Trabalho prático: LANs Ethernet e redes TCP/IP usando o CORE

Redes de Computadores I

Mestrado Integrado em Engenharia de Telecomunicações e Informática

Hélder Duarte da Costa Freitas A65225

João Pedro Mendes Pereira A68454

Guimarães, Janeiro de 2016

Índice

Índice de Figuras

1. Introdução

Na Unidade Curricular de Redes de Computadores I e, com base no método de avaliação, foi-nos proposto realizar uma implementação de várias topologias com o intuito de emular na ferramenta CORE vários tipos de redes e interligá-las entre si. Esta ferramenta será utilizada para criar e desenhar as topologias de redes, configurar endereços, ligações e serviços de cada uma das máquinas.

Cada topologia enunciada no projeto terá diferentes objectivos de aprendizagem, diferentes configurações e máquinas a ela associadas, tendo como objectivo efectuar o diagnóstico de conectividade e a análise de capturas de tráfego (utilizando a ferramenta Wireshark) e, posteriormente, comparar os modos de funcionamento. Para a realização destes testes iremos utilizar comandos como o "*ping*" e "*traceroute*", entre outros.

2. Emulação de LANs Ethernet

No primeiro exercício, deste capítulo, deparamo-nos com a necessidade de entender o funcionamento de HUBs e SWITCHs, assim como entender os protocolos ARP e ICMP envolvidos.

O protocolo ARP, *Adress Resolution Protocol*, destina-se a mapear um endereço de IP, *Internet Protocol*, para um endereço de máquina física ou MAC, *Media Access* Control, recorrendo a um ARP *request* e ARP reply. Neste protocolo é mantida, nos clientes, uma tabela, denominada por *cache* ARP, usada para manter a correlação entre cada endereço MAC e o endereço IP correspondente de modo a reduzir o número de solicitações de resolução de endereços permitindo um melhor fluxo.

O protocolo ICMP, *Internet Control Message Protocol*, é um protocolo utilizado por *routers,* dispositivos intermediários e *hosts* para diagnóstico e relatório de erros, considerado necessário em qualquer implementação IP. Estes dispositivos enviam mensagens ICMP á fonte do pacote que recebeu em diversas situações, como por exemplo, tempo de transmissão do pacote excedido, congestionamento do dispositivo ou, simplesmente, quando o dispositivo pode enviar o pacote por um caminho mais curto. Alguns dos utilitários de diagnóstico que utilizam este protocolo são o *ping* e *traceroute*.

O HUB assim que recebe informação numa determinada porta, reencaminha essa informação para todas as outras portas criando um único domínio de colisão e diminuindo a performance, uma vez que será possível, em qualquer sistema terminal fora da conversa, ter acesso a informação transmitida no HUB. Este dispositivo não consegue, nem tem funcionalidades, que lhe permita guardar informação relativamente ás maquinas a ele conectadas.

Ao contrário do HUB, o SWITCH permite que após uma primeira ligação entre sistemas (funcionamento igual ao HUB nesta primeira fase), seja registado o endereço MAC destes conectados a cada porta. Isto permite que o SWITCH comute o pacote directamente para a porta de destino da mesma.

Após o estudo dos conceitos, começamos por desenhar e emular uma topologia de rede local em estrela, utilizando um HUB, que será designada por topologia A (Figura I).

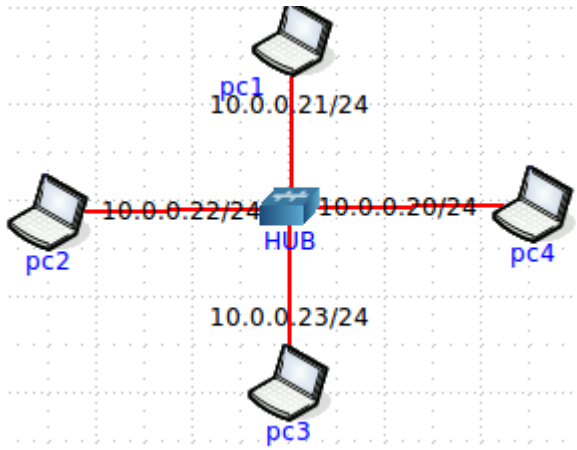


Figura 1: Topologia A

De modo a testar a conectividade entre os sistemas terminais, irá ser utilizado o comando "*ping* 10.0.0.23" no pc1 e iremos colocar o Wireshark em escuta no pc3 (máquina que irá ser "pingada" pelo pc1). O comando "*arp -a*" irá ser necessário para verificar as tabelas ARP dos sistemas terminais. Desde já, podemos antecipar alguns resultados experimentais, como por exemplo, as tabelas irão estar vazias e irá ser solicitado, em *broadcast,* um ARP *request* de modo a saber qual dos sistemas terminais possui o endereço MAC correspondente ao endereço IP do ping, sendo posteriormente retornado um ARP *reply,* em unicast, para o sistema terminal que efectuou o pedido. Quando é estabelecida uma comunicação entre sistemas, as tabelas dos mesmos são actualizadas e, caso haja nova necessidade de comunicação entre os dois sistemas, estes já irão efectuar a comunicação ainda antes de haver necessidade de o HUB efectuar o ARP *request*.

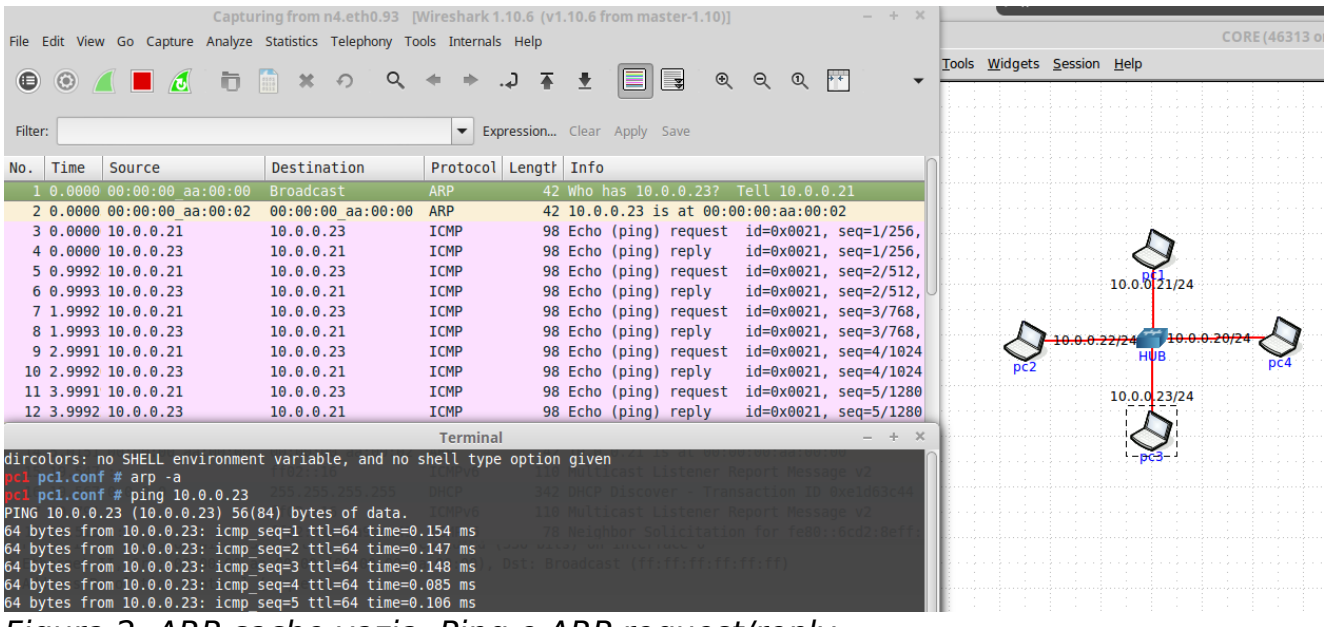


Figura 2: ARP cache vazia. Ping e ARP request/reply

Tal como podemos observar na Figura 2, o resultado obtido bate certo com o resultado esperado, ou seja, no inicio da comunicação, o pc1 envia, em broadcast, um ARP *request*, a perguntar quem tem o endereço MAC correspondente ao endereço IP 10.0.0.22. Em seguida o pc3 retorna, em unicast, um ARP *reply* indicando o seu endereço MAC permitindo assim a troca de informação entre os dois sistemas terminais. Após esta comunicação, as tabelas dos pc1 e pc3 foram actualizadas, enquanto que, as dos pc2 e pc4 permanecem vazias e, também, é possível verificar que quando é efectuado um novo *ping* entre os mesmos sistemas, estes começam a comunicar imediatamente entre si sendo enviado o ARP request após parte dos pacotes já terem sido efectuados, conforme mostra a Figura 3.

