ttl=64 (reply in 15)
15	6.250728246	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6aad, seq=4/1024, ttl=63 (request in 14)
16	7.274478650	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6aad, seq=5/1280, ttl=64 (reply in 17)
17	7.274763042	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6aad, seq=5/1280, ttl=63 (request in 16)
19	8.247384759	HewlettPacka_19:09:…	HewlettPacka_61:2c:…	ARP	60 Who has 172.16.50.1? Tell 172.16.50.254
20	8.247405851	HewlettPacka_61:2c:…	HewlettPacka_19:09:…	ARP	42 172.16.50.1 is at 00:21:5a:61:2c:54
21	8.298478455	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6aad, seq=6/1536, ttl=64 (reply in 22)
22	8.298743780	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6aad, seq=6/1536, ttl=63 (request in 21)
23	8.394442405	HewlettPacka_61:2c:…	HewlettPacka_19:09:…	ARP	42 Who has 172.16.50.254? Tell 172.16.50.1
24	8.394564138	HewlettPacka_19:09:…	HewlettPacka_61:2c:…	ARP	60 172.16.50.254 is at 00:22:64:19:09:5c
25	9.322480564	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6aad, seq=7/1792, ttl=64 (reply in 26)
26	9.322725007	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6aad, seq=7/1792, ttl=63 (request in 25)
28	10.346480579	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6aad, seq=8/2048, ttl=64 (reply in 29)
29	10.346727326	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6aad, seq=8/2048, ttl=63 (request in 28)

Tabela 9: Experiência 3 step 6 ping 172.16.51.1, exemplo de log do Wireshark



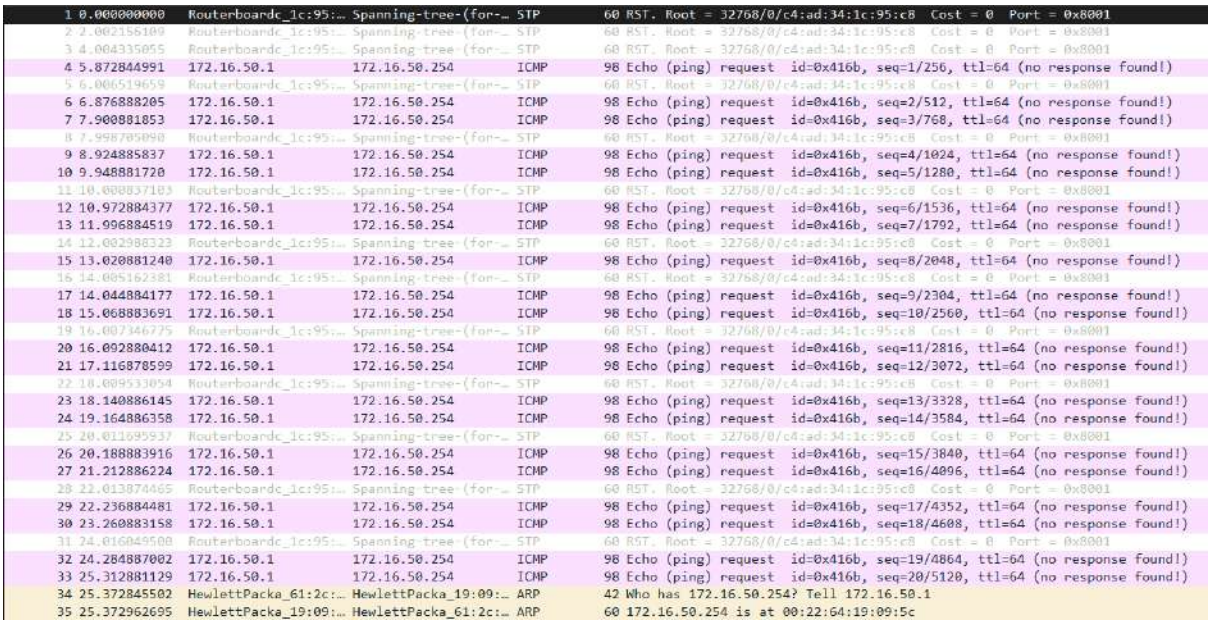
1	0.000000000	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
2	2.002234905	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
3	4.004479308	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
4	6.006720079	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
5	8.008967625	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
6	10.011217964	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
7	12.013453567	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
8	14.015698669	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
9	16.017938881	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
10	18.020187056	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
11	20.022434252	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001

