

Universidade do Minho
Escola de Engenharia

Redes de Computadores I
ENGENHARIA DE TELECOMUNICAÇÕES E INFORMÁTICA
2020/2021

(Docente: Maria João Mesquita Rodrigues Cunha Nicolau Pinto)

17 de janeiro de 2021

**Simulação de LANs Ethernet e redes
TCP/IP usando o CORE**

Marco António Cerqueira Araújo - a89387@alunos.uminho.pt

Rui Filipe Ribeiro Freitas - a84121@alunos.uminho.pt

Tiago João Pereira Ferreira - a85392@alunos.uminho.pt

Índice

Índice de figuras	3
Índice de tabelas.....	3
Introdução	4
Desenvolvimento.....	5
1. Emulação de LANs Ethernet	5
1.1. Elaboração da topologia da rede	5
1.2. Testes de conectividade	6
1.3. Capturas de tráfego no Wireshark	7
2. Interligação de LANs e redes IP	8
2.1. Elaboração da topologia da rede	8
2.2. Atribuição dos endereços IP.....	8
2.3. Tabelas de encaminhamento	10
2.4. Encaminhamento estático	12
3. DHCP – <i>Dynamic Host Configuration Protocol</i>	13
3.1. Configuração do Protocolo DHCP	13
3.2. Testes DHCP	14
4. Uso das camadas de rede e transporte pelas aplicações.....	15
4.1. Ativação do servidor HTTP num PC da rede C	15
4.2. Testes	16
4.3. Formulário HTTP.....	17
Conclusão.....	18

Índice de figuras

Figura 1 - Topologia da rede local da fase 1.	5
Figura 2 - PING A ₁ -> C ₁	6
Figura 3 - PING D ₂ -> B ₁	6
Figura 4 - Captura de tráfego PC A2.	7
Figura 5 - Captura de tráfego PC B2.	7
Figura 6 - Topologia da rede local da fase 2.	8
Figura 7 - Ligações entre os routers.	8
Figura 8 - Esquema para o endereçamento das ligações entre os routers.	9
Figura 9 - Configuração zebra Router B.	12
Figura 10 - Configuração zebra Router A.	12
Figura 11 - Teste ping entre B2i e E1.	12
Figura 12 - Teste traceroute entre B2i e E1.	12
Figura 13 - Nova topologia da rede B.	13
Figura 14 - Configuração DHCP server.	13
Figura 15 - Configuração DHCP Client.	13
Figura 16 - Wireshark DHCP.	14
Figura 17 - Comando ifconfig.	14
Figura 18 - Configuração servidor HTTP.	15
Figura 19 - Servidor HTTP.	15
Figura 20 - Wireshark HTTP.	16
Figura 21 - Wireshark Protocolo TCP/IP.	16
Figura 22 - Formulário HTTP.	17
Figura 23 - Teste do formulário.	17

Índice de tabelas

Tabela 1 - Endereçamento das redes.	9
Tabela 2 - Endereçamento das ligações entre os routers.	9
Tabela 3 - Endereçamento de cada ponto da ligação.	10
Tabela 4 - Encaminhamento Router A.	10
Tabela 5 - Encaminhamento Router B.	10
Tabela 6 - Encaminhamento Router C.	11
Tabela 7 - Encaminhamento Router D.	11
Tabela 8 - Encaminhamento Router E.	11
Tabela 9 - Encaminhamento Router R1.	11
Tabela 10 - Encaminhamento Router R2.	11

Introdução

No âmbito da Unidade Curricular de Redes de Computadores I foi-nos proposto o desenvolvimento de um trabalho com o objetivo de promover a aquisição de competências na área de redes de computadores através de um projeto com base nas tecnologias Ethernet e TCP/IP.

Para a realização deste projeto foram necessários conhecimentos adquiridos nas aulas teóricas e teórico-práticas permitindo estes o desenvolvimento de uma rede de computadores. Para isso foram utilizadas 2 ferramentas, o CORE, um emulador de redes informáticas e o Wireshark, uma ferramenta importante para o diagnóstico do tráfego de rede.

Este trabalho está dividido em 4 fases em que a primeira passa pela construção de uma topologia de uma rede local e pela realização de testes de *ping* e *traceroute* entre esta. Na segunda fase adicionamos routers ao nosso trabalho, criando uma topologia de 5 redes locais interligadas por routers. Para isto tivemos de endereçar tudo manualmente de modo a obter eficácia em todas as conexões. Na fase 3 tivemos de implementar o protocolo DHCP numa das redes para que fosse possível a configuração dos elementos de uma rede de forma automática e dinâmica. Por último, na fase 4 instalamos na rede C o serviço HTTP de modo a que os vários *hosts* das várias redes pudessem utilizar.

De modo a sermos capazes de cumprir com os objetivos pretendidos tivemos de colocar em prática tudo o que aprendemos na unidade curricular de Redes de Computadores I.

Desenvolvimento

1. Emulação de LANs Ethernet

1.1. Elaboração da topologia da rede

Nesta primeira etapa da fase 2 é necessário a elaboração no CORE de uma pequena rede local que permita realizar o diagnóstico de conectividade, capturas e análise de tráfego. Na figura 1 encontra-se a topologia da rede local já realizada no CORE.

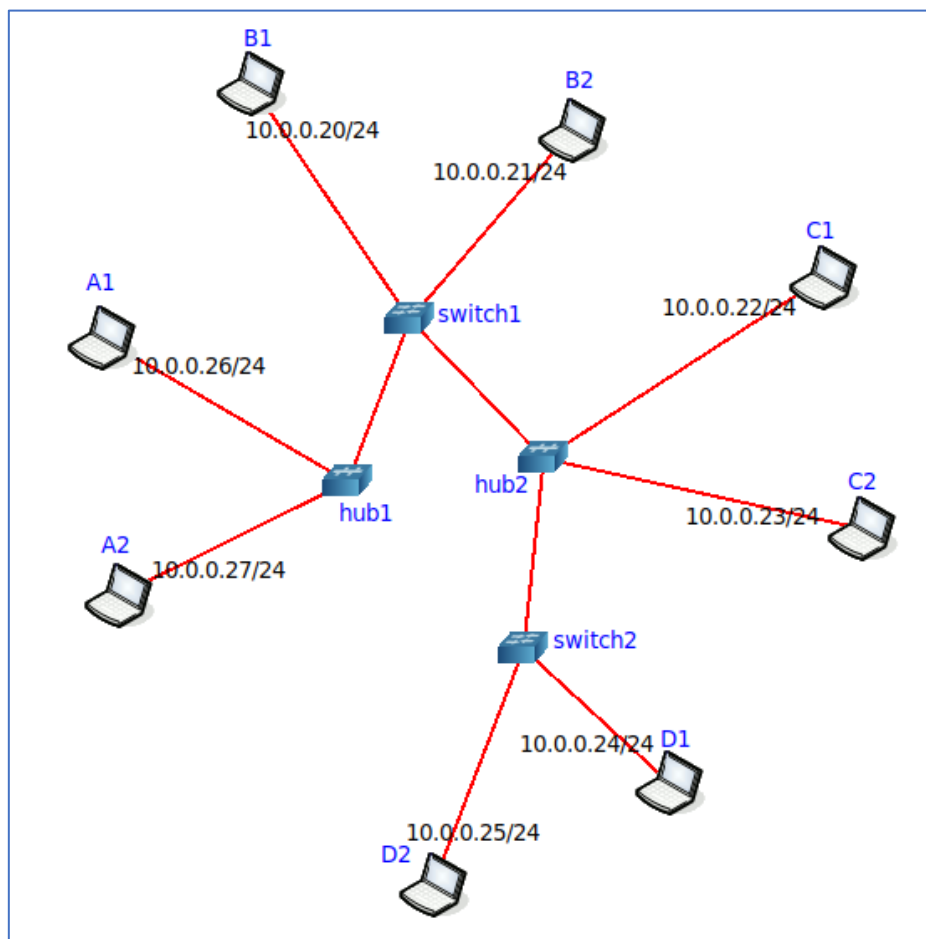


Figura 1 - Topologia da rede local da fase 1.

1.2. Testes de conectividade

Para testar a conectividade entre os vários sistemas terminais foi utilizado o comando *ping* como mostra a figura 2 e 3 onde testamos a conectividade entre o PC A₁ e C₁ e entre o D₂ e B₁ respetivamente.