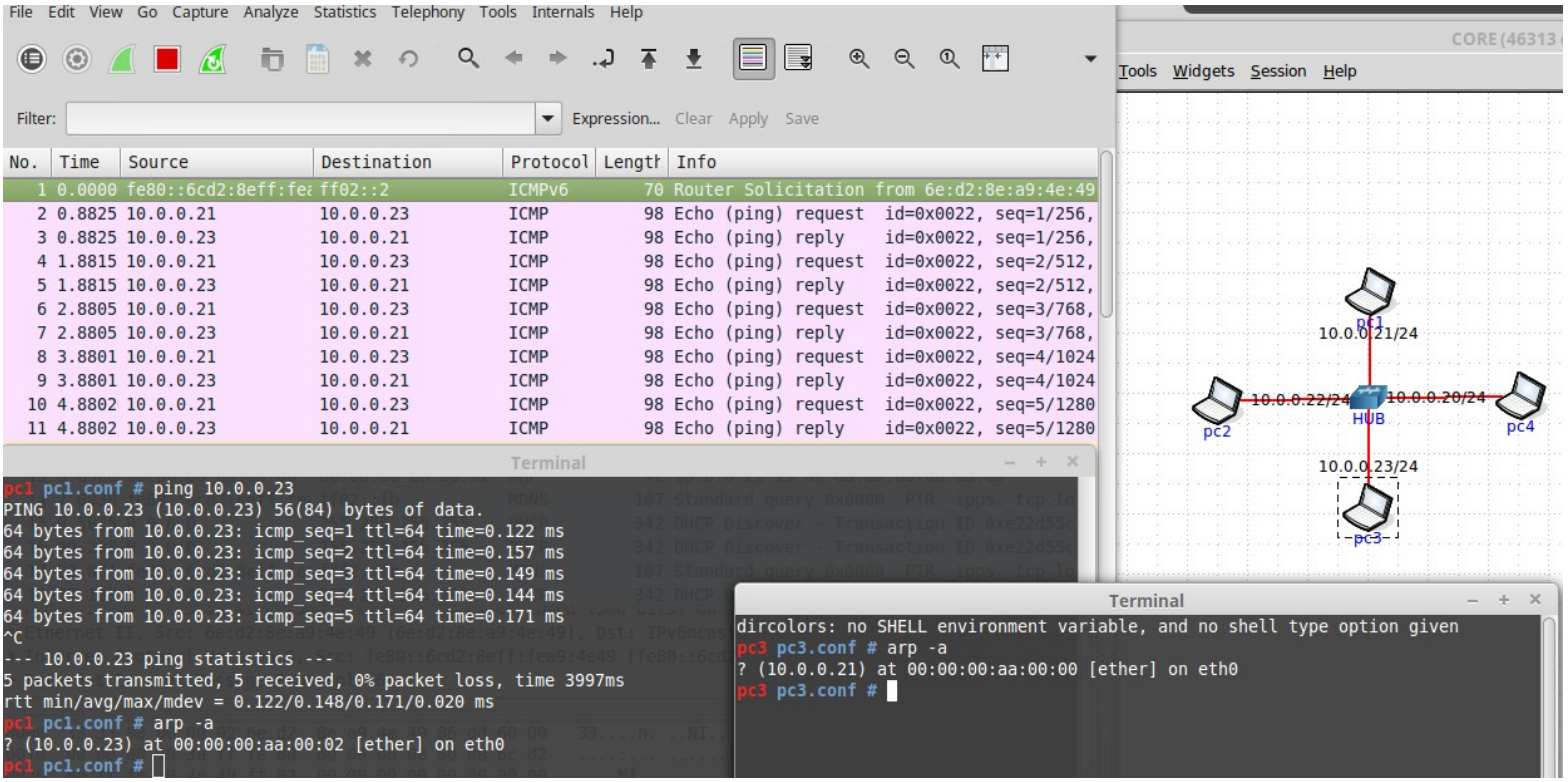


Figura 3: ARP cache atualizada. Comunicação imediata entre os dois sistemas terminais

No segundo exercício é-nos pedido para alterar o HUB, da topologia anterior, por um SWITCH e verificar as diferenças entre estes dois equipamentos. A principal diferença, nesta nova topologia, será o redirecionamento dos pacotes para a porta respectiva, evitando assim que qualquer outro sistema terminal conectado ao SWITCH seja sobrecarregado com informação inútil ao mesmo.

Na Figura 4 é efetuado, no pc1, o comando "*arp -a*" e de seguida o comando "*ping* 10.0.0.23" sendo possível verificar no Wireshark que, não existindo ligação anterior entre estas duas máquinas, o SWITCH funciona, numa primeira instância, da mesma maneira que o HUB, enviando por *broadcast* um pedido de correspondência entre o endereço IP e o endereço

MAC, no entanto, imediatamente a seguir a uma resposta positiva por parte de um sistema, este reencaminha todos os pacotes para a porta a qual corresponde esse endereço MAC o que faz com que o pc4 (e também no pc2) não escute a conversa entre estes dois sistemas.

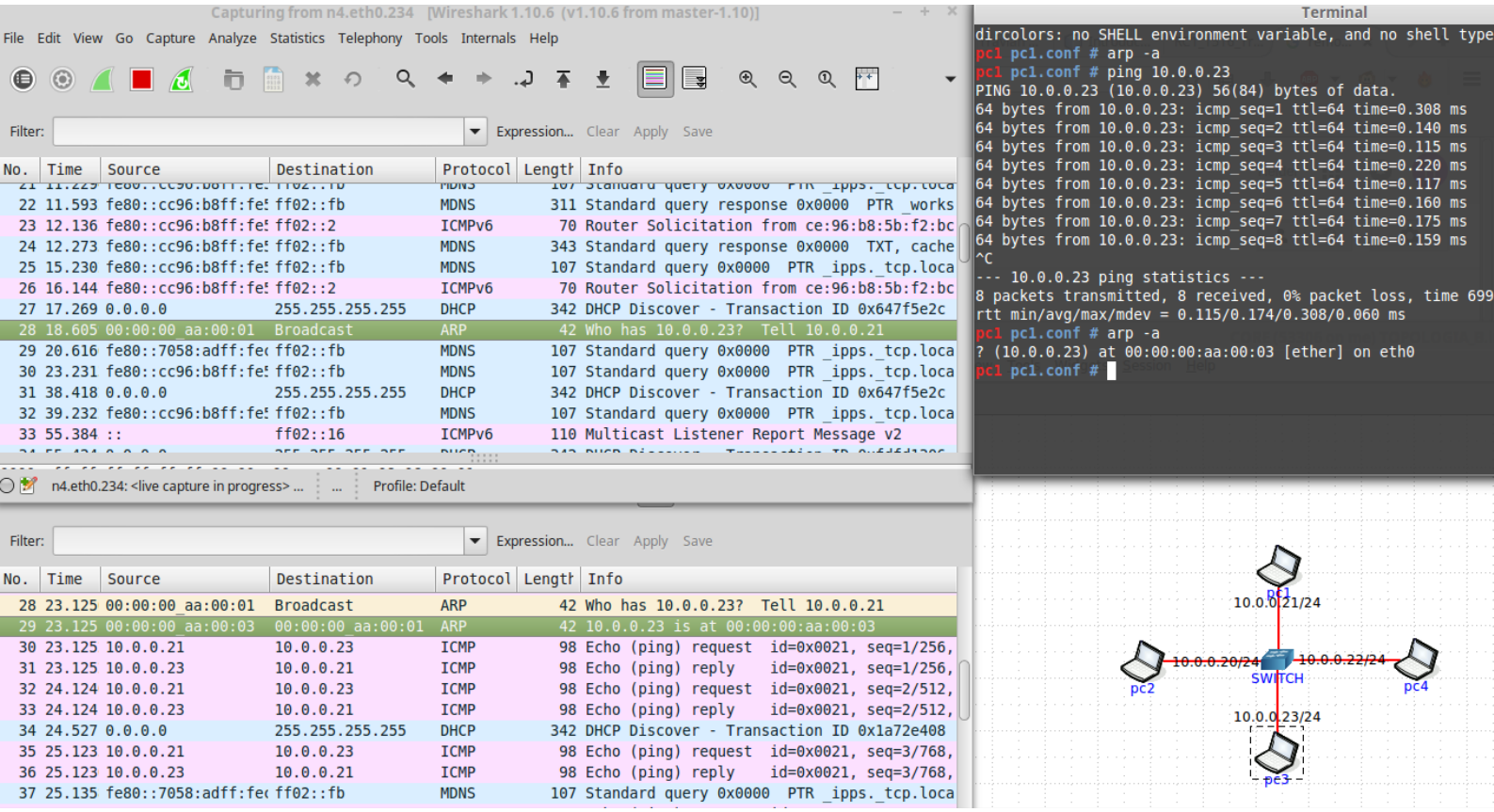


Figura 4: Ping entre pc1 e pc3. Wireshark no pc4 (canto superior esquerdo) e no pc3 (canto inferior esquerdo)

Com esta topologia é possível concluir que a utilização de um SWITCH torna a rede mais eficiente e, principalmente nos dias de hoje em que é um tema bastante debatido, permite um nível de privacidade elevado entre sistemas terminais.