Tabela 10: Experiência 3 step 6 ping 172.16.51.253, exemplo de log do Wireshark



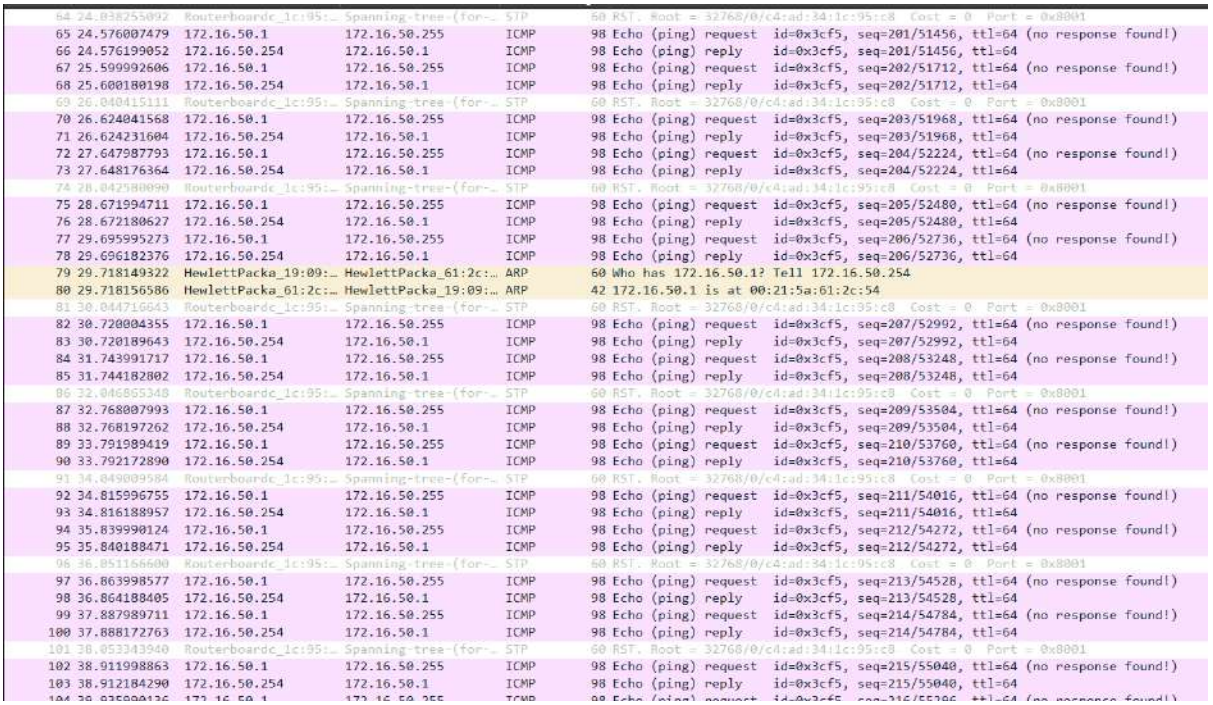
12	20.834680315	172.16.50.1	224.0.0.251	MDNS	160 Standard query 0x0000 PTR _nfs._tcp.local, "QM" question PTR _ftp._tcp.local, "QM" question PTR _webdav._tcp.…
17	29.280234435	HewlettPacka_61:2c:…	Broadcast	ARP	42 Who has 172.16.50.254? Tell 172.16.50.1
18	29.280405406	HewlettPacka_19:09:…	HewlettPacka_61:2c:…	ARP	60 172.16.50.254 is at 00:22:64:19:09:5c
19	29.280413367	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x695d, seq=1/256, ttl=64 (reply in 20)
20	29.280535519	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x695d, seq=1/256, ttl=64 (request in 19)
22	30.281786507	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x695d, seq=2/512, ttl=64 (reply in 23)
23	30.281921858	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x695d, seq=2/512, ttl=64 (request in 22)
24	30.703111885	0.0.0.0	255.255.255.255	MNDP	159 5678 → 5678 Len=117
25	30.703148551	Routerboardc_1c:95:…	CDP/VTP/DTP/PAgP/UD…	CDP	93 Device ID: MikroTik Port ID: bridge50
27	31.305790293	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x695d, seq=3/768, ttl=64 (reply in 28)
28	31.305918869	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x695d, seq=3/768, ttl=64 (request in 27)
30	32.329804555	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x695d, seq=4/1024, ttl=64 (reply in 31)
31	32.329966725	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x695d, seq=4/1024, ttl=64 (request in 30)
32	33.353781033	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x695d, seq=5/1280, ttl=64 (reply in 33)
33	33.353909959	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x695d, seq=5/1280, ttl=64 (request in 32)
35	34.381782574	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x695d, seq=6/1536, ttl=64 (reply in 36)
36	34.381912618	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x695d, seq=6/1536, ttl=64 (request in 35)
37	34.493181685	HewlettPacka_19:09:…	HewlettPacka_61:2c:…	ARP	60 Who has 172.16.50.1? Tell 172.16.50.254
38	34.493202358	HewlettPacka_61:2c:…	HewlettPacka_19:09:…	ARP	42 172.16.50.1 is at 00:21:5a:61:2c:54
39	35.401783646	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x695d, seq=7/1792, ttl=64 (reply in 40)
40	35.401911594	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x695d, seq=7/1792, ttl=64 (request in 39)
42	36.425782612	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x695d, seq=8/2048, ttl=64 (reply in 43)
43	36.425908884	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x695d, seq=8/2048, ttl=64 (request in 42)

Tabela 11: Experiência 3 step 6 ping 172.16.50.254, exemplo de log do Wireshark



1	0.000000000	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
2	2.002156109	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
3	4.004335055	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
4	5.872844991	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=1/256, ttl=64 (no response found!)
5	6.006519659	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
6	6.876888205	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=2/512, ttl=64 (no response found!)
7	7.900881853	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=3/768, ttl=64 (no response found!)
8	7.998705090	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
9	8.924885837	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=4/1024, ttl=64 (no response found!)
10	9.948881720	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=5/1280, ttl=64 (no response found!)
11	10.000837103	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
12	10.972884377	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=6/1536, ttl=64 (no response found!)
13	11.996884519	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=7/1792, ttl=64 (no response found!)
14	12.002988323	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
15	13.020881240	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=8/2048, ttl=64 (no response found!)
16	14.005162381	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
17	14.044884177	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=9/2304, ttl=64 (no response found!)
18	15.068883691	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=10/2560, ttl=64 (no response found!)
19	16.007346775	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
20	16.092880412	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=11/2816, ttl=64 (no response found!)
21	17.116878599	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=12/3072, ttl=64 (no response found!)
22	18.009533054	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
23	18.140886145	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=13/3328, ttl=64 (no response found!)
24	19.164886358	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=14/3584, ttl=64 (no response found!)
25	20.011695937	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
26	20.188883916	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=15/3840, ttl=64 (no response found!)
27	21.212886224	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=16/4096, ttl=64 (no response found!)
28	22.013874465	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
29	22.236884481	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=17/4352, ttl=64 (no response found!)
30	23.260883158	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=18/4608, ttl=64 (no response found!)
31	24.016049500	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
32	24.284887002	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=19/4864, ttl=64 (no response found!)
33	25.312881129	172.16.50.1	172.16.50.254	ICMP	98 Echo (ping) request id=0x416b, seq=20/5120, ttl=64 (no response found!)
34	25.372845502	HewlettPacka_61:2c:…	HewlettPacka_19:09:…	ARP	42 Who has 172.16.50.254? Tell 172.16.50.1
35	25.372962695	HewlettPacka_19:09:…	HewlettPacka_61:2c:…	ARP	60 172.16.50.254 is at 00:22:64:19:09:5c