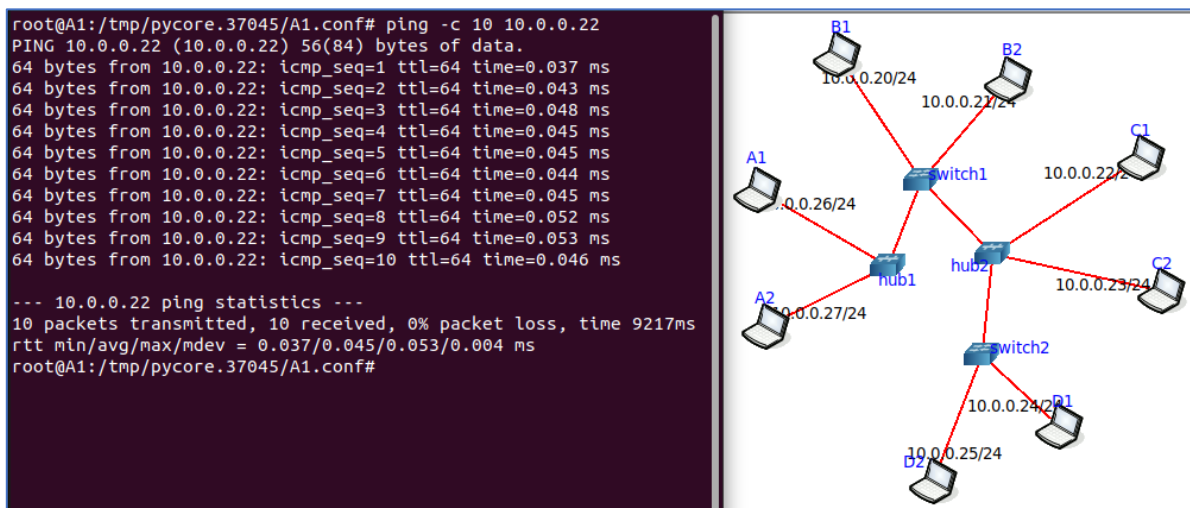


Figura 2 - PING A₁ -> C₁.

O comando ping utilizado pelo nosso grupo consiste no envio de 10 tramas para o PC destino esperando pelo eco das mesmas, sendo a conexão positiva se recebe as tramas e negativa caso contrário. Em ambas as situações reparamos que todas as tramas chegaram bem ao destinatário com um atraso de aproximadamente 0.045 milissegundos.

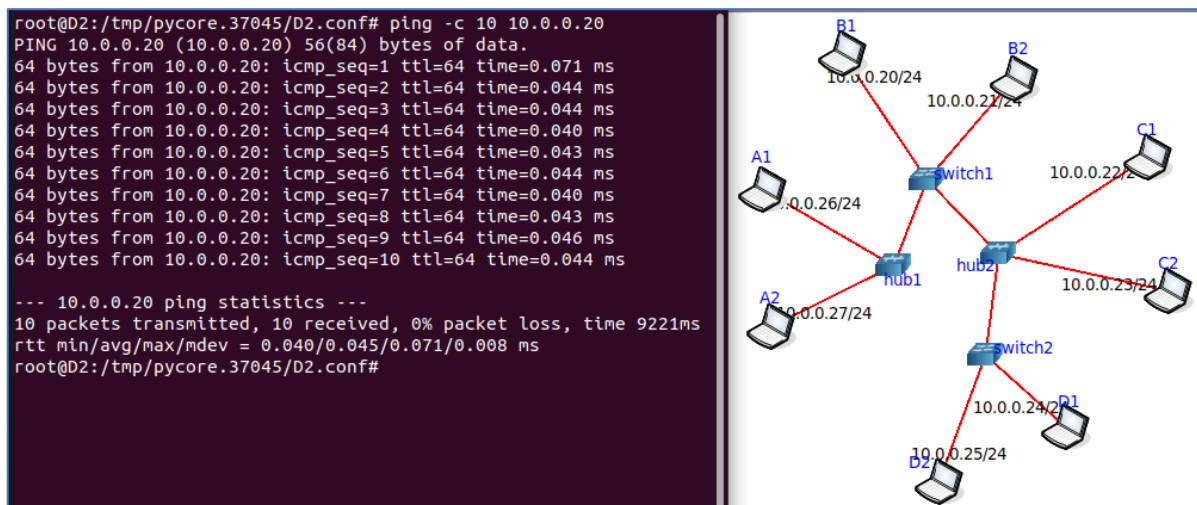


Figura 3 - PING D₂ -> B₁.

1.3. Capturas de tráfego no Wireshark

Para a captura de tráfego utilizámos o PC A₂ e o PC B₂ com o auxílio do comando ping que envia tramas do PC origem(A₂) para o PC destino(B₂). Na figura 4 e 5 estão presentes os resultados obtidos no Wireshark de ambos os PC's. Esta visualização no Wireshark também permitiu observar as diferenças entre um *Hub* e um *Switch*. No caso do *Hub* envia para todos os sistemas terminais a trama que recebeu enquanto que no *Switch* é enviada uma trama em *broadcast* para todos os sistemas terminais e após receber o endereço MAC do endereço IP respetivo é enviada apenas para esse as tramas respetivas. Podemos observar isto através do protocolo ARP que consiste no envio de uma chamada que procura endereços IP através do seu endereço MAC.

No.	Time	Source	Destination	Protocol	Length	Info
29	17.407726993	fe80::200:ff:feaa:1	ff02::2	ICMPv6	70	Router Solicitation from 00:00:00:aa:00:01
30	19.247580274	00:00:00_aa:00:07	Broadcast	ARP	42	Who has 10.0.0.21? Tell 10.0.0.27
31	19.247626666	00:00:00_aa:00:01	00:00:00_aa:00:07	ARP	42	10.0.0.21 is at 00:00:00:aa:00:01
32	19.247632931	10.0.0.27	10.0.0.21	ICMP	138	Echo (ping) request id=0x0011, seq=1/256, ttl...
33	19.247644135	10.0.0.21	10.0.0.27	ICMP	138	Echo (ping) reply id=0x0011, seq=1/256, ttl...
34	19.376521465	fe80::901f:faff:feb...	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "Q...
35	19.455737063	fe80::200:ff:feaa:5	ff02::2	ICMPv6	70	Router Solicitation from 00:00:00:aa:00:05
36	19.652516891	fe80::6f:1cff:fed:...	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "Q...
37	20.033572810	fe80::e48a:ff:fecf:...	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "Q...
38	20.064780861	fe80::5c8d:22ff:fe6...	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "Q...
39	20.224148617	fe80::e48a:ff:fecf:...	ff02::2	ICMPv6	70	Router Solicitation from e6:8a:00:cf:ea:06
40	20.224165334	fe80::78e2:4cff:fed...	ff02::2	ICMPv6	70	Router Solicitation from 7a:e2:4c:d0:82:00
41	20.224179258	fe80::5c8d:22ff:fe6...	ff02::2	ICMPv6	70	Router Solicitation from 5a:a6:11:7d:ac:68
42	20.224182680	fe80::200:ff:feaa:3	ff02::2	ICMPv6	70	Router Solicitation from 00:00:00:aa:00:03
43	20.224184797	fe80::4041:45ff:fe6...	ff02::2	ICMPv6	70	Router Solicitation from 42:41:45:6a:5c:d9
44	20.255838746	10.0.0.27	10.0.0.21	ICMP	138	Echo (ping) request id=0x0011 seq=2/512 ttl...
▶ Frame 30: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth6.0.f6, id 0						
▶ Ethernet II, Src: 00:00:00_aa:00:07 (00:00:00:aa:00:07), Dst: Broadcast (ff:ff:ff:ff:ff:ff)						
▼ Address Resolution Protocol (request)						
Hardware type: Ethernet (1)						
Protocol type: IPv4 (0x0800)						
Hardware size: 6						
Protocol size: 4						
Opcode: request (1)						
Sender MAC address: 00:00:00_aa:00:07 (00:00:00:aa:00:07)						
Sender IP address: 10.0.0.27						
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)						
Target IP address: 10.0.0.21						

Figura 4 - Captura de tráfego PC A2.