Para o último exercício é necessário criar uma topologia onde esteja aplicado, em árvore, um rede local com HUBs e SWITCHs. Como se pode verificar na Figura 5, foi efetuado um *ping* entre um sistema conectado ao HUB (n5) e outro conectado a um SWITCH (n11) e capturados os pacotes em outros dois dispositvos, n4 (conectado ao HUB) e n10 (conectado ao SWITCH) pelo que, o n4 captura toda a conversa entre os sistemas e o n10 apenas captura o *broadcast* inicial.

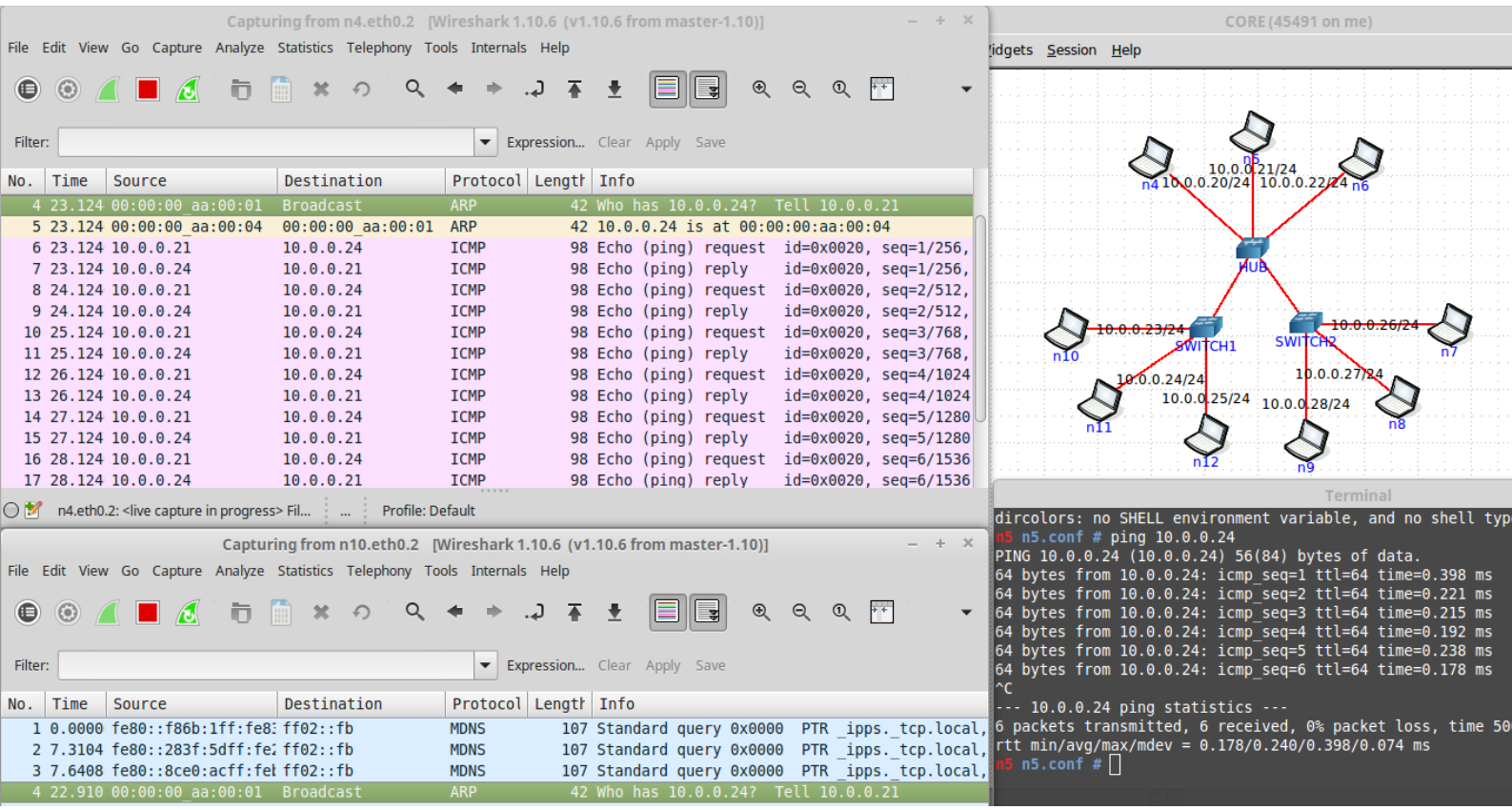


Figura 5: Comportamento de HUBs e SWITCHs com ping efetuado de n5 para n11 e captura em n4 e n10

3. DHCP

Enquanto que nas topologias anteriores, os endereços IPs eram estáticos e atribuidos manualmente, neste capítulo temos como objectivo utilizar o protocolo DHCP (*Dynamic Host Configuration Protocol*), o que irá permitir que um servidor atribua, automaticamente, endereço IPs únicos, dentro de uma gama configurada no mesmo, e os liberte e renove consoante a remoção ou adição de sistemas á rede local.

A utilização deste protocolo traz enúmeras vantagens, como evitar erros de configuração originado pela necessidade de introduzir valores manualmente em cada sistema, prevenção de conflitos de endereços que já estejam em utilização por outro sistema e uma diminuição do tempo dispendido na configuração e reconfiguração individual dos sistemas.

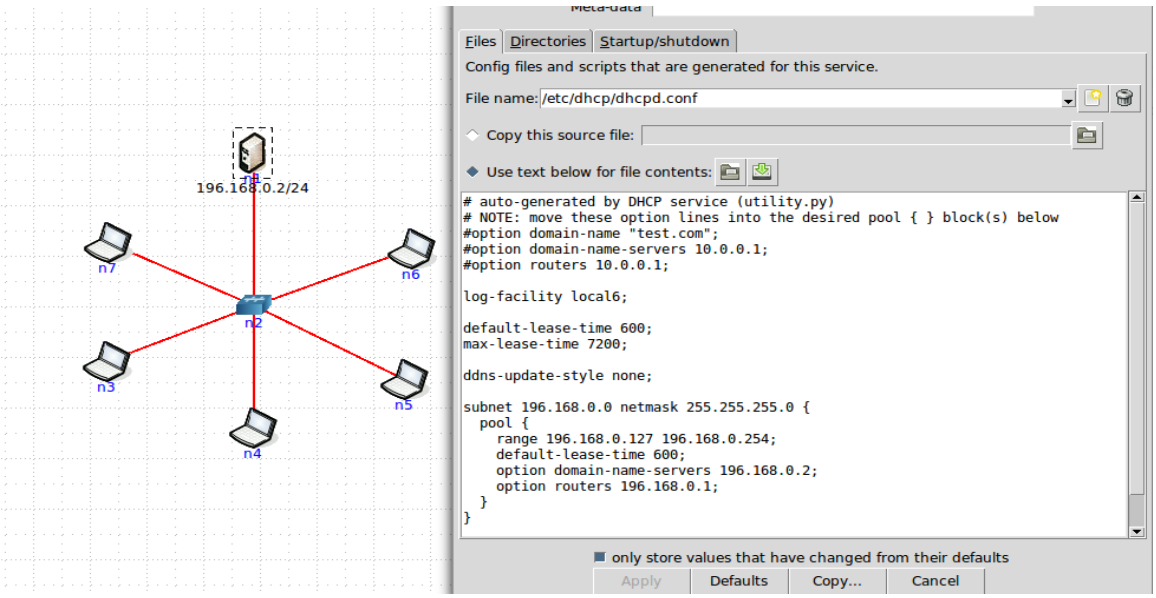


Figura 6: Topologia D e configuração do servidor DHCP

No servidor DHCP, para além desta configuração foi também definido um tempo de atraso de 15 segundos e num cliente (n3) um atraso de 20 segundos. Isto irá permitir que seja possível capturar a conexão do cliente ao servidor no Wireshark como mostra a Figura 7.

