

**TP2:**

**Desenvolvimento de Aplicação e Configuração de Rede Local**

Redes de Computadores 2014/2015

**Hugo Manuel Carvalho Fonseca 201109210**

**Nuno Diogo Gonçalves Martins 201105567**

**Nuno Figueiredo Pires 200907657**

**Turma 13**

**Sumário**

O relatório é constituído por uma análise detalhada quer à implementação da aplicação de *download*, quer às configurações da rede local. Desse modo, serão descritas a arquitetura da aplicação, bem como as análises às configurações feitas ao longo das várias experiências, sendo, sendo que necessário, auxiliadas de *logs* retirados e/ou comandos efetuados.

Após a descrição dos resultados obtidos, procuraremos retirar conclusões acerca dos conceitos envolvidos no projeto e que foram lecionados na unidade curricular durante o semestre.

**Introdução**

Este projeto foi desenvolvido no âmbito da unidade curricular de Redes de Computadores, lecionado no 4.º ano do MIEEC de Engenharia Eletrotécnica e de Computadores (MIEEC), ramo de Telecomunicações, Eletrónica e Computadores, na Faculdade de Engenharia da Universidade do Porto e tem como objetivos principais a implementação:

1. Desenvolvimento de uma aplicação, responsável pelo *download* de um único ficheiro em cada execução;
2. Configuração e estudo de uma rede local de computadores.

Deste modo, a aplicação irá consistir, numa primeira fase, no *download* de um ficheiro de um servidor FTP, através da implementação de um protocolo FTP auxiliado de *sockets*. Um *socket* será utilizado para comunicar ao servidor o pedido de entrada em modo passivo, bem como a inserção das credenciais respetivas. Após entrar em modo passivo, o servidor enviará 6 *bytes*, contendo o endereço IP do servidor e a porta à qual se deverá conectar, sendo criado, para o efeito, um novo *socket*. Este *socket* será responsável pela receção de dados do ficheiro escolhido.

Por último, será configurada uma rede local de computadores passo-a-passo, i.e., configuração de uma rede de IPs, criação e configuração de LANs virtuais, configurações de um *router* e de *router* comercial com implementação de NAT e, por último, configuração do DNS. Só após estes passos, separados em várias experiências, será possível testarmos a nossa aplicação na rede de computadores configurada.

Em relação à estrutura do relatório, esta encontra-se dividida em seis secções:

* A (presente) introdução, a qual faz uma breve introdução ao conteúdo e objetivo trabalho descrito;
* Descrição da arquitetura da aplicação, bem como exemplificação do seu bom funcionamento através da exposição de um caso de utilização;
* Descrição da configuração da rede local e análise dos resultados obtidos, bem como dos objetivos de cada experiência, auxiliada pelos *logs* obtidos (e incluídos em anexo);
* Apresentação dos diagramas de casos de uso, onde é indicada a sequência de chamadas das funções;
* Conclusões obtidos após finalização do projeto, onde será feito um resumo do que foi tratado e salientados os pontos fulcrais do projeto;
* Inclusão de anexos importantes, nomeadamente, o código responsável pelo *download* do ficheiro, assim como os comandos usados para configurar a rede e os *logs* obtidos.

**Parte 1 – Desenvolvimento de aplicação de *download* de ficheiro**

**Parte 2 – Configuração e análise da rede**

* 1. Experiência 1: configuração de rede IP

Esta experiência teve como objetivo a configuração de uma rede IP que, neste caso, foi estabelecida entre dois computadores através das suas portas série, auxiliado de um cabo série.

Inicialmente, começou-se por desligar o cabo que liga o *switch* ao *router* do laboratório, algo que resultou na perda da ligação à internet em ambos os computadores. Deste modo, pretendeu-se entender o funcionamento dos protocolos ARP e ICMP quando se recebe/envia um *ping*. Após isso, configurou-se os IPs de cada tux pelos valores especificados, através dos comandos especificados no Anexo A (cf. [B.1.1.](#comandos_EXP1)).