No.	Time	Source	Destination	Protocol	Length	Info
27	9.119832412	fe80::ccd6:d6ff:fe4...	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "Q...
28	9.281394714	fe80::8fd:8ff:fef6:...	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "Q...
29	9.438737624	fe80::dc77:86ff:fe8...	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "Q...
30	12.021752636	00:00:00_aa:00:07	Broadcast	ARP	42	Who has 10.0.0.21? Tell 10.0.0.27
31	12.021766309	00:00:00_aa:00:01	00:00:00_aa:00:07	ARP	42	10.0.0.21 is at 00:00:00:aa:00:01
32	12.021787185	10.0.0.27	10.0.0.21	ICMP	138	Echo (ping) request id=0x0011, seq=1/256, ttl...
33	12.021793212	10.0.0.21	10.0.0.27	ICMP	138	Echo (ping) reply id=0x0011, seq=1/256, ttl...
34	13.021986574	10.0.0.27	10.0.0.21	ICMP	138	Echo (ping) request id=0x0011, seq=2/512, ttl...
35	13.021999982	10.0.0.21	10.0.0.27	ICMP	138	Echo (ping) reply id=0x0011, seq=2/512, ttl...
36	14.046062668	10.0.0.27	10.0.0.21	ICMP	138	Echo (ping) request id=0x0011, seq=3/768, ttl...
37	14.046074937	10.0.0.21	10.0.0.27	ICMP	138	Echo (ping) reply id=0x0011, seq=3/768, ttl...
38	15.070678069	10.0.0.27	10.0.0.21	ICMP	138	Echo (ping) request id=0x0011, seq=4/1024, tt...
39	15.070689200	10.0.0.21	10.0.0.27	ICMP	138	Echo (ping) reply id=0x0011, seq=4/1024, tt...
40	16.094077890	10.0.0.27	10.0.0.21	ICMP	138	Echo (ping) request id=0x0011, seq=5/1280, tt...
41	16.094090033	10.0.0.21	10.0.0.27	ICMP	138	Echo (ping) reply id=0x0011, seq=5/1280, tt...
42	17.117962666	10.0.0.27	10.0.0.21	ICMP	138	Echo (ping) request id=0x0011 seq=6/1536 tt...
▶ Frame 30: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth8.0.f6, id 0						
▶ Ethernet II, Src: 00:00:00_aa:00:07 (00:00:00:aa:00:07), Dst: Broadcast (ff:ff:ff:ff:ff:ff)						
▼ Address Resolution Protocol (request)						
Hardware type: Ethernet (1)						
Protocol type: IPv4 (0x0800)						
Hardware size: 6						
Protocol size: 4						
Opcode: request (1)						
Sender MAC address: 00:00:00_aa:00:07 (00:00:00:aa:00:07)						
Sender IP address: 10.0.0.27						
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)						
Target IP address: 10.0.0.21						

Figura 5 - Captura de tráfego PC B2.

2. Interligação de LANs e redes IP

2.1. Elaboração da topologia da rede

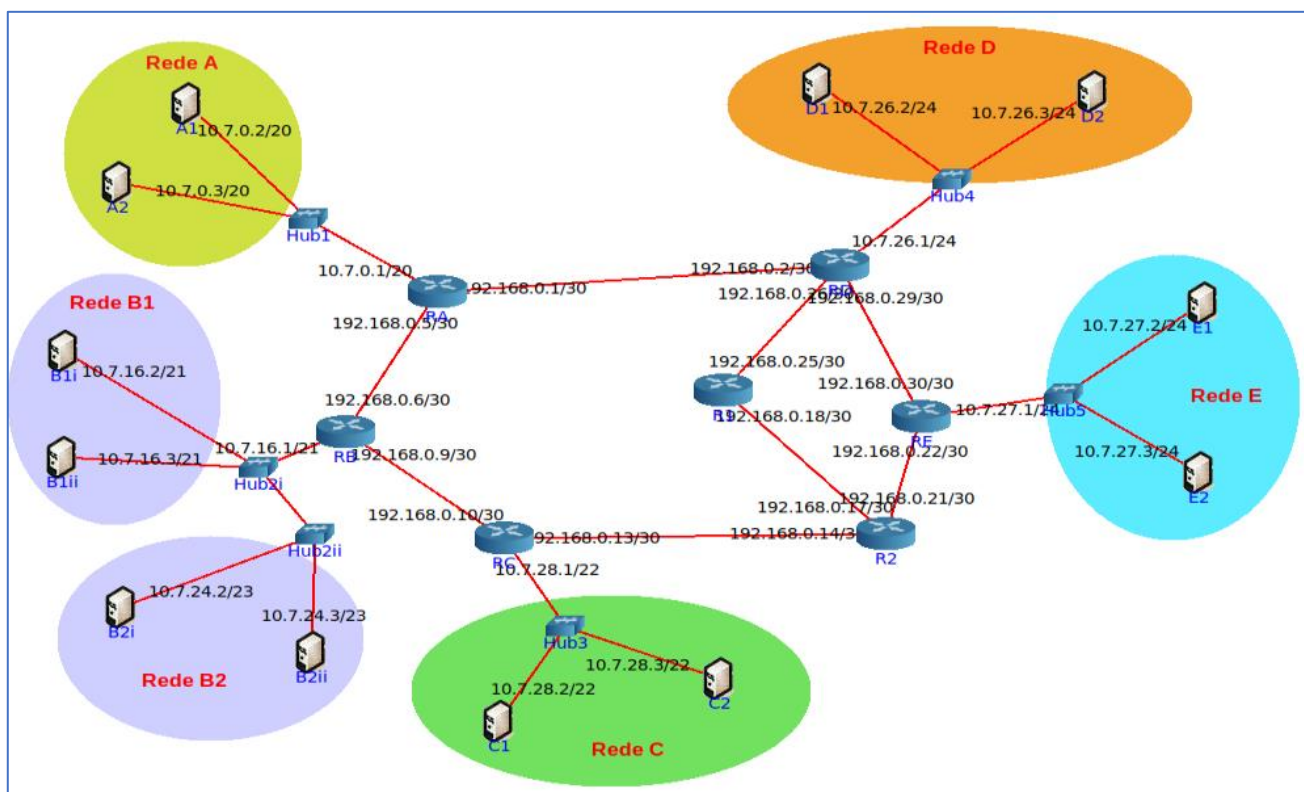


Figura 6 - Topologia da rede local da fase 2.

2.2. Atribuição dos endereços IP

Partindo do endereço IP 192.168.0.0/24 foram endereçadas as ligações entre os *routers* através de sub-redes com máscara de 30 bits. O facto de ser usado uma máscara de 30 bits é propositado uma vez que para o endereçamento da ligação são necessários apenas 4 endereços, o endereço de rede, de difusão e um endereço para cada nó da ligação. Para 4 endereços são necessários 2 bits fazendo com que os restantes 30 pertençam à sub-rede. Na figura seguinte estão presentes as denominações das ligações utilizadas entre os *routers*.

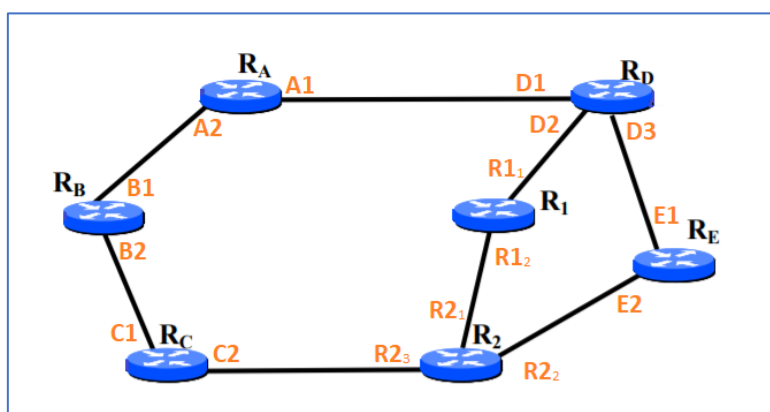


Figura 7 - Ligações entre os routers.

Na figura 8 demonstramos como chegámos ao endereçamento dos routers. Começamos por calcular a nossa máscara de rede ($M = 15+4 = 19$), como a máscara do nosso grupo dava 19, tínhamos 13 bits disponíveis para o endereçamento, o que correspondia quantitativamente a 8192 endereços disponíveis. Após isto calculamos o número de endereços a atribuir a cada rede indicados no canto superior esquerdo da figura 8. Finalmente realizámos o endereçamento das ligações apresentado na tabela 1.