Tabela 12: Experiência 3 step 7 ping 50_254, exemplo de log do Wireshark



64	24.038255092	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
65	24.576007479	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=201/51456, ttl=64 (no response found!)
66	24.576199052	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=201/51456, ttl=64
67	25.599992606	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=202/51712, ttl=64 (no response found!)
68	25.600180198	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=202/51712, ttl=64
69	26.040415111	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
70	26.624041568	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=203/51968, ttl=64 (no response found!)
71	26.624231604	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=203/51968, ttl=64
72	27.647987793	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=204/52224, ttl=64 (no response found!)
73	27.648176364	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=204/52224, ttl=64
74	28.042580090	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
75	28.671994711	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=205/52480, ttl=64 (no response found!)
76	28.672180627	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=205/52480, ttl=64
77	29.695995273	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=206/52736, ttl=64 (no response found!)
78	29.696182376	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=206/52736, ttl=64
79	29.718149322	HewlettPacka_19:09:…	HewlettPacka_61:2c:…	ARP	60 Who has 172.16.50.1? Tell 172.16.50.254
80	29.718156586	HewlettPacka_61:2c:…	HewlettPacka_19:09:…	ARP	42 172.16.50.1 is at 00:21:5a:61:2c:54
81	30.044716643	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
82	30.720004355	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=207/52992, ttl=64 (no response found!)
83	30.720189643	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=207/52992, ttl=64
84	31.743991717	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=208/53248, ttl=64 (no response found!)
85	31.744182802	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=208/53248, ttl=64
86	32.046865348	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
87	32.768007993	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=209/53504, ttl=64 (no response found!)
88	32.768197262	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=209/53504, ttl=64
89	33.791989419	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=210/53760, ttl=64 (no response found!)
90	33.792172890	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=210/53760, ttl=64
91	34.049009584	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
92	34.815996755	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=211/54016, ttl=64 (no response found!)
93	34.816188957	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=211/54016, ttl=64
94	35.839990124	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=212/54272, ttl=64 (no response found!)
95	35.840188471	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=212/54272, ttl=64
96	36.051166600	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
97	36.863998577	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=213/54528, ttl=64 (no response found!)
98	36.864188405	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=213/54528, ttl=64
99	37.887989711	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=214/54784, ttl=64 (no response found!)
100	37.888172763	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=214/54784, ttl=64
101	38.053343940	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
102	38.911998863	172.16.50.1	172.16.50.255	ICMP	98 Echo (ping) request id=0x3cf5, seq=215/55040, ttl=64 (no response found!)
103	38.912184290	172.16.50.254	172.16.50.1	ICMP	98 Echo (ping) reply id=0x3cf5, seq=215/55040, ttl=64

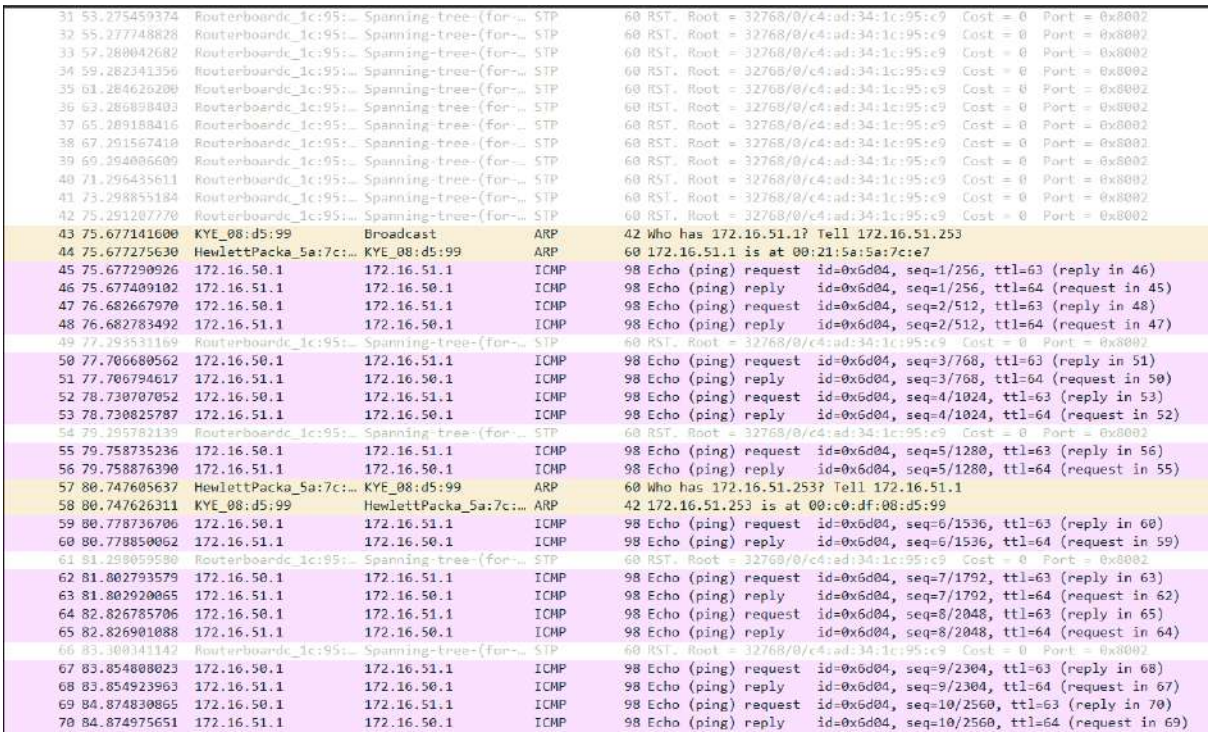
Tabela 13: Experiência 3 step 7 ping 51_1, exemplo de log do Wireshark

1	0.000000000	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
2	1.806256881	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=1/256, ttl=64 (no response found!)
3	2.001429974	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
4	2.833202628	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=2/512, ttl=64 (no response found!)
5	3.857192574	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=3/768, ttl=64 (no response found!)
6	4.003585315	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
7	4.881176723	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=4/1024, ttl=64 (no response found!)
8	5.905189298	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=5/1280, ttl=64 (no response found!)
9	6.005728712	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
10	6.929168349	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=6/1536, ttl=64 (no response found!)
11	6.993149125	HewlettPacka_61:2c:…	HewlettPacka_19:09:…	ARP	42 Who has 172.16.50.254? Tell 172.16.50.1
12	6.993267854	HewlettPacka_19:09:…	HewlettPacka_61:2c:…	ARP	60 172.16.50.254 is at 00:22:64:19:09:5c
13	7.953167164	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=7/1792, ttl=64 (no response found!)
14	8.007741018	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
15	8.977165911	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=8/2048, ttl=64 (no response found!)
16	10.001168637	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=9/2304, ttl=64 (no response found!)
17	10.009995113	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
18	11.025192317	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=10/2560, ttl=64 (no response found!)
19	12.012167635	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
20	12.049169203	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=11/2816, ttl=64 (no response found!)
21	13.073194139	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=12/3072, ttl=64 (no response found!)
22	14.014309845	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
23	14.097186879	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=13/3328, ttl=64 (no response found!)
24	15.121197009	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=14/3584, ttl=64 (no response found!)
25	16.016455477	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
26	16.145196244	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=15/3840, ttl=64 (no response found!)
27	17.173192047	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=16/4096, ttl=64 (no response found!)
28	18.018581135	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8001
29	18.193187799	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=17/4352, ttl=64 (no response found!)
30	19.217189618	172.16.50.1	172.16.51.253	ICMP	98 Echo (ping) request id=0x4324, seq=18/4608, ttl=64 (no response found!)