18	12.875	0.0.0.0	255.255.255.255	DHCP	342 DHCP Discover - Transaction ID 0x296f1e21
19	12.875	00:00:00_aa:00:00	Broadcast	ARP	42 Who has 196.168.0.130? Tell 196.168.0.2
20	13.874	00:00:00_aa:00:00	Broadcast	ARP	42 Who has 196.168.0.130? Tell 196.168.0.2
21	13.877	196.168.0.2	196.168.0.130	DHCP	342 DHCP Offer - Transaction ID 0x296f1e21
22	13.877	0.0.0.0	255.255.255.255	DHCP	342 DHCP Request - Transaction ID 0x296f1e21
23	13.938	196.168.0.2	196.168.0.130	DHCP	342 DHCP ACK - Transaction ID 0x296f1e21
24	14.874	00:00:00_aa:00:00	Broadcast	ARP	42 Who has 196.168.0.130? Tell 196.168.0.2
25	14.874	00:00:00_aa:00:02	00:00:00_aa:00:00	ARP	42 196.168.0.130 is at 00:00:00:aa:00:02
26	14.874	196.168.0.2	196.168.0.130	ICMP	62 Echo (ping) request id=0xcaa8, seq=0/0, ttl=
27	14.874	196.168.0.130	196.168.0.2	ICMP	62 Echo (ping) reply id=0xcaa8, seq=0/0, ttl=
28	19.882	00:00:00_aa:00:02	00:00:00_aa:00:00	ARP	42 Who has 196.168.0.2? Tell 196.168.0.130
29	19.882	00:00:00_aa:00:00	00:00:00_aa:00:02	ARP	42 196.168.0.2 is at 00:00:00:aa:00:00

Figura 7: Conexão entre servidor e cliente DHCP

Após o tempo de atraso expirar, o cliente DHCP envia, em broadcast, um pedido “DHCP Discover” de modo a obter uma resposta do servidor DHCP. O servidor, por sua vez, baseado na disponibilidade e configurações definidas nele, determina, reserva e retorna um endereço IP na forma de “DHCP Offer”, ficando a espera de uma confirmação de uso do endereço IP por parte do cliente, “DHCP Request”e reponde mais uma vez com um “DHCP ACK” que irá informar ao cliente o tempo de concessão daquele endereço IP. No final, são actualizadas as tabelas de ambos com a respectiva correlação entre o endereço IP e endereço MAC recorrendo aos protocolos estudados anteriormente (ARP e ICMP).

4. Interligação de redes

Neste exercício pretende-se emular uma topologia que contenha as anteriores, sendo que estas deverão estar ligadas a um *router* apenas e, estes ligados entre si com caminhos alternativos. O *router* permite que seja feita a comunicação entre os dispositivos das diferentes redes (e de redes com diferentes tecnologias) tendo a capacidade de fazer chegar os pacotes de uma rede de origem a uma rede de destino.

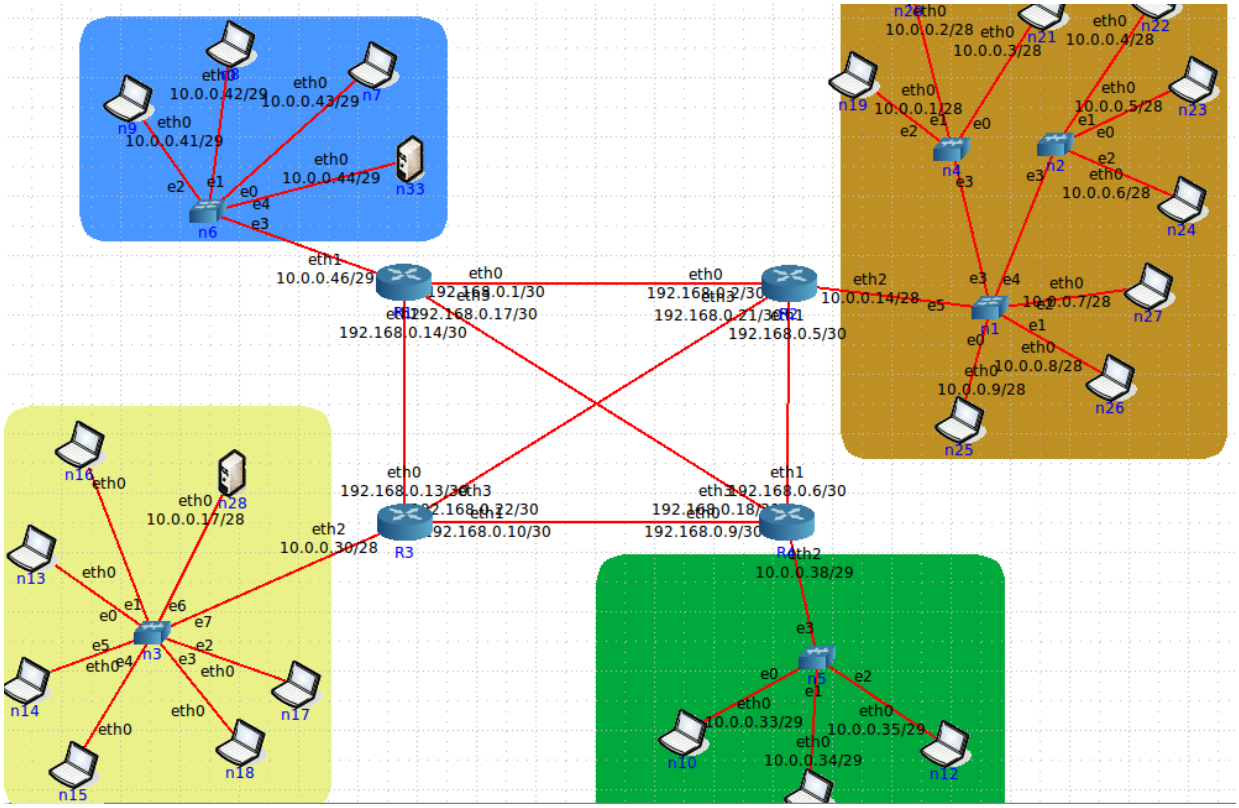


Figura 8: Topologia E

Como ilustrado na Figura 8, e referenciado no enunciado, era necessário as redes das topologias anteriores estarem na gama de endereços 10.0.0.0/24 e os *routers* interligados por uma sub-rede com máscara de 30 *bits* da rede 192.168.0.0/24, assim como desativar o encaminhamento dinâmico em todos os *routers* de modo a garantir a conectividade IPv4 entre todas as redes de acordo com o esquema concebido para encaminhamento dos mesmos. Para tal era necessário, em modo execução, aceder a *shell window* de vtysh de cada router e efectuar o comando "show running-config" e de seguida "ip route #rededestino #próximonó" como exemplificado na Figura 9. Cada router terá, necessáriamente, três caminhos sendo que ele já está conectado a uma rede local. Para confirmar se tudo foi introduzido correctamente executa-se, ainda na *shell window* o comando "show running-config".

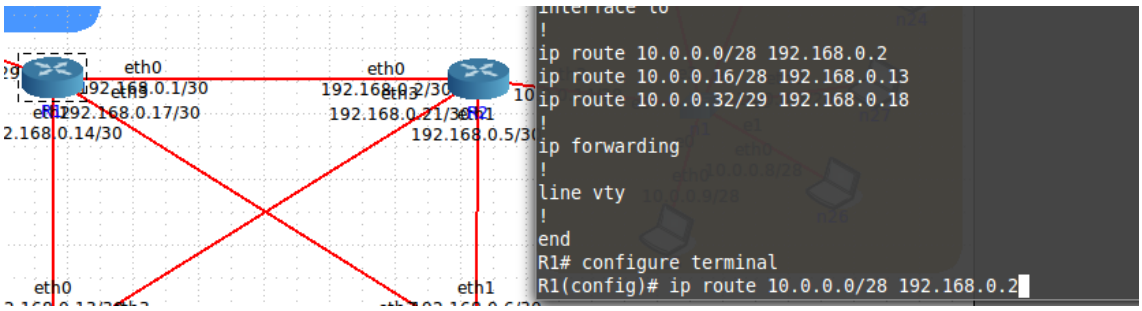


Figura 9: Configuração manual de um router

Terminado o processo de configuração, é altura de testar a conectividade entre todas as redes com os comandos “ping” e “traceroute”. Por simplificação na apresentação dos resultados, irá ser efetuado os comandos desde a topologia A (a azul) e as restantes três.