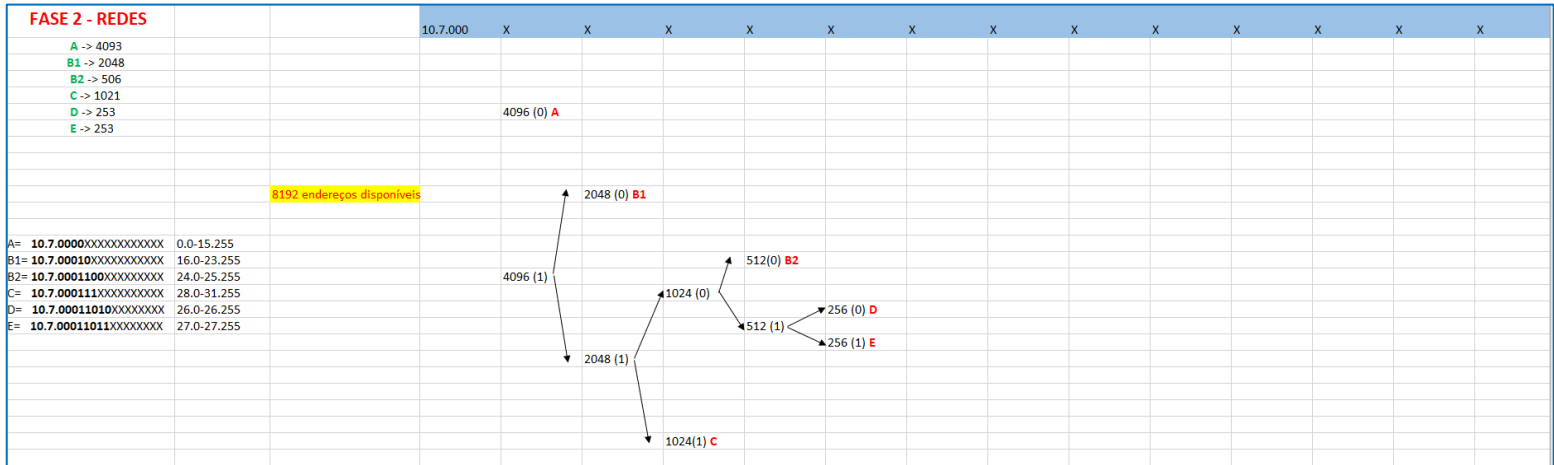


Figura 8 - Esquema para o endereçamento das ligações entre os routers.

Tabela 1 - Endereçamento das redes.

Rede	Número de endereços	IpRede	IpDifusão	Hosts	IpRouter
A	4093	10.7.00.0	10.7.15.255	10.7.0.1 - 10.7.15.254	10.7.00.1
B ₁	2048	10.7.16.0	10.7.23.255	10.7.16.1 - 10.7.23.254	10.7.16.1
B ₂	506	10.7.24.0	10.7.25.255	10.7.24.1 - 10.7.25.254	10.7.24.1
C	1021	10.7.28.0	10.7.31.255	10.7.28.1 - 10.7.31.254	10.7.28.1
D	253	10.7.26.0	10.7.26.255	10.7.26.1 - 10.7.26.254	10.7.26.1
E	253	10.7.27.0	10.7.27.255	10.7.27.1 - 10.7.27.254	10.7.27.1

Tabela 2 - Endereçamento das ligações entre os routers.

Ligação	IP Rede	Host	IP Difusão
A ₁ -D ₁	192.168.0.0	192.168.0.1 - 192.168.0.2	192.168.0.3
A ₂ -B ₁	192.168.0.4	192.168.0.5 - 192.168.0.6	192.168.0.7
B ₂ -C ₁	192.168.0.8	192.168.0.9 - 192.168.0.10	192.168.0.11
C ₂ -R ₂ ₃	192.168.0.12	192.168.0.13 - 192.168.0.14	192.168.0.15
R ₂ ₁ -R ₁ ₂	192.168.0.16	192.168.0.17 - 192.168.0.18	192.168.0.19
R ₂ ₂ -E ₂	192.168.0.20	192.168.0.21 - 192.168.0.22	192.168.0.23
R ₁ ₁ -D ₂	192.168.0.24	192.168.0.25 - 192.168.0.26	192.168.0.27
D ₃ -E ₁	192.168.0.28	192.168.0.29 - 192.168.0.30	192.168.0.31

Na tabela 2 está presente o endereçamento das ligações entre os routers, onde partimos do endereço IP 192.168.0.0 com máscara 255.255.255.252(/30) e preenchemos as ligações tendo em conta que precisávamos de um endereço de rede, um de difusão e 2 para a gama.

Na tabela 3 efetuamos o endereçamento de cada ponto da ligação entre os routers. Para isto utilizamos ambos os endereços da gama onde o primeiro corresponde à saída da ligação e o segundo à entrada no router destino.

Tabela 3 - Endereçamento de cada ponto da ligação.

Ligação	IP
A ₁	192.168.0.1
A ₂	192.168.0.5
B ₁	192.168.0.6
B ₂	192.168.0.9
C ₁	192.168.0.10
C ₂	192.168.0.13
D ₁	192.168.0.2
D ₂	192.168.0.26
D ₃	192.168.0.29
E ₁	192.168.0.30
E ₂	192.168.0.22
R1 ₁	192.168.0.25
R1 ₂	192.168.0.18
R2 ₁	192.168.0.17
R2 ₂	192.168.0.21
R2 ₃	192.168.0.14

2.3. Tabelas de encaminhamento

Feito o endereçamento dos routers efetuamos as tabelas de encaminhamento de modo a que fosse possível a comunicação entre os routers e os *hosts*. Nas tabelas seguintes são apresentadas estas tabelas para todos os *routers*.

Tabela 4 - Encaminhamento Router A.

ROUTER A				
Tabela de encaminhamento				
REDE	Rede Destino	Máscara	Interface Saída	Próximo Nó
A	10.7.00.0	/20	10.7.00.1	Direto
B1	10.7.16.0	/21	192.168.0.5	192.168.0.6
B2	10.7.24.0	/23	192.168.0.5	192.168.0.6
C	10.7.28.0	/22	192.168.0.5	192.168.0.6
D	10.7.26.0	/24	192.168.0.1	192.168.0.2
E	10.7.27.0	/24	192.168.0.1	192.168.0.2

Tabela 5 - Encaminhamento Router B.

ROUTER B				
Tabela de encaminhamento				
REDE	Rede Destino	Máscara	Interface Saída	Próximo Nó
A	10.7.00.0	/20	192.168.0.6	192.168.0.5
B1	10.7.16.0	/21	10.7.16.1	Direto
B2	10.7.24.0	/23	10.7.24.1	Direto
C	10.7.28.0	/22	192.168.0.9	192.168.0.10
D	10.7.26.0	/24	192.168.0.6	192.168.0.5
E	10.7.27.0	/24	192.168.0.9	192.168.0.10

Tabela 6 - Encaminhamento Router C.

ROUTER C				
Tabela de encaminhamento				
REDE	Rede Destino	Máscara	Interface Saída	Próximo Nó
A	10.7.00.0	/20	192.168.0.10	192.168.0.9
B1	10.7.16.0	/21	192.168.0.10	192.168.0.9
B2	10.7.24.0	/23	192.168.0.10	192.168.0.9
C	10.7.28.0	/22	10.7.28.1	Direto
D	10.7.26.0	/24	192.168.0.13	192.168.0.14
E	10.7.27.0	/24	192.168.0.13	192.168.0.14

Tabela 7 - Encaminhamento Router D.

ROUTER D				
Tabela de encaminhamento				
REDE	Rede Destino	Máscara	Interface Saída	Próximo Nó
A	10.7.00.0	/20	192.168.0.2	192.168.0.1
B1	10.7.16.0	/21	192.168.0.2	192.168.0.1
B2	10.7.24.0	/23	192.168.0.2	192.168.0.1
C	10.7.28.0	/22	192.168.0.26	192.168.0.25
D	10.7.26.0	/24	10.7.26.1	Direto
E	10.7.27.0	/24	192.168.0.29	192.168.0.30

Tabela 8 - Encaminhamento Router E.

ROUTER E				
Tabela de encaminhamento				
REDE	Rede Destino	Máscara	Interface Saída	Próximo Nó
A	10.7.00.0	/20	192.168.0.30	192.168.0.29
B1	10.7.16.0	/21	192.168.0.22	192.168.0.21
B2	10.7.24.0	/23	192.168.0.22	192.168.0.21
C	10.7.28.0	/22	192.168.0.22	192.168.0.21
D	10.7.26.0	/24	192.168.0.30	192.168.0.29
E	10.7.27.0	/24	10.7.27.1	Direto

Tabela 9 - Encaminhamento Router R1.

ROUTER R1				
Tabela de encaminhamento				
REDE	Rede Destino	Máscara	Interface Saída	Próximo Nó
A	10.7.00.0	/20	192.168.0.25	192.168.0.26
B1	10.7.16.0	/21	192.168.0.25	192.168.0.26
B2	10.7.24.0	/23	192.168.0.25	192.168.0.26
C	10.7.28.0	/22	192.168.0.18	192.168.0.17
D	10.7.26.0	/24	192.168.0.25	192.168.0.26
E	10.7.27.0	/24	192.168.0.18	192.168.0.17

Tabela 10 - Encaminhamento Router R2.