Tabela 14: Experiência 3 step 7 ping 172.16.51.253, exemplo de log do Wireshark

36	64.071494862	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
37	66.073768951	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
38	68.076055402	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
39	68.797976528	fe80::221:5aff:fe61…	ff02::fb	MDNS	180 Standard query 0x0000 PTR _nfs._tcp.local, "QM" question PTR _ftp._tcp.local,
40	70.068428941	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
41	72.070866324	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
42	74.073295047	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
43	76.075716156	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
44	77.462736953	HewlettPacka_61:2c:…	Broadcast	ARP	60 Who has 172.16.50.254? Tell 172.16.50.1
45	77.462764052	HewlettPacka_19:09:…	HewlettPacka_61:2c:…	ARP	42 172.16.50.254 is at 00:22:64:19:09:5c
46	77.462883485	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=1/256, ttl=64 (reply in 47)
47	77.463178714	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=1/256, ttl=63 (request in 46)
48	78.078067352	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
49	78.468400915	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=2/512, ttl=64 (reply in 50)
50	78.468561904	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=2/512, ttl=63 (request in 49)
51	79.492412878	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=3/768, ttl=64 (reply in 52)
52	79.492572121	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=3/768, ttl=63 (request in 51)
53	80.080391868	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
54	80.516439438	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=4/1024, ttl=64 (reply in 55)
55	80.516610206	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=4/1024, ttl=63 (request in 54)
56	81.544463850	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=5/1280, ttl=64 (reply in 57)
57	81.544653895	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=5/1280, ttl=63 (request in 56)
58	82.082645982	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
59	82.564468882	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=6/1536, ttl=64 (reply in 60)
60	82.564635180	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=6/1536, ttl=63 (request in 59)
61	82.660755433	HewlettPacka_19:09:…	HewlettPacka_61:2c:…	ARP	42 Who has 172.16.50.1? Tell 172.16.50.254
62	82.660891698	HewlettPacka_61:2c:…	HewlettPacka_19:09:…	ARP	60 172.16.50.1 is at 00:21:5a:61:2c:54
63	83.588515348	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=7/1792, ttl=64 (reply in 64)
64	83.588725787	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=7/1792, ttl=63 (request in 63)
65	84.084928312	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
66	84.612519279	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=8/2048, ttl=64 (reply in 67)
67	84.612678243	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=8/2048, ttl=63 (request in 66)
68	85.640539710	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=9/2304, ttl=64 (reply in 69)
69	85.640706776	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=9/2304, ttl=63 (request in 68)
70	86.087204146	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002
71	86.660563949	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=10/2560, ttl=64 (reply in 72)
72	86.660754133	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=10/2560, ttl=63 (request in 71)
73	87.684579893	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=11/2816, ttl=64 (reply in 74)
74	87.684734876	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=11/2816, ttl=63 (request in 73)
75	88.089488362	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c8 Cost = 0 Port = 0x8002

Tabela 15: Experiência 3 step 11 tux54 eth0, exemplo de log do Wireshark



31	53.275459374	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c9 Cost = 0 Port = 0x8002
32	55.277748828	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c9 Cost = 0 Port = 0x8002
33	57.280042682	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c9 Cost = 0 Port = 0x8002
34	59.282341356	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c9 Cost = 0 Port = 0x8002
35	61.284626200	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c9 Cost = 0 Port = 0x8002
36	63.286898403	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c9 Cost = 0 Port = 0x8002
37	65.289188416	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c9 Cost = 0 Port = 0x8002
38	67.291567410	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c9 Cost = 0 Port = 0x8002
39	69.294006609	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c9 Cost = 0 Port = 0x8002
40	71.296435611	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c9 Cost = 0 Port = 0x8002
41	73.298855184	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c9 Cost = 0 Port = 0x8002
42	75.291207770	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c9 Cost = 0 Port = 0x8002
43	75.677141600	KYE_08:d5:99	Broadcast	ARP	42 Who has 172.16.51.1? Tell 172.16.51.253
44	75.677275630	HewlettPacka_5a:7c:…	KYE_08:d5:99	ARP	60 172.16.51.1 is at 00:21:5a:5a:7c:e7
45	75.677290926	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=1/256, ttl=63 (reply in 46)
46	75.677409102	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=1/256, ttl=64 (request in 45)
47	76.682667970	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=2/512, ttl=63 (reply in 48)
48	76.682783492	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=2/512, ttl=64 (request in 47)
49	77.295531169	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c9 Cost = 0 Port = 0x8002
50	77.706680562	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=3/768, ttl=63 (reply in 51)
51	77.706794617	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=3/768, ttl=64 (request in 50)
52	78.730707052	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=4/1024, ttl=63 (reply in 53)
53	78.730825787	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=4/1024, ttl=64 (request in 52)
54	79.295782139	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c9 Cost = 0 Port = 0x8002
55	79.758735236	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=5/1280, ttl=63 (reply in 56)
56	79.758876390	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=5/1280, ttl=64 (request in 55)
57	80.747605637	HewlettPacka_5a:7c:…	KYE_08:d5:99	ARP	60 Who has 172.16.51.253? Tell 172.16.51.1
58	80.747626311	KYE_08:d5:99	HewlettPacka_5a:7c:…	ARP	42 172.16.51.253 is at 00:c0:df:08:d5:99
59	80.778736706	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=6/1536, ttl=63 (reply in 60)
60	80.778850062	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=6/1536, ttl=64 (request in 59)
61	81.298059580	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c9 Cost = 0 Port = 0x8002
62	81.802793579	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=7/1792, ttl=63 (reply in 63)
63	81.802920065	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=7/1792, ttl=64 (request in 62)
64	82.826785706	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=8/2048, ttl=63 (reply in 65)
65	82.826901088	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=8/2048, ttl=64 (request in 64)
66	83.300341142	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/c4:ad:34:1c:95:c9 Cost = 0 Port = 0x8002
67	83.854808023	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=9/2304, ttl=63 (reply in 68)
68	83.854923963	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=9/2304, ttl=64 (request in 67)
69	84.874830865	172.16.50.1	172.16.51.1	ICMP	98 Echo (ping) request id=0x6d04, seq=10/2560, ttl=63 (reply in 70)
70	84.874975651	172.16.51.1	172.16.50.1	ICMP	98 Echo (ping) reply id=0x6d04, seq=10/2560, ttl=64 (request in 69)

Tabela 16: Experiência 3 step 11 tux54 eth1, exemplo de log do Wireshark

```
1st
traceroute to 172.16.50.1 (172.16.50.1), 30 hops max, 60 byte packets
 1  172.16.51.254  0.190 ms  0.184 ms  0.195 ms
 2  172.16.51.253  0.328 ms  0.312 ms  0.295 ms
 3  172.16.50.1  0.522 ms  0.509 ms  0.493 ms

2nd
root@gnu52:~# traceroute -n 172.16.50.1
traceroute to 172.16.50.1 (172.16.50.1), 30 hops max, 60 byte packets
 1  172.16.51.253  0.178 ms  0.156 ms  0.142 ms
 2  172.16.50.1  0.403 ms  0.386 ms  0.369 ms
```