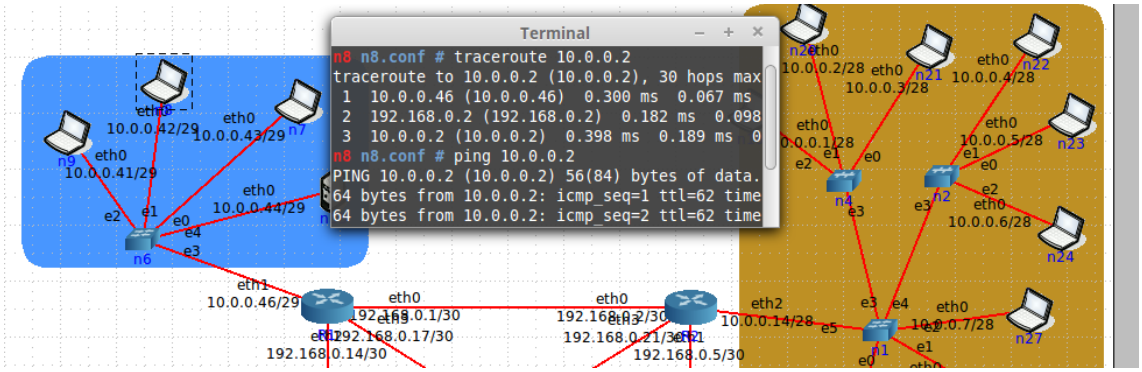


Figura 10: Ping e Traceroute entre topologia A e topologia C

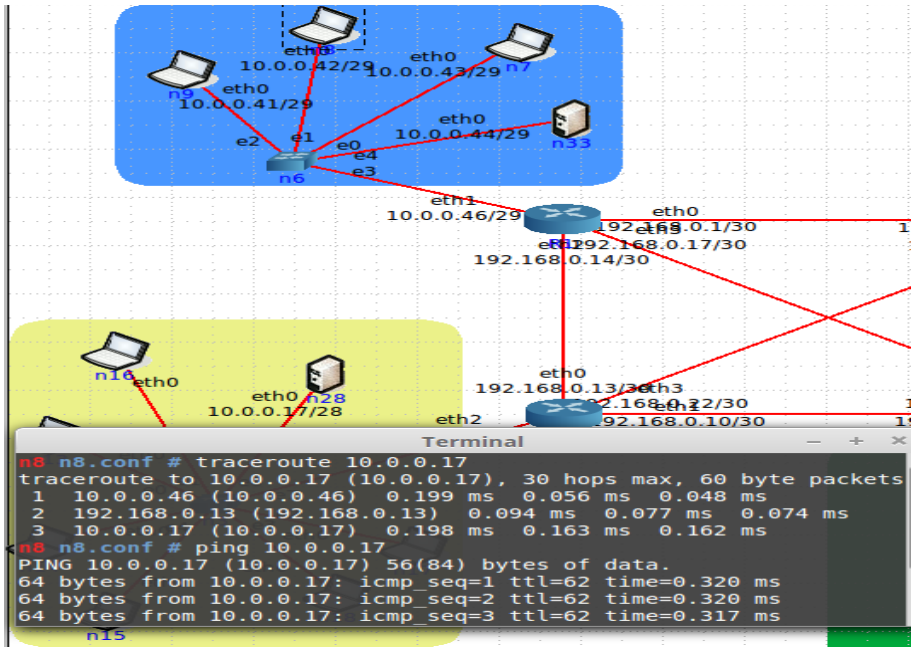


Figura 11: Ping e Traceroute entre topologia A e topologia D

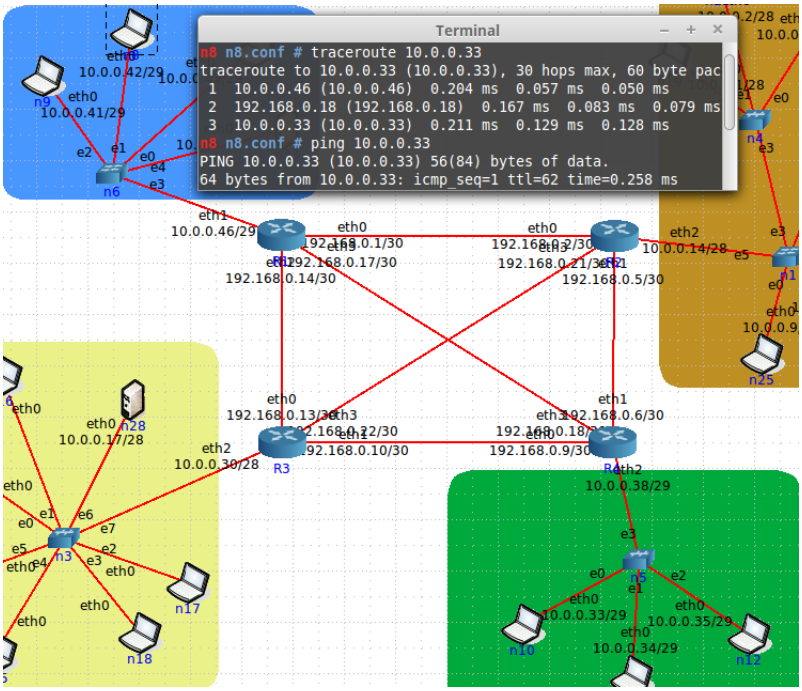


Figura 12: Ping e Traceroute entre topologia A e topologia B

Nas três figuras anteriores, é possível verificar o caminho efetuado pelo pacotes desde a topologia A até ás outras três, assim como verificar o bom funcionamento com o comando “ping”.

5. Uso das camadas de rede e transporte por parte de aplicações

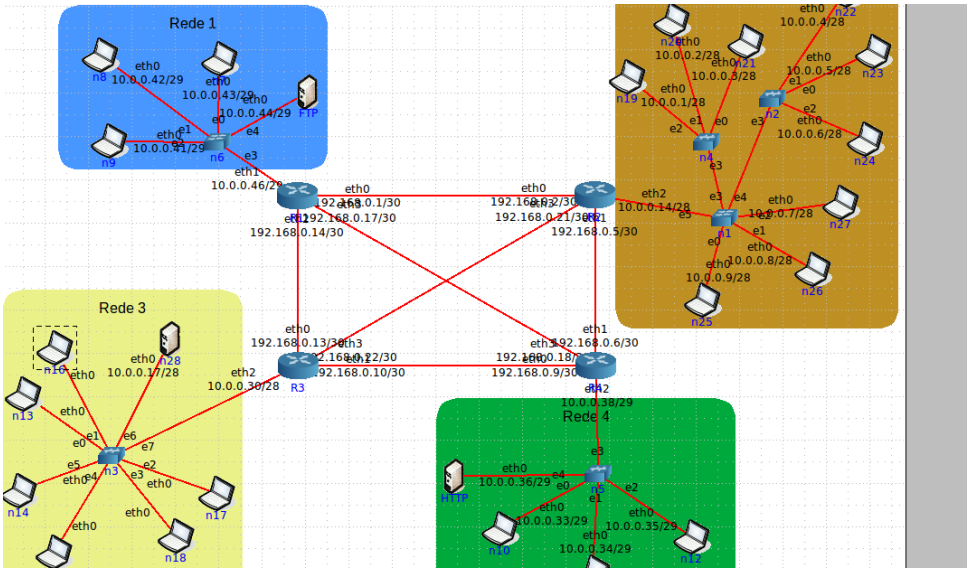


Figura 13: Topologia anterior com servidor FTP e HTTP

De modo a satisfazer o que nos é pedido neste exercício, foram instalados serviços de rede conhecidos, como um servidor FTP (vsftpd) e um servidor HTTP (mini-http) e, então, implementados em *hosts* localizados na rede 1 e rede 4, como representado na Figura 13. Após as configurações dos mesmos, é-nos pedido para, através de clientes de redes distintas, testarmos ambos os servidores, capturar pacotes e analisar os protocolos das diferentes camada da pilha TCP/IP. Ambos os servidores irão estar activos e prontos a serem utilizados uma vez que foram definidos comandos de inicio de modo a serem executados assim que forem ligados.