ROUTER R2				
Tabela de encaminhamento				
REDE	Rede Destino	Máscara	Interface Saída	Próximo Nó
A	10.7.00.0	/20	192.168.0.17	192.168.0.18
B1	10.7.16.0	/21	192.168.0.14	192.168.0.13
B2	10.7.24.0	/23	192.168.0.14	192.168.0.13
C	10.7.28.0	/22	192.168.0.14	192.168.0.13
D	10.7.26.0	/24	192.168.0.17	192.168.0.18
E	10.7.27.0	/24	192.168.0.21	192.168.0.22

2.4. Encaminhamento estático

Após a realização das tabelas descritas anteriormente tivemos de configurar o serviço *zebra* do CORE de modo a efetuar um encaminhamento estático, ou seja, colocar manualmente os caminhos dos *routers*. Nas figuras seguintes apresentamos a configuração *zebra* do router A e do router B como demonstração de exemplo. Através da observação da figura 9 vemos que a interface *eth0* contém 2 endereços, isto deve-se ao facto de a nossa rede B ter 2 sub-redes devido ao seu endereçamento.

Router A

```
interface eth0
 ip address 10.7.0.1/20
!
interface eth1
 ip address 192.168.0.5/30
!
interface eth2
 ip address 192.168.0.1/30
!
ip route 10.7.16.0/21 192.168.0.6
ip route 10.7.24.0/23 192.168.0.6
ip route 10.7.28.0/22 192.168.0.6
ip route 10.7.26.0/24 192.168.0.2
ip route 10.7.27.0/24 192.168.0.2
```

Figura 10 - Configuração zebra Router A.

Router B

```
interface eth0
 ip address 10.7.16.1/21
 ip address 10.7.24.1/23
!
interface eth1
 ip address 192.168.0.6/30
!
interface eth2
 ip address 192.168.0.9/30
!
ip route 10.7.0.0/20 192.168.0.5
ip route 10.7.28.0/22 192.168.0.10
ip route 10.7.26.0/24 192.168.0.5
ip route 10.7.27.0/24 192.168.0.10
```

Figura 9 - Configuração zebra Router B.

Após alterarmos manualmente as rotas necessárias para garantir a conectividade IPV4 entre todas as redes, realizamos testes de *ping* e *traceroute* para garantir que tudo estava a funcionar corretamente. Como demonstração, nas figuras seguintes apresentamos os testes feitos entre o PC B2i e o PC E1 da figura 6.

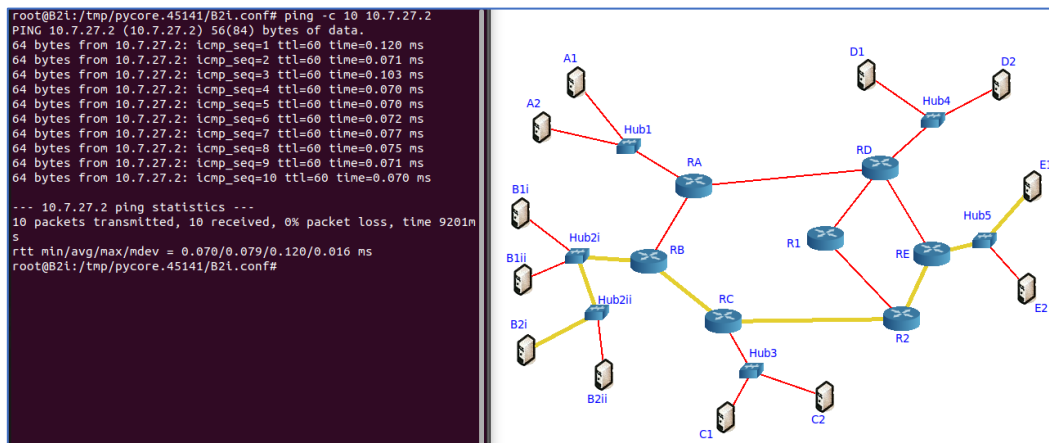


Figura 11 - Teste ping entre B2i e E1.

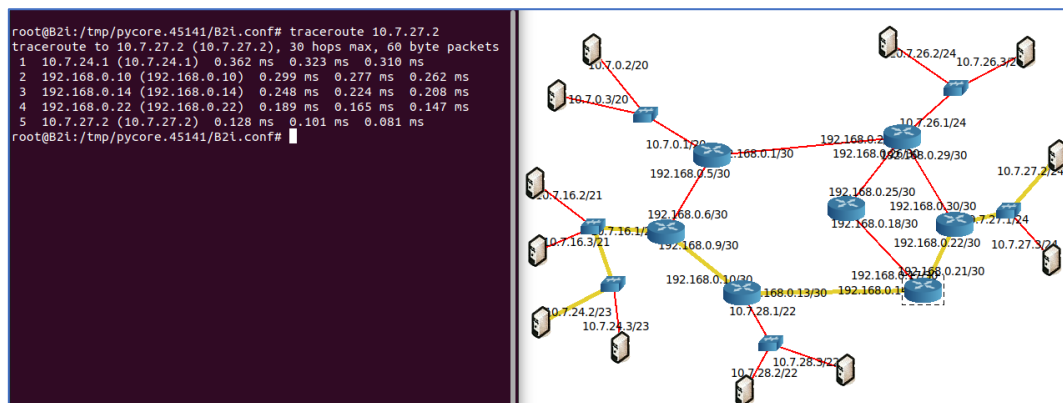


Figura 12 - Teste traceroute entre B2i e E1.

3. DHCP – *Dynamic Host Configuration Protocol*

3.1. Configuração do Protocolo DHCP

Nesta fase utilizamos o protocolo DHCP na rede B da nossa topologia de modo a observar o seu funcionamento e a forma como o endereçamento é realizado. Através deste protocolo é possível uma configuração automática e dinâmica dos elementos de uma rede. Para que isto fosse possível tivemos de adicionar um *host* que tratava do endereçamento através do protocolo DHCP. Na figura seguinte apresentamos a nova topologia da rede B onde tanto para a sub-rede B1 e B2 adicionamos um servidor, *serverB1* para a sub-rede B1 e *serverB2* para B2.

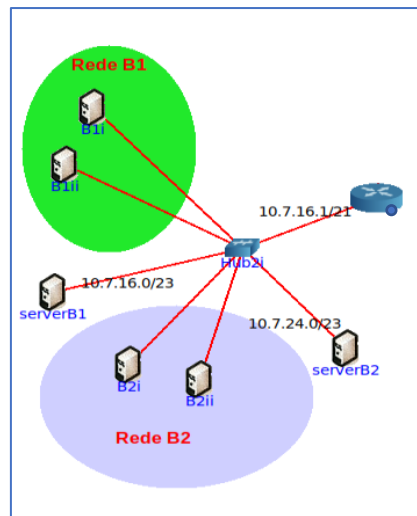


Figura 13 - Nova topologia da rede B.

De seguida passamos à configuração de todos os *hosts* removendo os IPs dos que iam ser *clients* e configurando os que iam servir como *servers*, na figura 15 está o exemplo do *serverB1*. Para os *clients* também tivemos de habilitar o DHCP *Client* a partir do CORE (Figura 14).

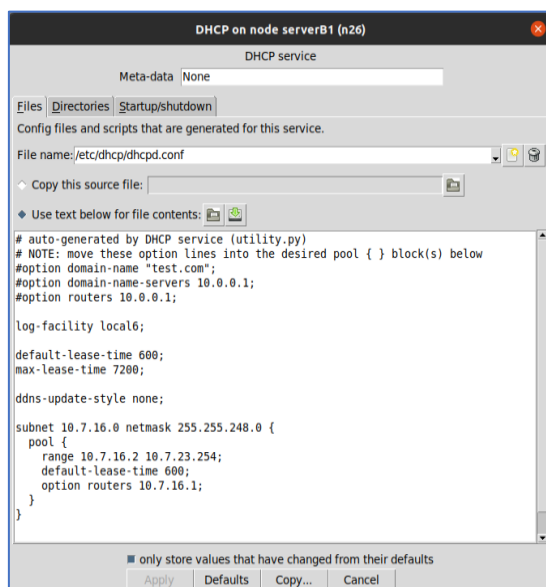


Figura 14 - Configuração DHCP server.

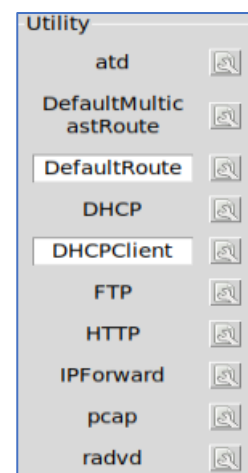


Figura 15 - Configuração DHCP Client.