Figura 1: Experiência 4 step 4

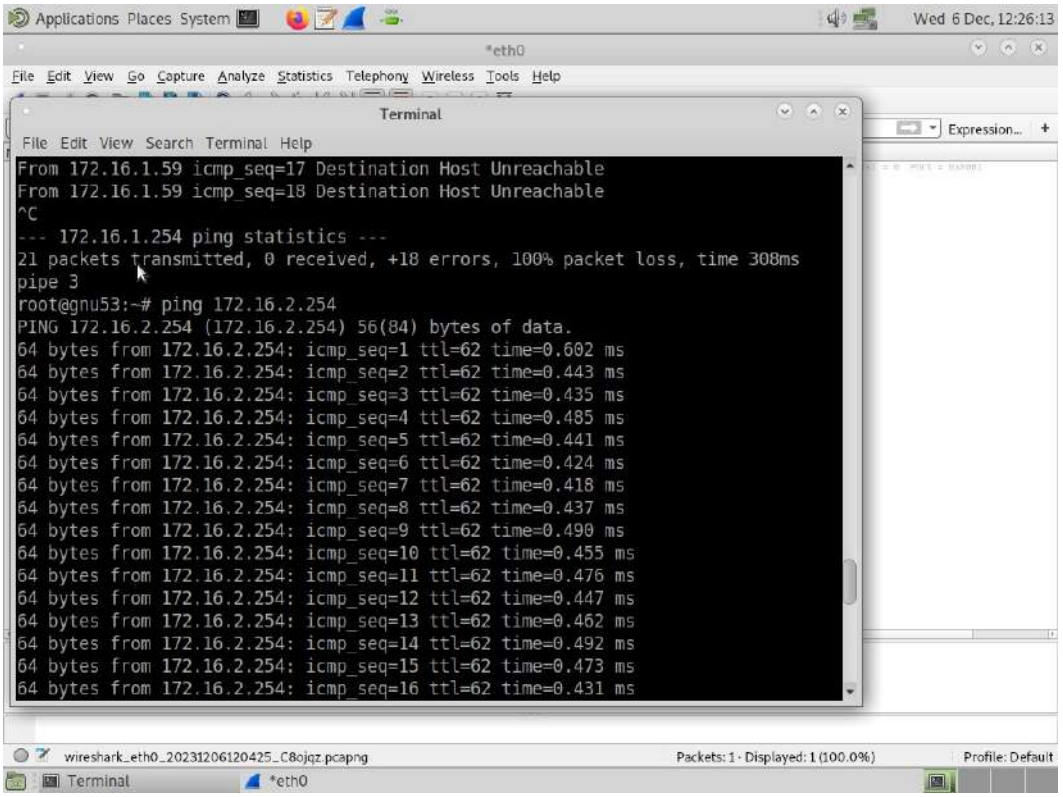


Figura 1: Experiência 4 step 5

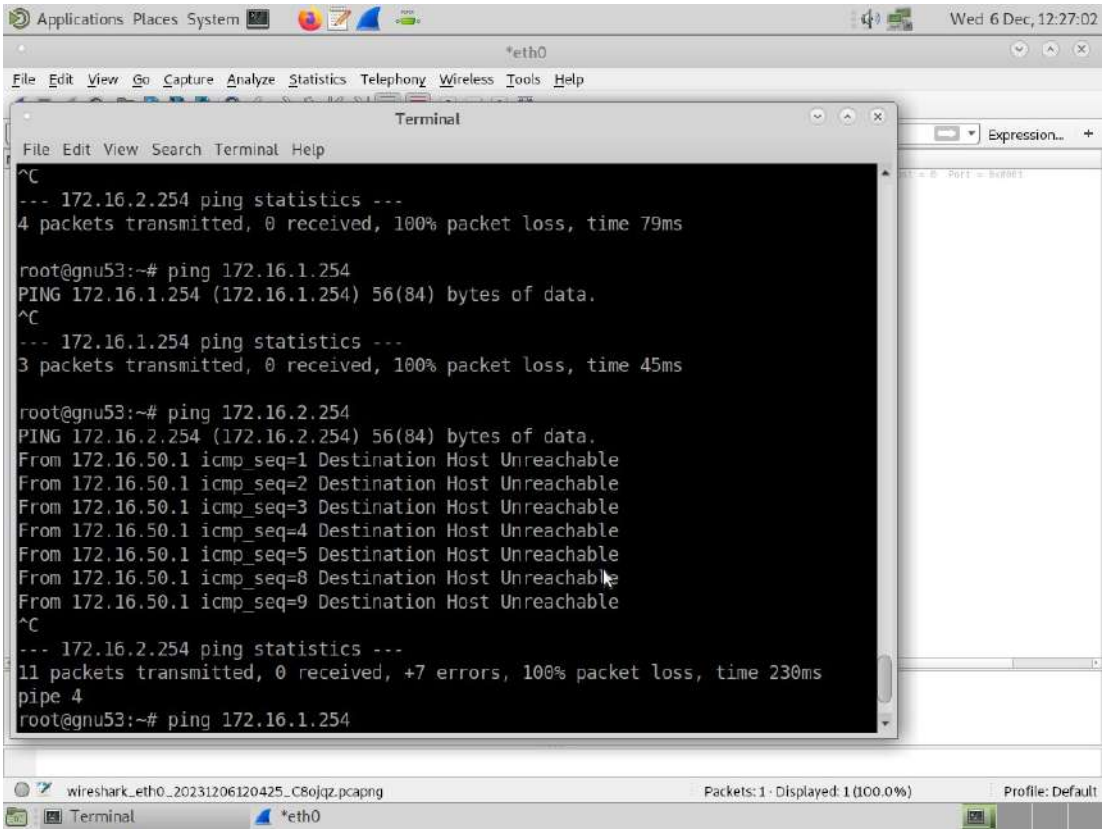


Figura 3: Experiência 4 step 7

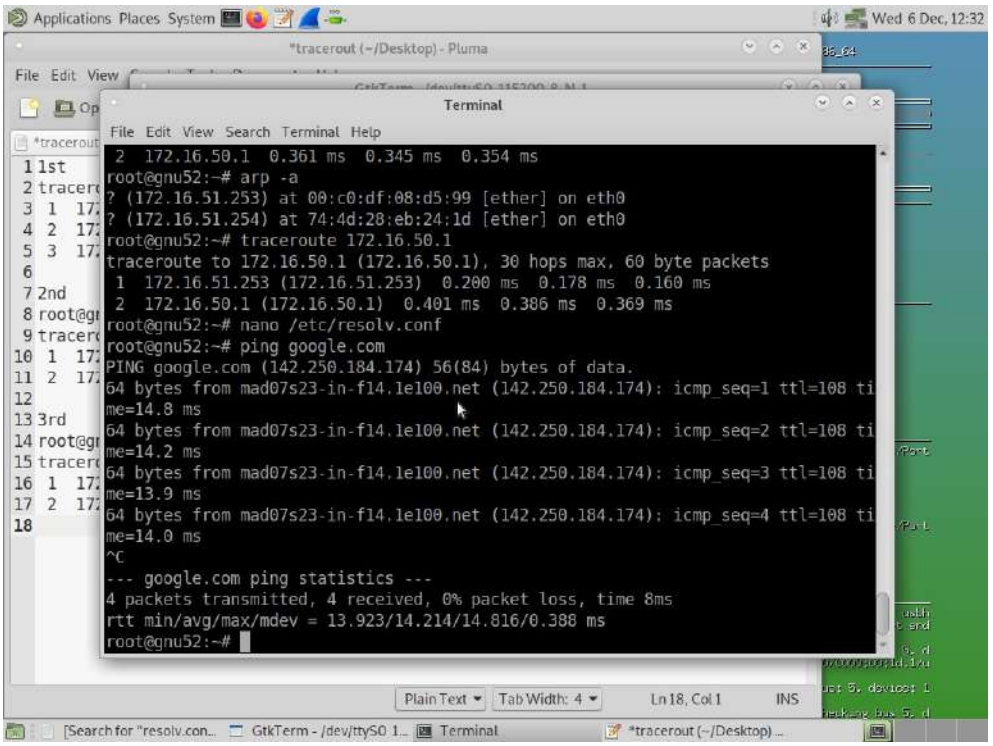


Figura 4: Experiência 5 step 3

1	0.000000000	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:1d Cost = 10 Port = 0x8001
2	0.014660881	172.16.51.1	172.16.2.1	DNS	70	Standard query 0xdb6e A google.com
3	0.014671497	172.16.51.1	172.16.2.1	DNS	70	Standard query 0x8777 AAAA google.com
4	0.015258175	172.16.2.1	172.16.51.1	DNS	86	Standard query response 0xdb6e A google.com A 142.250.184.174
5	0.015272074	172.16.2.1	172.16.51.1	DNS	98	Standard query response 0x8777 AAAA google.com AAAA 2a00:1450:4003:80c::200e
6	0.015599846	172.16.51.1	142.250.184.174	ICMP	98	Echo (ping) request id=0x4bc7, seq=1/256, ttl=64 (reply in 7)
7	0.030317369	142.250.184.174	172.16.51.1	ICMP	98	Echo (ping) reply id=0x4bc7, seq=1/256, ttl=108 (request in 6)
8	0.030482826	172.16.51.1	172.16.2.1	DNS	88	Standard query 0xabc0 PTR 174.184.250.142.in-addr.arpa
9	0.030915851	172.16.2.1	172.16.51.1	DNS	127	Standard query response 0xabc0 PTR 174.184.250.142.in-addr.arpa PTR mad07s23-in-f14.1e100.net
10	1.017018083	172.16.51.1	142.250.184.174	ICMP	98	Echo (ping) request id=0x4bc7, seq=2/512, ttl=64 (reply in 11)
11	1.031254531	142.250.184.174	172.16.51.1	ICMP	98	Echo (ping) reply id=0x4bc7, seq=2/512, ttl=108 (request in 10)
12	2.002117798	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:1d Cost = 10 Port = 0x8001
13	2.017581800	172.16.51.1	142.250.184.174	ICMP	98	Echo (ping) request id=0x4bc7, seq=3/768, ttl=64 (reply in 14)