O protocolo ARP (*Address Resolution Protocol*) é usado quando um computador, conectado a uma rede, pretende enviar uma trama *ethernet*, sendo conhecedor do endereço IP do computador a que se destina. Desse modo, antes de enviar o trama *ethernet*, o computador procura saber qual é o endereço MAC correspondente ao IP que se pretende. Tal é conseguido através do envio de uma ARP *Request*, em *broadcast*, obtendo essa informação. Cada computador que se encontra conectado à rede recebe essa mensagem e, caso verifique que essa mensagem lhe é destinada, retornará a informação pretendida, igualmente em *broadcast*. Após isso, o computador que pretende as informações irá atualizar a sua ARP *table* – tabela que contém os endereços MAC para IPs já conhecidos -, enviando, posteriormente, a trama *ethernet* com as informações que já contém.

Deste modo, cada trama ARP contém quer o IP de origem, quer o de destino, identificando assim o caminho que se pretende fazer. No entanto, é também identificado o endereço MAC do computador no qual partiu a trama pela última vez, assim como o endereço MAC para onde esta se dirige. Tal deve-se devido ao facto de, em cada momento, a trama ARP poder situar-se num qualquer sítio aleatório da rede e é necessário saber o último endereço MAC para poder enviar a resposta e o endereço MAC de destino onde continuar a procurar (sendo que este é, inicialmente, o de *broadcast*).

Quando enviamos um comando *ping*, no caso deste ainda não ter na sua tabela ARP a identificação de qual computador possui determinado IP, envia antes uma trama ARP, por forma a obter esta informação (tal como foi anteriormente descrito). Após isso, será enviado uma trama ICMP (tipo 8, que corresponde a um “*echo* *request*”) para o computador respetivo. Em consequência disso, o computador que recebeu a trama enviará contrariamente uma trama ICMP (tipo 0, correspondente a um “*echo reply*”). Note-se que os endereços MAC e IP destas duas tramas estão em conformidade com o que foi descrito no parágrafo anterior.

É também possível determinar qual o tipo de trama *Ethernet*. Para tal, basta observar os 2 *bytes* correspondentes ao *header*. No caso de conter o valor 0x0800, corresponde ao protocolo ARP, ao invés do valor 0x0806, que corresponde ao protocolo IPv4. Para além destes, no caso de se tratar de um protocolo IP, existe no *header* a informação sobre o protocolo correspondente e que, no caso do ICMP, conterá o valor 1. Para saber o tamanho de uma trama, quer o programa *Wireshark*, quer o pacote IP contêm esta informação: através do pacote IP, existe um parâmetro que refere o tamanho, enquanto, no *Wireshark*, facilmente obtemos esta informação ao observar os *logs* obtidos (cada trama descrita possui uma coluna com o tamanho respetivo).

Por último, uma interface *loopback* é uma interface virtual que não está associada a qualquer componente física num computador, sendo habitualmente usada para testes. Assim, tal como seu nome indica, qualquer mensagem transmitida para esta interface é imediatamente recebida pelo mesmo. De notar que, por definição, esta interface deve estar sempre ativa e que, quando é estabelecida uma nova vizinhança através desta, apenas haverá perda de vizinhança quando as interfaces físicas ficarem inativas.

* 1. Experiência 2: implementação de duas LANs virtuais através do *switch*

Esta experiência teve como objetivo a criação e implementação de duas LANs virtuais (VLAN) através do *switch*, podendo as portas serem escolhidas de forma personalizada.

Para se criar e configurar uma VLAN, foram necessários determinados comandos executados no ambiente do *GTKTerm* e que poderão ser observados em anexo (cf. anexo [B.2.2.](#comandos_EXP2_2)). No entanto, foram ainda necessários outros comandos para tornar as VLANs criadas funcionais, nomeadamente a permissão a cada computador para responder a *pings broadcast*, bem como configurar quais os IPs correspondentes (cf. o comando *echo* presente no anexo [B.2.1.](#comandos_EXP2_1)) e as rotas que cada computador possuirá.