3.2. Testes DHCP

Depois de feita a configuração do DHCP em todos os elementos da rede B abrimos o Wireshark de modo a observar o funcionamento do protocolo. Na figura 16 apresentamos o resultado. Desta figura retiramos que foi realizado a correta configuração de todos os elementos visto que o protocolo está a funcionar como era suposto. Este começa por enviar um DHCP *Discover* [1] do cliente para o servidor, que responde com um DHCP *Offer* [2]. Após esta troca, o cliente pede um endereço IP ao servidor através do DHCP *Request* [3] que responde com um DHCP *ACK* [4] em que o servidor envia o endereço IP ao cliente.

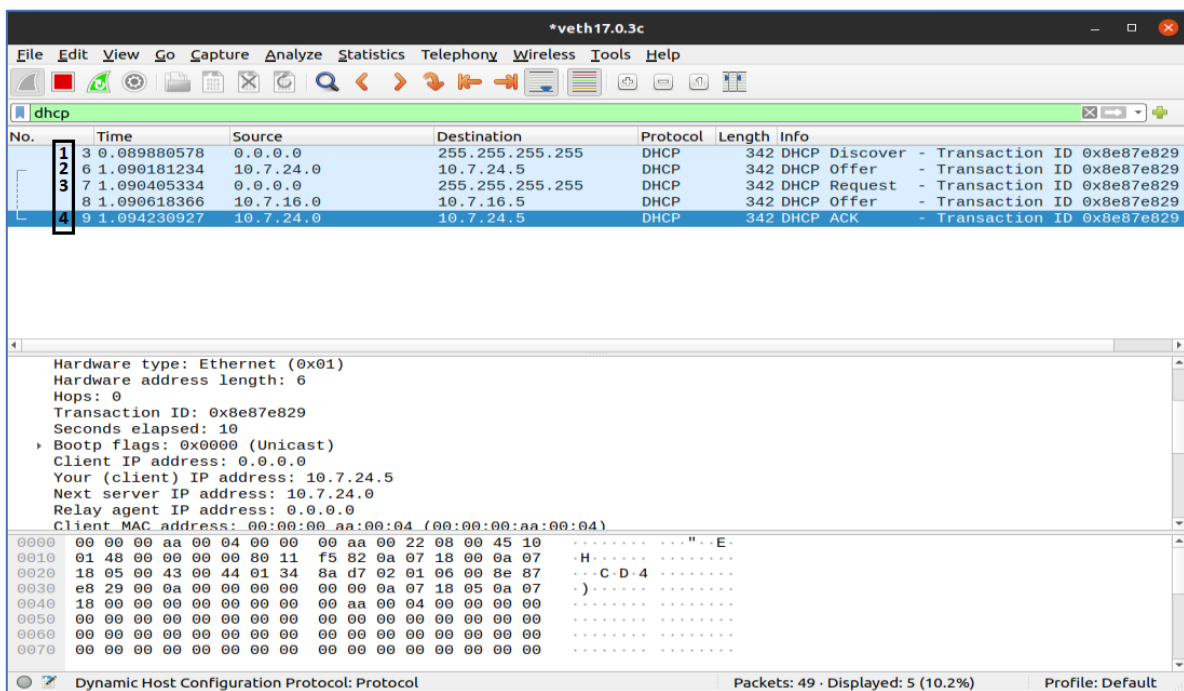


Figura 16 - Wireshark DHCP.

Após feita a observação no Wireshark utilizamos o comando *ifconfig* para observar os IPs atribuídos aos PCs. Na figura 17 estão presentes as janelas obtidas onde conseguimos ver que foi atribuído ao PC B2i o endereço IP 10.7.24.2 e ao PC B2ii o endereço 10.7.24.3.

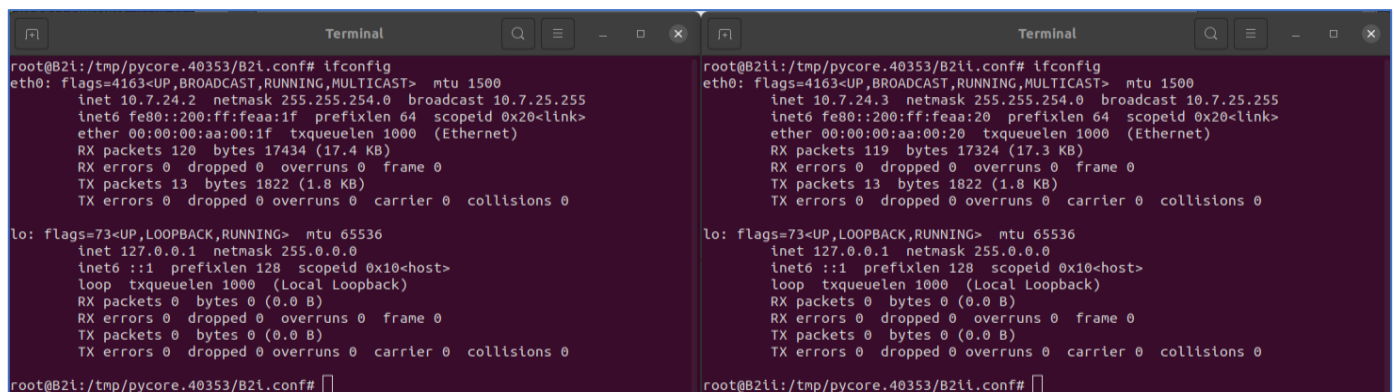


Figura 17 - Comando ifconfig.

4. Uso das camadas de rede e transporte pelas aplicações

4.1. Ativação do servidor HTTP num PC da rede C

Nesta fase começamos por ativar um servidor HTTP no PC C1 configurando-o como mostrado na figura 18. Após isso realizamos o teste para ver se estava tudo a funcionar corretamente através do comando *wget* (Figura 19). Para isto utilizamos o PC E1 para enviar e o PC C1, cujo servidor tinha sido configurado previamente, para receber.

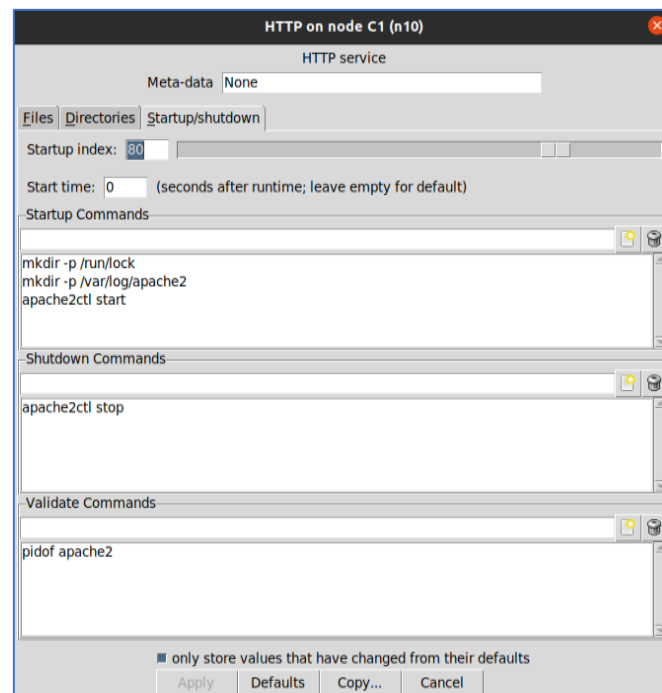


Figura 18 - Configuração servidor HTTP.

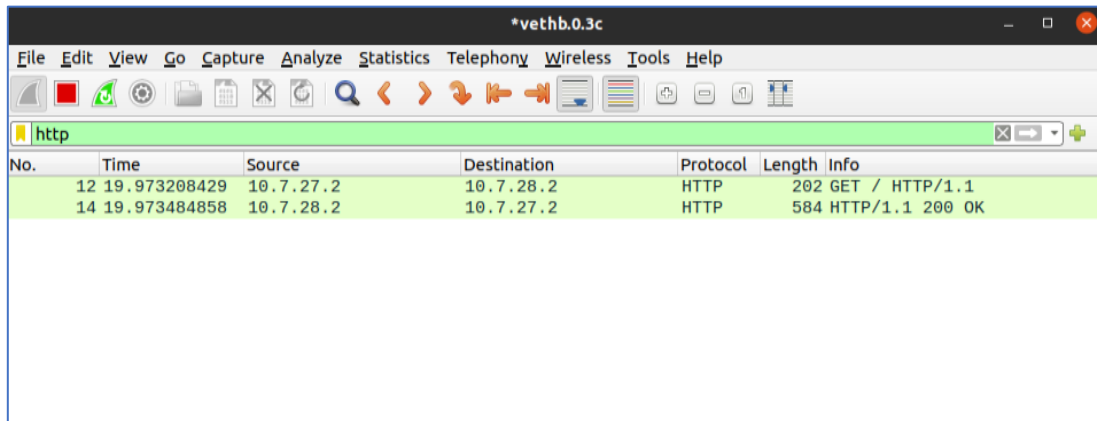
```
Terminal
root@E1:/tmp/pycore.40353/E1.conf# wget -S 10.7.28.2
--2021-01-14 23:42:59-- http://10.7.28.2/
Connecting to 10.7.28.2:80... connected.
HTTP request sent, awaiting response...
HTTP/1.1 200 OK
Date: Thu, 14 Jan 2021 23:42:59 GMT
Server: Apache/2.4.41 (Ubuntu)
Last-Modified: Thu, 14 Jan 2021 23:42:42 GMT
ETag: "102-5b8e4cfd9a0"
Accept-Ranges: bytes
Content-Length: 258
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Length: 258
Saving to: 'index.html'

index.html      100%[=====>]      258  --.-KB/s   in 0s
2021-01-14 23:42:59 (48,6 MB/s) - 'index.html' saved [258/258]
root@E1:/tmp/pycore.40353/E1.conf#
```

Figura 19 - Servidor HTTP.