14	2.031255735	142.250.184.174	172.16.51.1	ICMP	98	Echo (ping) reply id=0x4bc7, seq=3/768, ttl=108 (request in 13)
15	3.019339203	172.16.51.1	142.250.184.174	ICMP	98	Echo (ping) request id=0x4bc7, seq=4/1024, ttl=64 (reply in 16)
16	3.033261499	142.250.184.174	172.16.51.1	ICMP	98	Echo (ping) reply id=0x4bc7, seq=4/1024, ttl=108 (request in 15)
17	4.004255940	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:1d Cost = 10 Port = 0x8001
18	4.021343505	172.16.51.1	142.250.184.174	ICMP	98	Echo (ping) request id=0x4bc7, seq=5/1280, ttl=64 (reply in 19)
19	4.035252251	142.250.184.174	172.16.51.1	ICMP	98	Echo (ping) reply id=0x4bc7, seq=5/1280, ttl=108 (request in 18)
20	5.012208605	Routerboardc_eb:24:…	HewlettPacka_5a:7c:…	ARP	60	Who has 172.16.51.1? Tell 172.16.51.254
21	5.012228790	HewlettPacka_5a:7c:…	Routerboardc_eb:24:…	ARP	42	172.16.51.1 is at 00:21:5a:5a:7c:e7
22	5.023323017	172.16.51.1	142.250.184.174	ICMP	98	Echo (ping) request id=0x4bc7, seq=6/1536, ttl=64 (reply in 23)
23	5.037239794	142.250.184.174	172.16.51.1	ICMP	98	Echo (ping) reply id=0x4bc7, seq=6/1536, ttl=108 (request in 22)
24	5.239547640	HewlettPacka_5a:7c:…	Routerboardc_eb:24:…	ARP	42	Who has 172.16.51.254? Tell 172.16.51.1
25	5.239637039	Routerboardc_eb:24:…	HewlettPacka_5a:7c:…	ARP	60	172.16.51.254 is at 74:4d:28:eb:24:1d
26	6.006393332	Routerboardc_1c:95:…	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:1d Cost = 10 Port = 0x8001
27	6.025317829	172.16.51.1	142.250.184.174	ICMP	98	Echo (ping) request id=0x4bc7, seq=7/1792, ttl=64 (reply in 28)
28	6.039252626	142.250.184.174	172.16.51.1	ICMP	98	Echo (ping) reply id=0x4bc7, seq=7/1792, ttl=108 (request in 27)