Nesta experiência, existem dois domínios *broadcast*, sendo tal comprovado pelos *logs* obtidos e que se encontram em anexo (cf. anexos [B.2.](#log_EXP2)). Os *logs* foram obtidos na sequência de um *ping broadcast* enviado pelo tux21. A partir deles, pode-se concluir que o tux22 não recebe os pacotes ICMP, ao invés do tux24, que recebe. No entanto, este último apenas não responde a eles pois a opção para resposta a *pings broadcast* está desativada. Nos anexos são ainda incluídos os *logs* obtidos na sequência de um *ping broadcast* enviado pelo tux22 e, tal como se pode verificar, o tux22 é o único que recebe os pacotes do *ping*.

* 1. Experiência 3: Configuração de um *router*

Esta experiência teve por base a configuração de um computador como *router* (neste caso, do tux24), por forma a conseguir conectar as duas VLANs criadas na experiência anterior.

Deste modo, é necessário que os tux21 e tux22 conheçam um caminho pelo qual pode enviar o pacote quando o computador destinado não se encontra na rede. Configurando o tux24 como *router* – colocando cada interface deste em cada uma das VLANs criadas – e definindo-o como *gateway* dos tux21 e tux22 (cf. anexo [B.3.1.](#comandos_EXP3_1)), é obtido o pretendido. Esta configuração é descrita pelas tabelas de reencaminhamento em cada tux, a qual contém as rotas com os destinos para os quais se devem enviar pacotes, sabendo o IP de destino. As tabelas de encaminhamento caraterizam as rotas em várias componentes: IP de destino, *gateway* – IP para o qual se devem enviar os pacotes, caso não exista nenhuma rota para o IP de destino –, máscara de rede e a interface na qual se envia o pacote (existem outros campos, mas só estes interessam para a nossa análise).

Por este motivo, é justo dizer que o tux24 assume um papel vital na nossa rede, pois é ele que reencaminha os pacotes para o tux respetivo. Além deste tux ser o *gateway* do tux21 e do tux22, contém também na sua tabela de endereçamento a informação de que, para a rede do lado do tux21 (i.e., a rede 172.16.20.0/24), a informação passa pela interface eth0, enquanto para a rede do lado oposto (i.e., a rede 172.16.21.0/24), a informação passa pela interface eth1.

Em relação aos *logs* observados (cf. logs do anexo [B.3.](#log_EXP3)), é possível observar o que se conclui na experiência 1 sobre os endereços IP e MAC existentes nos pacotes ARP e ICMP. Tal foi derivado por um *ping* enviado do tux21 para o tux22, observado a partir do tux24, e que se concluiu que, para enviar esse *ping* para o tux22, o pacote ICMP respetivo que passa na interface eth0 contém o endereço MAC de destino do tux24 (do lado da interface eth0) e não o do tux22. Na situação contrária, isto é, do envio da resposta ao *ping* pelo tux22, o pacote ICMP contém igualmente o endereço MAC de origem do tux24 (eth0) e não o do tux22. Em relação aos pacotes ICMP que passam pela interface eth1, a situação é idêntica, contudo, os endereços MAC de origem e destino serão o do tux24, conforme seja receção ou resposta ao *ping*.

* 1. Experiência 4: configuração de *router* comercial e implementação do NAT

Esta experiência teve como objetivos a configuração de um *router* comercial para que, posteriormente, possa ser implementado o NAT. Além disso, as restantes configurações seguem o que foi feito na experiência anterior (nomeadamente, as VLANs existentes e a atuação do tux24 como *router*).