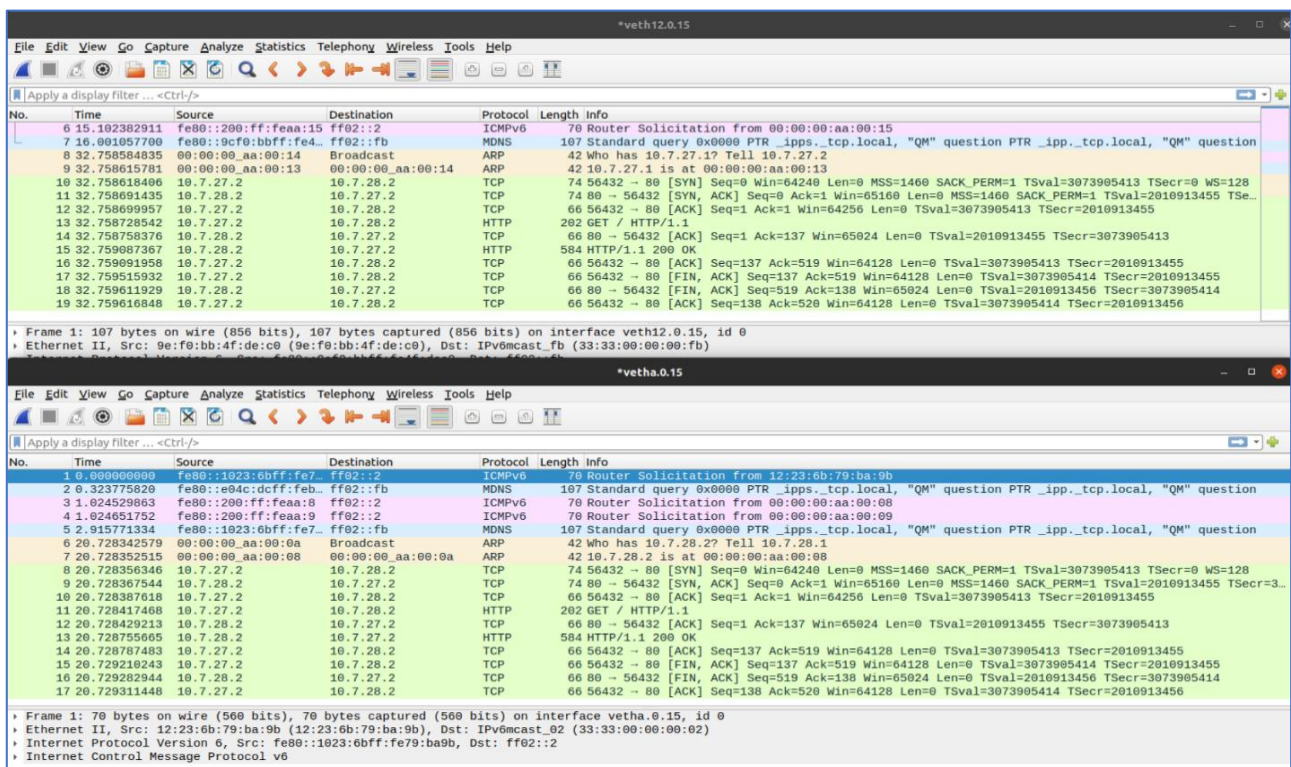
4.2. Testes

Após ter sido feito o teste para ver se estava tudo a correr como desejado decidimos passar para o Wireshark de modo a observar a captura de tráfego no PC C1. Na figura 20 visualizamos essa mesma captura onde conseguimos visualizar 2 linhas relativas ao HTTP, em que a primeira corresponde ao envio de um pedido HTTP pelo *host* origem e a segunda a sua devida resposta pelo *host* destino. Na figura 21 temos a observação da conexão TCP.



No.	Time	Source	Destination	Protocol	Length	Info
12	19.973208429	10.7.27.2	10.7.28.2	HTTP	202	GET / HTTP/1.1
14	19.973484858	10.7.28.2	10.7.27.2	HTTP	584	HTTP/1.1 200 OK

Figura 20 - Wireshark HTTP.



No.	Time	Source	Destination	Protocol	Length	Info
6	15.102382911	fe80::200:ff:feaa:15	ff02::2	ICMPv6	70	Router Solicitation from 00:00:00:aa:00:15
7	16.001057700	fe80::9cf0:bbff:fe4...	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR _ipp._tcp.local, "QM" question
8	32.758584835	00:00:00:aa:00:14	Broadcast	ARP	42	Who has 10.7.27.1? Tell 10.7.27.2
9	32.758615781	00:00:00:aa:00:13	00:00:00:aa:00:14	ARP	42	10.7.27.1 is at 00:00:00:aa:00:13
10	32.758618406	10.7.27.2	10.7.28.2	TCP	74	56432 -> 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3073905413 TSecr=0 WS=128
11	32.758691435	10.7.28.2	10.7.27.2	TCP	74	80 -> 56432 [ACK] Seq=1 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=2010913455 TSecr=3073905413
12	32.758699957	10.7.27.2	10.7.28.2	TCP	66	56432 -> 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3073905413 TSecr=2010913455
13	32.758728542	10.7.27.2	10.7.28.2	HTTP	202	GET / HTTP/1.1
14	32.758758376	10.7.28.2	10.7.27.2	TCP	66	80 -> 56432 [ACK] Seq=1 Ack=137 Win=65024 Len=0 TSval=2010913455 TSecr=3073905413
15	32.759087367	10.7.28.2	10.7.27.2	HTTP	584	HTTP/1.1 200 OK
16	32.759091958	10.7.27.2	10.7.28.2	TCP	66	56432 -> 80 [ACK] Seq=137 Ack=519 Win=64128 Len=0 TSval=3073905413 TSecr=2010913455
17	32.759515932	10.7.27.2	10.7.28.2	TCP	66	56432 -> 80 [FIN, ACK] Seq=137 Ack=519 Win=64128 Len=0 TSval=3073905413 TSecr=2010913455
18	32.759611929	10.7.28.2	10.7.27.2	TCP	66	80 -> 56432 [FIN, ACK] Seq=519 Ack=138 Win=65024 Len=0 TSval=2010913456 TSecr=3073905414
19	32.759616848	10.7.27.2	10.7.28.2	TCP	66	56432 -> 80 [ACK] Seq=138 Ack=520 Win=64128 Len=0 TSval=3073905414 TSecr=2010913456

Frame 1: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface veth12.0.15, id 0
Ethernet II, Src: 9e:f0:bb:4f:de:c0 (9e:f0:bb:4f:de:c0), Dst: IPv6mcast_fb (33:33:00:00:00:fb)