Tabela 17: Experiência 5 step 3, exemplo de log do Wireshark

6	4.475250489	172.16.50.1	172.16.1.1	DNS	69	Standard query 0xc1f2 A ftp.up.pt
7	4.476093410	172.16.1.1	172.16.50.1	DNS	197	Standard query response 0xc1f2 A ftp.up.pt CNAME mirrors.up.pt A 193.137.29.15
8	4.476250421	172.16.50.1	193.137.29.15	TCP	74	37074 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=778752941 TSecr=0 WS=128
9	4.479865261	193.137.29.15	172.16.50.1	TCP	74	21 → 37074 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1380 SACK_PERM=1 TSval=1740801032 TSecr=778752941 WS=128
10	4.479902442	172.16.50.1	193.137.29.15	TCP	66	37074 → 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=778752945 TSecr=1740801032
11	4.485723128	193.137.29.15	172.16.50.1	FTP	139	Response: 220-Welcome to the University of Porto's mirror archive (mirrors.up.pt)
12	4.485733884	172.16.50.1	193.137.29.15	TCP	66	37074 → 21 [ACK] Seq=1 Ack=74 Win=64256 Len=0 TSval=778752951 TSecr=1740801036
13	4.485761332	193.137.29.15	172.16.50.1	FTP	141	Response: 220----------
14	4.485767827	172.16.50.1	193.137.29.15	TCP	66	37074 → 21 [ACK] Seq=1 Ack=149 Win=64256 Len=0 TSval=778752951 TSecr=1740801036
15	4.485831034	193.137.29.15	172.16.50.1	FTP	298	Response: 220-All connections and transfers are logged. The max number of connections is 200.
16	4.485836412	172.16.50.1	193.137.29.15	TCP	66	37074 → 21 [ACK] Seq=1 Ack=381 Win=64128 Len=0 TSval=778752951 TSecr=1740801036
17	4.485839275	193.137.29.15	172.16.50.1	FTP	78	Response: 220-
18	4.485843186	172.16.50.1	193.137.29.15	TCP	66	37074 → 21 [ACK] Seq=1 Ack=393 Win=64128 Len=0 TSval=778752951 TSecr=1740801037
19	4.486243031	172.16.50.1	193.137.29.15	FTP	81	Request: user anonymous
20	4.488753704	193.137.29.15	172.16.50.1	TCP	66	21 → 37074 [ACK] Seq=393 Ack=16 Win=65280 Len=0 TSval=1740801041 TSecr=778752951
21	4.488799660	193.137.29.15	172.16.50.1	FTP	100	Response: 331 Please specify the password.
22	4.488853927	172.16.50.1	193.137.29.15	FTP	80	Request: pass password
23	4.492568114	193.137.29.15	172.16.50.1	FTP	89	Response: 230 Login successful.
24	4.492611967	172.16.50.1	193.137.29.15	FTP	71	Request: pasv
25	4.495069638	193.137.29.15	172.16.50.1	FTP	118	Response: 227 Entering Passive Mode (193,137,29,15,220,147).
26	4.495165601	172.16.50.1	193.137.29.15	TCP	74	42642 → 56771 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=778752960 TSecr=0 WS=128
27	4.499775439	193.137.29.15	172.16.50.1	TCP	74	56771 → 42642 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1380 SACK_PERM=1 TSval=1740801051 TSecr=778752960 WS=1
28	4.499787941	172.16.50.1	193.137.29.15	TCP	66	42642 → 56771 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=778752965 TSecr=1740801051
29	4.499802887	172.16.50.1	193.137.29.15	FTP	143	Request: retr pub/kodi/test-builds/darwin/tvos/kodi-20201003-09847870-master-tvos.deb
30	4.501992288	193.137.29.15	172.16.50.1	FTP	202	Response: 150 Opening BINARY mode data connection for pub/kodi/test-builds/darwin/tvos/kodi-20201003-09847870-m

Tabela 18: Experiência 6 step 4, exemplo de log do Wireshark

5.2. Código fonte da aplicação Download

5.2.1 - parser.h

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

#define MAX_LENGTH 256
#define STATE_FTP 0
#define STATE_USER 1
#define STATE_PASSWORD 2
#define STATE_HOST 3
#define STATE_PATH 4
#define SERVICE_READY 220
#define USER_NAME_OK 331
#define LOGGED_IN 230
#define ENTER_PASSIVE 227
#define FILE_OK 150
#define DOWNLOAD 226

typedef struct URL{
    char user[MAX_LENGTH];
    char password[MAX_LENGTH];
    char path[MAX_LENGTH];
    char host[MAX_LENGTH];
    char filename[MAX_LENGTH];
    int port;
} url;

void defaultParser(url *url);
void setDefaultUserInfo(url *url);
void parse(url *url, const char *input);
```

5.2.2 - parser.c

```c
#include "../include/parser.h"
#include "string.h"

void defaultParser(url* url) {
    memset(url->user, 0, MAX_LENGTH);
    memset(url->password, 0, MAX_LENGTH);
    memset(url->host, 0, MAX_LENGTH);
    memset(url->path, 0, MAX_LENGTH);
    memset(url->filename, 0, MAX_LENGTH);
     url->port = 21; // default de ftp control port -> Permite a conexão
com o server uma vez que abre automaticamente
}

void setDefaultUserInfo(url *url) {
    memcpy(url->host, url->user, MAX_LENGTH);
    memset(url->user, 0, MAX_LENGTH);
    memcpy(url->user, "anonymous", 9);
    memcpy(url->password, "anonymous", 9);
}

void extract_filename(url *url) {
    int it = 0;
    for (int i = strlen(url->path) - 1; i >= 0 ; i--) {
        if (url->path[i] == '/') {
            break;
        }
        url->filename[it++] = url->path[i];
    }
    url->filename[it] = '\0'; // Null terminate the string

    // Reverse the filename
    for (int start = 0, end = it - 1; start < end; start++, end--) {
        char temp = url->filename[start];
        url->filename[start] = url->filename[end];
        url->filename[end] = temp;
    }
}
```

```
void parse(url *url, const char *input) {
    printf("Parsing...\n");

    int state = STATE_FTP;
    int infIndex = 0;
    int inputSize = strlen(input);
    char urlStart[] = "ftp://";

    for (int i = 0; i < inputSize; i++) {
        switch (state) {
            case STATE_FTP:
            if (strncmp(input, "ftp://", 6) != 0) {
                printf("Error: URL does not start with ftp://\n");
                return;
            }
            state = STATE_USER;
            i += 5; // Skip the "ftp://" part
            break;
            case STATE_USER:
                if (input[i] == ':') {
                    state = STATE_PASSWORD;
                    infIndex = 0;
                      } else if (input[i] == '/') { // No Username or
Password

                    setDefaultUserInfo(url);
                    state = STATE_PATH; // Skip to path
                    infIndex = 0;
                } else {
                    url->user[infIndex] = input[i];
                    infIndex++;
                }
                break;
            case STATE_PASSWORD:
                if (input[i] == '@') {
                    state = STATE_HOST;
                    infIndex = 0;
                } else {
                    url->password[infIndex] = input[i];
                    infIndex++;
                }
```