Para o servidor FTP, os comandos de inicio que foram definidos são: “chmod a-w /var/run/vsftpd/empty”, “chmod a-w /var/ftp” e “vsftpd ./vsftpd.conf”. Para o servidor HTTP basta o comando “mini-httpd”.

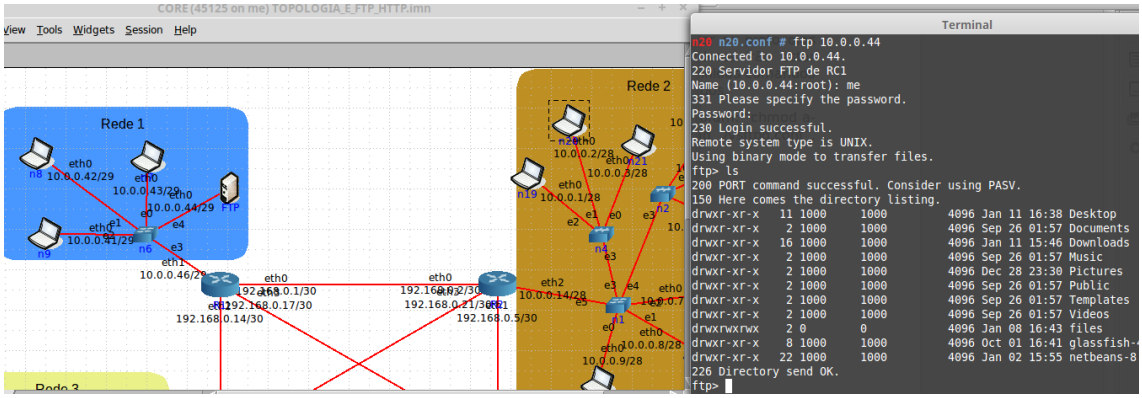


Figura 14: Conexão ao servidor FTP por um sistema da rede 2

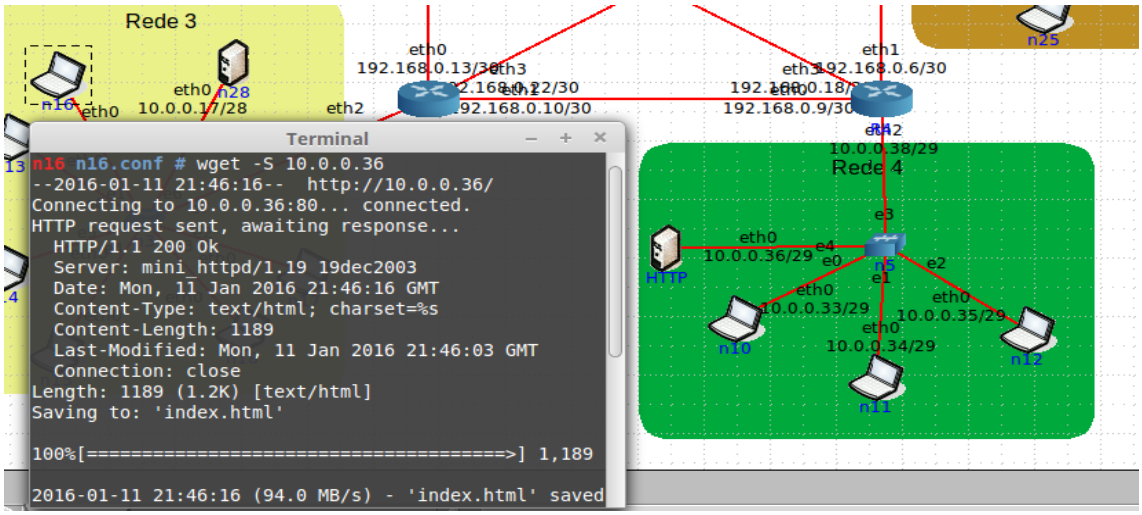
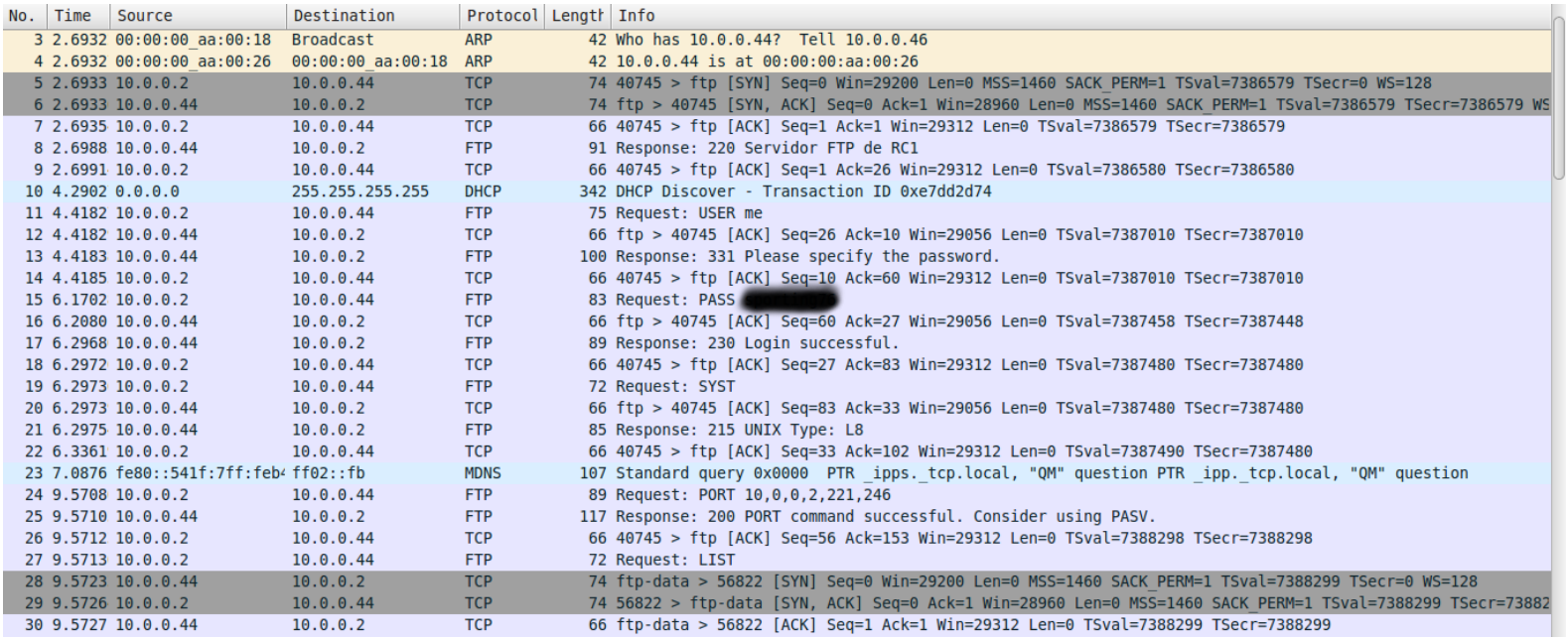


Figura 15: Conexão ao servidor HTTP por um sistema da rede 3

Nas Figuras 14 e 15, após a rede ter iniciada em modo de execução no CORE, verifica-se que ambos os servidores já estão activos e prontos a receber conexões, sendo que na Figura 14 e conforme o enunciado, o sistema n20 da Rede 2 (rede local distinta do servidor FTP) efectua o comando “ftp 10.0.0.44” e, prontamente, recebe a mensagem de boas-vindas pedindo, também, para efectuar o *login* no servidor. Após a introdução dos dados correctamente, efectua o comando “ls” com o intuito

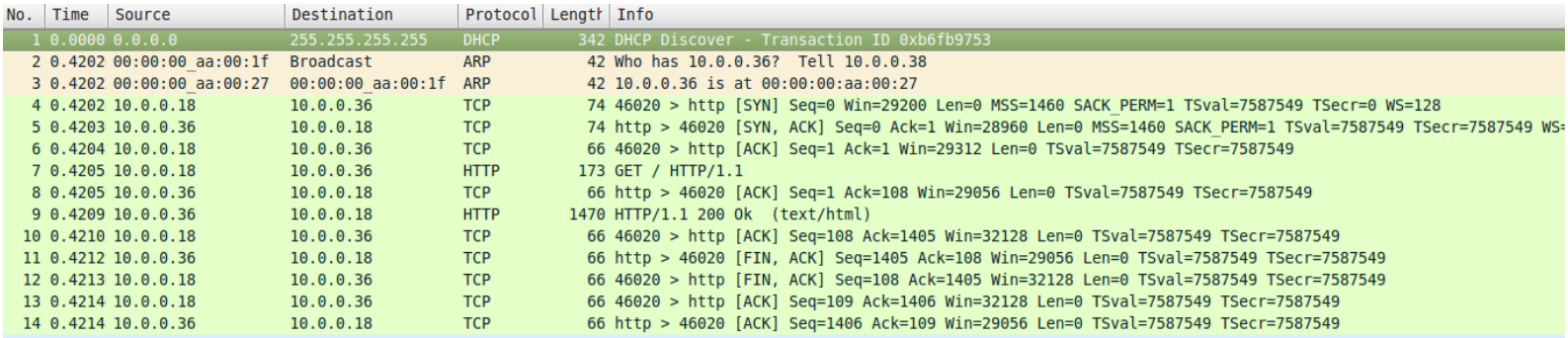
de listar os ficheiros da pasta local do servidor, como se pode verificar. Na Figura 15, o sistema n16 da rede 3 (rede que utiliza o protocolo DHCP) conecta com sucesso ao servidor HTTP através do comando “wget -S 10.0.0.36”, o qual imprime no ecrã o cabeçalho enviado pelo servidor HTTP.