A configuração do *router* comercial começa por atribuir as interfaces para a rede configurada, tendo conta as ligações efetuadas o router e o *switch*, adicionando posteriormente as rotas que possibilitem a correta comunicação entre todos os tuxs. Os comandos relativas à configuração do *router* comercial pode ser observada em anexo (cf. anexo [A.4.3.](#comandos_EXP4_3)), destacam-se que, para definir uma rota estática, será necessário o seguinte comando:

*ip route* <IP\_DESTINO> <MÁSCARA\_SUBREDE> <GATEWAY>

Após isso, o tux24 (eth0) foi definido como *default gateway* do tux21. Já o *router* comercial foi definido como *default gateway* do tux22 e do tux24 (eth1). Por forma a mostrar a eficácia destas configurações, foi efetuado um *ping* a partir do tux21 para as interfaces do tux24 e para o tux22 e o *router* comercial.

Como era esperado, após esta configuração, os pacotes enviados pelo tux21 vão para o tux24 e, a partir deste, redirecionados para o *router* comercial ou para o tux22, conforme o que se especificou.

Por fim, e por forma a obter resposta às tentativas de contacto à rede exterior, procedeu-se à configuração do NAT, para que os terminais exteriores possuem uma forma de saber a qual endereço devem responder, uma vez que todos os tux’s são vistos como possuindo o mesmo endereço. Configurando o NAT, todos os endereços da rede são mapeados para endereços públicos e, desta forma, a resposta é concretizável, pois a rede exterior já sabe identificar cada tux separadamente. Os comandos responsáveis pela configuração do NAT podem ser consultados nos anexos (cf. anexo [A.4.4.](#comandos_EXP4_4)).

* 1. Experiência 5: DNS

Esta experiência teve como objetivo a possibilidade de a rede criada possuir serviços de DNS (*Domain Name System*). Deste modo, foi permitida a utilização direta do *hostname* ao invés de apenas ser possível o uso do IP pretendido.

Para um *host* poder ter serviços de DNS basta que tenha, no mínimo, uma tabela na qual, para cada *hostname*, seja identificado o IP a que corresponde. Assim, na experiência feita, bastou apenas identificar no computador a localização de um servidor DNS pré-existente (neste caso, o servidor *lixa.netlab.fe.up.pt*, com IP 172.16.2.1), contendo já estas tabelas. Para o computador puder ter acesso a estas informações, bastou editar o ficheiro *resolve.conf*, existente na pasta */etc*, colocando a informação seguinte:

*search netlab.fe.up.pt*

*nameserver 172.16.2.1*

Podemos observar pelos Anexos [B.5.](#log_EXP5) que, quando executado um *ping* para www.facebook.com, é enviado um pacote DNS para o servidor DNS configurado, requerendo o IP do *site* “pingado”. Após este pedido, é recebido um outro pacote DNS enviado pelo servidor com a informação pretendida e, após isso, o primeiro *ping* é enviado. Posteriormente, antes de cada pacote enviado, é visualizado o mecanismo reverso, que perante um IP conhecido, o servidor DNS informa sobre qual a máquina correspondente.

**Conclusão**

**Anexo A – Comandos para as configurações**

1. Experiência 1
   1. Configuração dos IPs

tux21>> *ifconfig eth0 172.16.20.1/24*

tux24>> *ifconfig eth0 172.16.20.254/24*

1. Experiência 2
   1. Configuração dos IPs

tux21>> *ifconfig eth0 172.16.20.1/24*

tux24>> *ifconfig eth0 172.16.20.254/24*

tux22>> *ifconfig eth1 172.16.21.1/24*

tux24>> *echo 0 > /proc/sys/net/ipv4/icmp\_echo\_ignore\_broadcasts*

* 1. Criação e configuração das VLANs (*switch*)