```
                break;
            case STATE_HOST:
                if (input[i] == '/') {
                    state = STATE_PATH;
                    infIndex = 0;
                } else {
                    url->host[infIndex] = input[i];
                    infIndex++;
                }
                break;
            case STATE_PATH:
                url->path[infIndex] = input[i];
                infIndex++;
                break;
            default:
                break;
        }
    }

    if (state == STATE_USER || state == STATE_PASSWORD) {
        printf("Error: Incomplete URL\n");
        return;
    }

    url->path[infIndex] = '\0'; // Null terminate the path

    extract_filename(url);
}
```

5.2.3 - getip.c

```
#include <stdio.h>
#include <stdlib.h>
#include <netdb.h>
#include <netinet/in.h>
#include<arpa/inet.h>

int getIP(char *hostname, char* IPaddress){
    struct hostent *h;

    if ((h = gethostbyname(hostname)) == NULL) {
        herror("gethostbyname()");
        exit(-1);
    }

    /* Retrieve IP*/
    strcpy(IPaddress, inet_ntoa(*((struct in_addr *) h->h_addr)));

    printf("Hostname  : %s\n", h->h_name);
        printf("IP  Address  :  %s\n",  inet_ntoa(*((struct  in_addr  *)
h->h_addr)));

    return 0;
}
```

5.2.4 - clientTCP.c

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

int createSocket(char *serverAddress,int port){ //sv_add = ip macro
    int sockfd;
    struct sockaddr_in server_addr;
```

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

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

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

    return sockfd;
}

// Ex: If the input string is something like "200\n404\n500\n", the
function will return 500.
int getLastLineStatusCode(const char *buf) {
    int a = 0;
    const char *start = buf;

    while (*buf != '\0') {
        if (*buf == '\n') {
            a = atoi(start);
            start = buf + 1;
        }
        buf++;
    }

    if (start != buf) {
        a = atoi(start);
```

```
    }

    return a;
}

// "192,168,1,1,123,34", the function will return 123*256 + 34 = 31490.
int getPortNumber(char *buf) {
    int num[5] = {0};
    int i = 0;
    char *pt = strtok(buf, ",");
    pt = strtok(NULL, ",");

    while (pt != NULL) {
        int a = atoi(pt);
        num[i] = a;
        pt = strtok(NULL, ",");
        i++;
    }
    return (num[3] * 256 + num[4]);
}

int downloadFileFromDataConnection(int dataSocket, FILE *fileptr) {
    char buffer[500];
    size_t bytesRead;

     while ((bytesRead = read(dataSocket, buffer, sizeof(buffer))) > 0)
{
        fwrite(buffer, 1, bytesRead, fileptr);
    }

    return bytesRead;
}

int handleEnterPassive(int sockfd, char* buf, char* IPAddress, char*
retrvPath) {
    int port = getPortNumber(buf);
    int sockfd2 = createSocket(IPAddress, port);

    if (sockfd2 == -1) {
        fprintf(stderr, "Error creating data connection\n");
```

```
        return -1;
    }

    write(sockfd, retrvPath, strlen(retrvPath));
    printf("Data connection created\n");
    return sockfd2;
}

void handleDownload(int sockfd2, FILE* fileptr) {
    char bufDownload[1024];
    ssize_t bytesDownload;
    while (1) {
        memset(bufDownload, 0, sizeof(bufDownload));
                        bytesDownload   =   read(sockfd2,   bufDownload,
sizeof(bufDownload));
        if (bytesDownload > 0) {
            fwrite(bufDownload, sizeof(char), bytesDownload, fileptr);
            printf("%.*s", (int)bytesDownload, bufDownload);
        } else {
            break;
        }
    }
}

int connectionDownload(url *url, char *IPAddress) {
    int sockfd = createSocket(IPAddress, url->port);
    if (sockfd == -1) {
        fprintf(stderr, "Error creating socket\n");
        return -1;
    }

    char userLogin[263];
    snprintf(userLogin, sizeof(userLogin), "user %s\r\n", url->user);

    char passwdLogin[263];
          snprintf(passwdLogin,   sizeof(passwdLogin),   "pass   %s\r\n",
url->password);

    char retrvPath[263];
    snprintf(retrvPath, sizeof(retrvPath), "retr %s\r\n", url->path);
```

```
    int sockfd2 = 0;
    FILE *fileptr = NULL;
    int port = 0;
    int STOP = 0;
    int visited = 0;
    size_t bytes, bytesDownload;
    char buf[500] = {0};
    char bufDownload[500] = {0};

    while (!STOP) {
        memset(buf, 0, sizeof(buf));
        if (bytes = read(sockfd, buf, sizeof(buf)) <= 0) continue;

        printf("\n%s\n", buf);
        int statusCode = getLastLineStatusCode(buf);

        switch (statusCode) {
            case SERVICE_READY:
                if (!visited) {
                    visited = 1;
                    write(sockfd, userLogin, strlen(userLogin));
                    printf("User login sent\n");
                }
                break;
            case USER_NAME_OK:
                write(sockfd, passwdLogin, strlen(passwdLogin));
                printf("Password sent\n");
                break;
            case LOGGED_IN:
                write(sockfd, "pasv\r\n", 6);
                printf("Entering passive mode\n");
                break;
            case ENTER_PASSIVE:
                   sockfd2 = handleEnterPassive(sockfd, buf, IPAddress,
retrvPath);
                if (sockfd2 == -1) {
                    return -1;
                }
                break;
```

```
            case FILE_OK:
                fileptr = fopen(url->filename, "w");
                printf("File opened and ready for download\n");
                break;
            case DOWNLOAD:
                handleDownload(sockfd2, fileptr);
                STOP = 1;
                printf("\nDownload completed\n");
                break;
            default:
                    fprintf(stderr, "Received unexpected status code:
%d\n", statusCode);
                return -1;
        }
    }


    if (fclose(fileptr) < 0) {
        return -1;
    }

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

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

    return 0;
}
```

5.2.5 - download.c

```c
#include "src/parser.c"
#include "src/clientTCP.c"
#include "src/getip.c"

int main(int argc, char *argv[]) {
    if (argc != 2) {
            printf("ERROR MAIN START // URL should be as follows:
ftp://[<user>:<password>@]<host>/<url-path>\n");
        exit(-1);
    }
    url inputURL;
    defaultParser(&inputURL);
    parse(&inputURL, argv[1]);

    printf("Username: %s\n", inputURL.user);
    printf("Password: %s\n", inputURL.password);
    printf("Host: %s\n", inputURL.host);
    printf("Path: %s\n", inputURL.path);
    printf("Filename: %s\n", inputURL.filename);

    char IPAddress[20] = "";

    if (getIP(inputURL.host, IPAddress) != 0) {
        printf("getIP error on main\n");
        exit(-1);
    }

    if (connectionDownload(&inputURL, IPAddress) != 0) {
        printf("connectionDownload error on main \n");
        exit(-1);
    }

    return 0;
}
```