No.	Time	Source	Destination	Protocol	Length	Info
3	2.6932	00:00:00_aa:00:18	Broadcast	ARP	42	Who has 10.0.0.44? Tell 10.0.0.46
4	2.6932	00:00:00_aa:00:26	00:00:00_aa:00:18	ARP	42	10.0.0.44 is at 00:00:00:aa:00:26
5	2.6933	10.0.0.2	10.0.0.44	TCP	74	40745 > ftp [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=7386579 TSecr=0 WS=128
6	2.6933	10.0.0.44	10.0.0.2	TCP	74	ftp > 40745 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=7386579 TSecr=7386579 WS
7	2.6935	10.0.0.2	10.0.0.44	TCP	66	40745 > ftp [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=7386579 TSecr=7386579
8	2.6988	10.0.0.44	10.0.0.2	FTP	91	Response: 220 Servidor FTP de RC1
9	2.6991	10.0.0.2	10.0.0.44	TCP	66	40745 > ftp [ACK] Seq=1 Ack=26 Win=29312 Len=0 TSval=7386580 TSecr=7386580
10	4.2902	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xe7dd2d74
11	4.4182	10.0.0.2	10.0.0.44	FTP	75	Request: USER me
12	4.4182	10.0.0.44	10.0.0.2	TCP	66	ftp > 40745 [ACK] Seq=26 Ack=10 Win=29056 Len=0 TSval=7387010 TSecr=7387010
13	4.4183	10.0.0.44	10.0.0.2	FTP	100	Response: 331 Please specify the password.
14	4.4185	10.0.0.2	10.0.0.44	TCP	66	40745 > ftp [ACK] Seq=10 Ack=60 Win=29312 Len=0 TSval=7387010 TSecr=7387010
15	6.1702	10.0.0.2	10.0.0.44	FTP	83	Request: PASS [redacted]
16	6.2080	10.0.0.44	10.0.0.2	TCP	66	ftp > 40745 [ACK] Seq=60 Ack=27 Win=29056 Len=0 TSval=7387458 TSecr=7387448
17	6.2968	10.0.0.44	10.0.0.2	FTP	89	Response: 230 Login successful.
18	6.2972	10.0.0.2	10.0.0.44	TCP	66	40745 > ftp [ACK] Seq=27 Ack=83 Win=29312 Len=0 TSval=7387480 TSecr=7387480
19	6.2973	10.0.0.2	10.0.0.44	FTP	72	Request: SYST
20	6.2973	10.0.0.44	10.0.0.2	TCP	66	ftp > 40745 [ACK] Seq=83 Ack=33 Win=29056 Len=0 TSval=7387480 TSecr=7387480
21	6.2975	10.0.0.44	10.0.0.2	FTP	85	Response: 215 UNIX Type: L8
22	6.3361	10.0.0.2	10.0.0.44	TCP	66	40745 > ftp [ACK] Seq=33 Ack=102 Win=29312 Len=0 TSval=7387490 TSecr=7387480
23	7.0876	fe80::541f:7ff:feb4	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR _ipp._tcp.local, "QM" question
24	9.5708	10.0.0.2	10.0.0.44	FTP	89	Request: PORT 10,0,0,2,221,246
25	9.5710	10.0.0.44	10.0.0.2	FTP	117	Response: 200 PORT command successful. Consider using PASV.
26	9.5712	10.0.0.2	10.0.0.44	TCP	66	40745 > ftp [ACK] Seq=56 Ack=153 Win=29312 Len=0 TSval=7388298 TSecr=7388298
27	9.5713	10.0.0.2	10.0.0.44	FTP	72	Request: LIST
28	9.5723	10.0.0.44	10.0.0.2	TCP	74	ftp-data > 56822 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=7388299 TSecr=0 WS=128
29	9.5726	10.0.0.2	10.0.0.44	TCP	74	56822 > ftp-data [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=7388299 TSecr=73882
30	9.5727	10.0.0.44	10.0.0.2	TCP	66	ftp-data > 56822 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=7388299 TSecr=7388299

Figura 16: Captura dos pacotes no servidor FTP

Na Figura 16 estão apresentados os pacotes, recebidos e enviados, pelo servidor FTP no seguinte encadeamente de comandos: efectua-se a tentativa de conexão com o comando “ftp 10.0.0.44”, de seguida é apresentado o menu de boas-vindas assim como o de *login* por parte do sistema que acede ao servidor e, por fim, este executa o comando “ls” no servidor.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xb6fb9753
2	0.4202	00:00:00_aa:00:1f	Broadcast	ARP	42	Who has 10.0.0.36? Tell 10.0.0.38
3	0.4202	00:00:00_aa:00:27	00:00:00_aa:00:1f	ARP	42	10.0.0.36 is at 00:00:00:aa:00:27
4	0.4202	10.0.0.18	10.0.0.36	TCP	74	46020 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=7587549 TSecr=0 WS=128
5	0.4203	10.0.0.36	10.0.0.18	TCP	74	http > 46020 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=7587549 TSecr=7587549 WS=
6	0.4204	10.0.0.18	10.0.0.36	TCP	66	46020 > http [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=7587549 TSecr=7587549
7	0.4205	10.0.0.18	10.0.0.36	HTTP	173	GET / HTTP/1.1
8	0.4205	10.0.0.36	10.0.0.18	TCP	66	http > 46020 [ACK] Seq=1 Ack=108 Win=29056 Len=0 TSval=7587549 TSecr=7587549
9	0.4209	10.0.0.36	10.0.0.18	HTTP	1470	HTTP/1.1 200 Ok (text/html)
10	0.4210	10.0.0.18	10.0.0.36	TCP	66	46020 > http [ACK] Seq=108 Ack=1405 Win=32128 Len=0 TSval=7587549 TSecr=7587549
11	0.4212	10.0.0.36	10.0.0.18	TCP	66	http > 46020 [FIN, ACK] Seq=1405 Ack=108 Win=29056 Len=0 TSval=7587549 TSecr=7587549
12	0.4213	10.0.0.18	10.0.0.36	TCP	66	46020 > http [FIN, ACK] Seq=108 Ack=1405 Win=32128 Len=0 TSval=7587549 TSecr=7587549
13	0.4214	10.0.0.18	10.0.0.36	TCP	66	46020 > http [ACK] Seq=109 Ack=1406 Win=32128 Len=0 TSval=7587549 TSecr=7587549
14	0.4214	10.0.0.36	10.0.0.18	TCP	66	http > 46020 [ACK] Seq=1406 Ack=109 Win=29056 Len=0 TSval=7587549 TSecr=7587549

Figura 17: Captura dos pacotes no servidor HTTP

Na Figura 16 estão apresentados os pacotes, recebidos e enviados, pelos servidor HTTP quando um sistema executa o comando “wget -S 10.0.0.36” com o objectivo de vizualizar o cabeçalho dos pacotes enviados pelo servidor HTTP.

6. Interligação via NAT (*Network Address Translator*)

Neste exercicio é acrescentada uma rede local, utilizando endereços privados na gama 192.168.1.0/24, e ligada a uma das redes existente na topologia anterior através de um *router* NAT. Para obter este tipo de *router* será utilizado o programa iptables instalado na distribuição de Linux e respectiva configuração no *router*.

Este *router* NAT irá ser projetado com o intuito de conservar os endereços IP da rede privada. Isto significa que será anunciado, a qualquer outra rede, apenas um endereço IP, o qual representa toda a sua rede privada, funcionando como um agente entre a rede privada e as restantes.