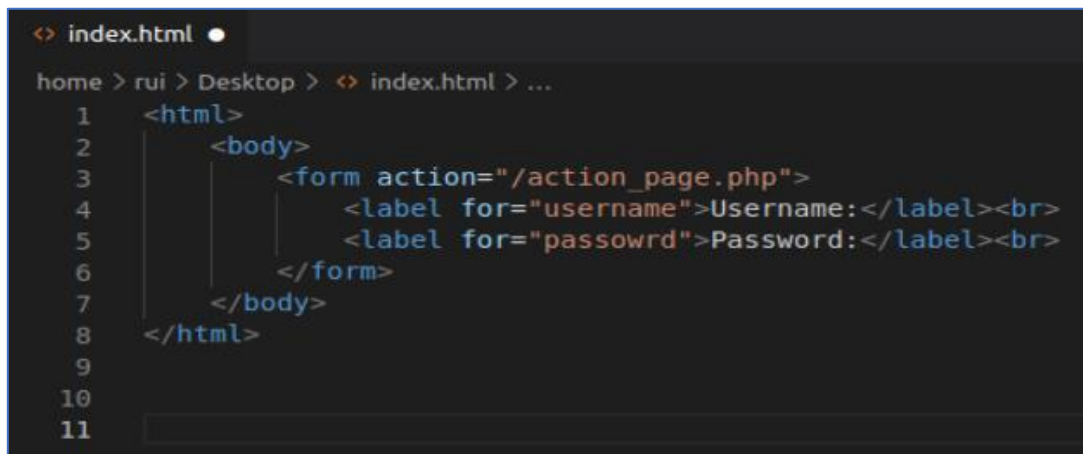
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fe80::1023:6bff:fe7...	ff02::2	ICMPv6	70	Router Solicitation from 12:23:6b:79:ba:9b
2	0.323775829	fe80::e94c:dcff:feb...	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR _ipp._tcp.local, "QM" question
3	1.024529863	fe80::200:ff:feaa:9	ff02::2	ICMPv6	70	Router Solicitation from 00:00:00:aa:00:09
4	1.024651752	fe80::200:ff:feaa:9	ff02::2	ICMPv6	70	Router Solicitation from 00:00:00:aa:00:09
5	2.915771334	fe80::1023:6bff:fe7...	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR _ipp._tcp.local, "QM" question
6	20.728342579	00:00:00:aa:00:0a	Broadcast	ARP	42	Who has 10.7.28.2? Tell 10.7.28.1
7	20.728352515	00:00:00:aa:00:0a	00:00:00:aa:00:0a	ARP	42	10.7.28.2 is at 00:00:00:aa:00:0a
8	20.728356346	10.7.27.2	10.7.28.2	TCP	74	56432 -> 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3073905413 TSecr=0 WS=128
9	20.728367544	10.7.28.2	10.7.27.2	TCP	74	80 -> 56432 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=2010913455 TSecr=3073905413
10	20.728387618	10.7.27.2	10.7.28.2	TCP	66	56432 -> 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3073905413 TSecr=2010913455
11	20.728417468	10.7.27.2	10.7.28.2	HTTP	202	GET / HTTP/1.1
12	20.728429213	10.7.28.2	10.7.27.2	TCP	66	80 -> 56432 [ACK] Seq=1 Ack=137 Win=65024 Len=0 TSval=2010913455 TSecr=3073905413
13	20.728755665	10.7.28.2	10.7.27.2	HTTP	584	HTTP/1.1 200 OK
14	20.728767483	10.7.27.2	10.7.28.2	TCP	66	56432 -> 80 [ACK] Seq=137 Ack=519 Win=64128 Len=0 TSval=3073905413 TSecr=2010913455
15	20.729218243	10.7.27.2	10.7.28.2	TCP	66	56432 -> 80 [FIN, ACK] Seq=137 Ack=519 Win=64128 Len=0 TSval=3073905414 TSecr=2010913455
16	20.729282944	10.7.28.2	10.7.27.2	TCP	66	80 -> 56432 [FIN, ACK] Seq=519 Ack=138 Win=65024 Len=0 TSval=2010913456 TSecr=3073905414
17	20.729311448	10.7.27.2	10.7.28.2	TCP	66	56432 -> 80 [ACK] Seq=138 Ack=520 Win=64128 Len=0 TSval=3073905414 TSecr=2010913456

Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface vetha.0.15, id 0
Ethernet II, Src: 12:23:6b:79:ba:9b (12:23:6b:79:ba:9b), Dst: IPv6mcast_fb (33:33:00:00:00:02)
Internet Protocol Version 6, Src: fe80::1023:6bff:fe79:ba9b, Dst: ff02::2
Internet Control Message Protocol v6

Figura 21 - Wireshark Protocolo TCP/IP.

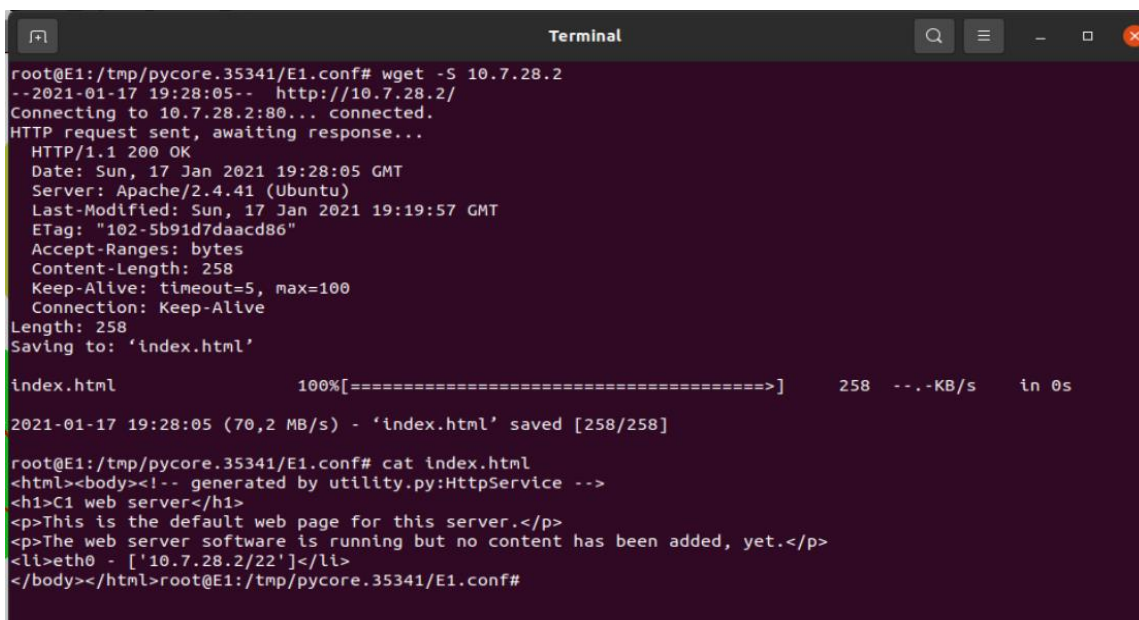
4.3. Formulário HTTP

Nesta fase opcional do projeto tivemos de criar um formulário http (Figura 22) e adicionamos ao servidor de modo a que todos os *hosts* pudessem aceder a este. Após isso realizamos testes através de vários sistemas terminais, no entanto sem sucesso visto que o *host* recebia um formulário, mas não o que nós tínhamos criado, recebia sim um formulário *default* como observado na figura 23.



```
<> index.html ●
home > rui > Desktop > <> index.html > ...
1  <html>
2      <body>
3          <form action="/action_page.php">
4              <label for="username">Username:</label><br>
5              <label for="password">Password:</label><br>
6          </form>
7      </body>
8  </html>
9
10
11
```

Figura 22 - Formulário HTTP.



```
Terminal
root@E1:/tmp/pycore.35341/E1.conf# wget -S 10.7.28.2
--2021-01-17 19:28:05-- http://10.7.28.2/
Connecting to 10.7.28.2:80... connected.
HTTP request sent, awaiting response...
HTTP/1.1 200 OK
Date: Sun, 17 Jan 2021 19:28:05 GMT
Server: Apache/2.4.41 (Ubuntu)
Last-Modified: Sun, 17 Jan 2021 19:19:57 GMT
ETag: "102-5b91d7daacd86"
Accept-Ranges: bytes
Content-Length: 258
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Length: 258
Saving to: 'index.html'

index.html          100%[=====]          258  --.-KB/s   in 0s

2021-01-17 19:28:05 (70,2 MB/s) - 'index.html' saved [258/258]

root@E1:/tmp/pycore.35341/E1.conf# cat index.html
<html><body><!-- generated by utility.py:HttpService -->
<h1>C1 web server</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content has been added, yet.</p>
<li>eth0 - ['10.7.28.2/22']</li>
</body></html>root@E1:/tmp/pycore.35341/E1.conf#
```

Figura 23 - Teste do formulário.

Conclusão

Com a realização deste trabalho ficamos a conhecer novas ferramentas de simulação de redes como o CORE bem como na análise e captura de tráfego entre sistemas como o Wireshark. Este trabalho serviu também para um estudo mais aprofundado através de exemplos práticos do endereçamento de redes e da utilização de protocolos como o TCP e IP, o DHCP e o ARP.

Pensamos ter sido bem-sucedidos na elaboração do trabalho, no entanto em relação à sua parte opcional, a parte da criação do formulário HTTP tivemos bastantes questões e não conseguimos executar nem cumprir na totalidade com o que era desejado.