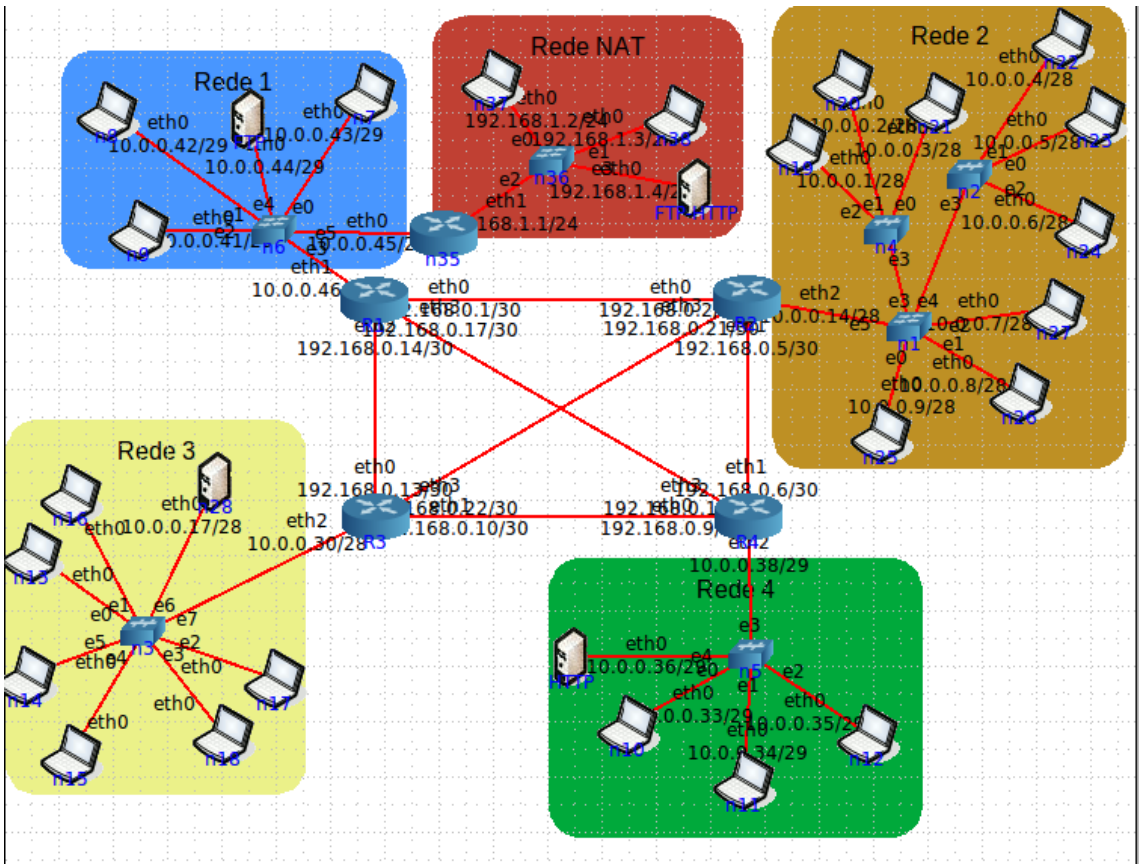


Figura 18: Topologia com NAT

Na Figura 18 está representada a topologia anterior com uma nova rede privada (a vermelho) ligada a uma rede local já existente através do *router* NAT (n35).

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -A FORWARD -i eth1 -j ACCEPT
```

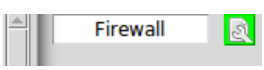


Figura 19: Configuração do router NAT na secção Firewall

Efectuada, novamente, a configuração dos “ip routes” nos *routers* e na Rede 1, é possível capturar os pacotes enviados, de um sistema exterior a rede privada, sendo que a comunicação é efectuada através do endereço IP do *router* NAT, mesmo que o comando “*ping*” seja efectuado para um sistema interno da rede, como ilustrado na Figura 19. O sistema terminal onde é executado o comando “*ping*” está situado na Rede 1.

Figura 20: Captura dos pacotes no sistema onde é efetuado o comando "ping"

Para terminar, era pedido para se adicionar um servidor FTP e HTTP á rede privada (já incluído na Figura 18) e testar o seu funcionamento (Figura 21).

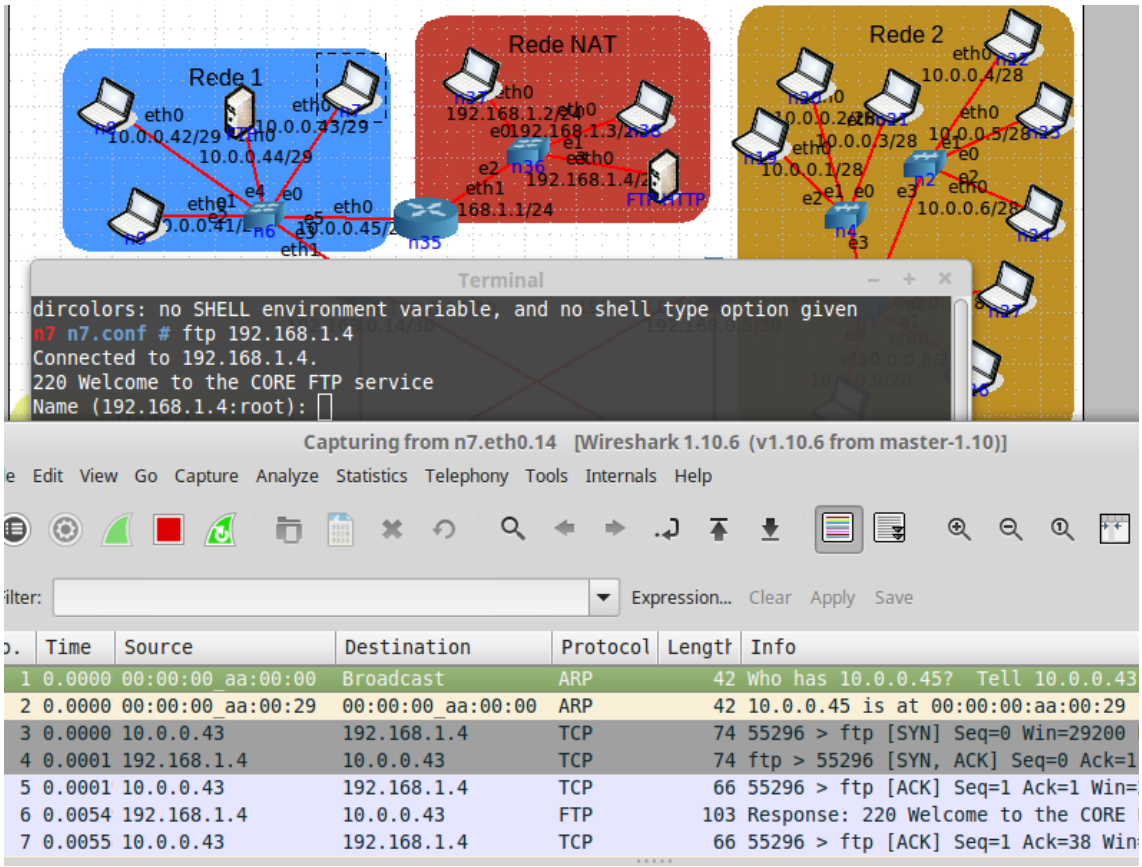


Figura 21: Conexão ao servidor FTP na rede privada por um sistema exterior

7. Conclusão

Neste trabalho prático da UC de Redes de Computadores I, foi nos proposto implementar várias topologias de Rede, com o objetivo de verificar o funcionamento de diversos sistemas e alguns dos protocolos de encaminhamento que foram lecionados.

Durante a elaboração deste projeto, deparamo-nos com algumas dificuldades, sendo estas maioritariamente devido a falta de conhecimentos de comandos e configurações necessárias dos sistemas pedidos. Um exemplo de falta de conhecimento na configuração de Apache para servidores HTTP, tendo sido necessário a implementação de outro semelhante, mini-httpd.

Contudo, sentimo-nos bastante satisfeitos com as resoluções apresentadas dos exercícios propostos no enunciado do projeto pois, permitiu consolidar a componente teórica da UC, assim como aprender a configurar máquinas Linux e trabalhar num ambiente de rede simulado, mas ao mesmo tempo bastante perto da realidade, sendo possível visualizar os comportamentos das máquinas quando sujeitos a diferentes protocolos.