Criação das duas VLANs

Atribuição das portas às VLANs criadas

>> configure terminal

>> vlan 20

>> vlan 21

>> interface fastethernet 0/1

>> switchport mode access

>> switchport access vlan 20

>> interface fastethernet 0/4

>> switchport mode access

>> switchport access vlan 20

>> interface fastethernet 0/2

>> switchport mode access

>> switchport access vlan 21

>> end

1. Experiência 3
   1. Configuração dos IPs

tux21>> *ifconfig eth0 172.16.20.1/24*

tux21>> *route add default gateway 172.16.20.254/24*

tux24>> *ifconfig eth0 172.16.20.254/24*

tux24>> *ifconfig eth1 172.16.21.253/24*

tux22>> *ifconfig eth1 172.16.21.1/24*

tux22>> *route add default gateway 172.16.21.253/24*

tux24>> *echo 0 > /proc/sys/net/ipv4/icmp\_echo\_ignore\_broadcasts*

tux24>> *echo 1 > /proc/sys/net/ipv4/ip\_forward*

* 1. Criação e configuração das VLANs (*switch*)

Atribuição das portas às VLANs criadas

>> *configure terminal*

>> *vlan 20*

>> *vlan 21*

>> *interface fastethernet 0/1*

>> *switchport mode access*

>> *switchport access vlan 20*

>> *interface fastethernet 0/4*

>> *switchport mode access*

>> *switchport access vlan 20*

>> *interface fastethernet 0/3*

>> *switchport mode access*

>> *switchport access vlan 21*

>> *interface fastethernet 0/2*

>> *switchport mode access*

>> *switchport access vlan 21*

>> *end*

Criação das duas VLANs

1. Experiência 4
   1. Configuração dos IPs e predefinição do tux24 e do *router* comercial como *default gateway*

tux21>> *ifconfig eth0 172.16.20.1/24*

tux21>> *route add default gateway 172.16.20.254/24*

tux24>> *ifconfig eth0 172.16.20.254/24*

tux24>> *ifconfig eth1 172.16.21.253/24*

tux24>> *route add default gateway 172.16.21.254/24*

tux22>> *ifconfig eth1 172.16.21.1/24*

tux22>> *route add default gateway 172.16.21.253/24*

tux24>> *echo 0 > /proc/sys/net/ipv4/icmp\_echo\_ignore\_broadcasts*

tux24>> *echo 1 > /proc/sys/net/ipv4/ip\_forward*

* 1. Criação e configuração das VLANs (*switch*)

Atribuição das portas às VLANs criadas

>> *configure terminal*

>> *vlan 20*

>> *vlan 21*

>> *interface fastethernet 0/1*

>> *switchport mode access*

>> *switchport access vlan 20*

>> *interface fastethernet 0/4*

>> *switchport mode access*

>> *switchport access vlan 20*

>> *interface fastethernet 0/3*

>> *switchport mode access*

>> *switchport access vlan 21*

>> *interface fastethernet 0/2*

>> *switchport mode access*

>> *switchport access vlan 21*

>> *interface fastethernet 0/5*

>> *switchport mode access*

>> *switchport access vlan 21*

>> *end*

Criação das duas VLANs

* 1. Configuração do *router* comercial

>> *configure terminal*

>> *interface fastethernet 0/0*

>> *ip address 172.16.21.254*

>> *no shutdown*

>> *exit*

>> *interface fastethernet 0/1*

>> *ip address 172.16.2.29*

>> *no shutdown*

>> *exit*

>> *ip route 0.0.0.0 0.0.0.0 172.16.2.254*

>> *ip route 172.16.20.0 255.255.255.0 172.16.21.253*

>> *end*

* 1. Implementação do NAT no *router* comercial

>> *configure terminal*

>> *interface fastethernet 0/0*

>> *ip address 172.16.21.254 255.255.255.0*

>> *no shutdown*

>> *ip nat inside*

>> *exit*

>> *interface fastethernet 0/1*

>> *ip address 172.16.2.29 255.255.255.0*

>> *no shutdown*

>> *ip nat outside*

>> *exit*

>> *ip nat pool ovrld 172.16.2.29 172.16.2.29 prefix 24*

>> *ip nat inside source list 1 pool ovrld overload*

>> *access-list 1 permit 172.16.20.0 0.0.0.7*

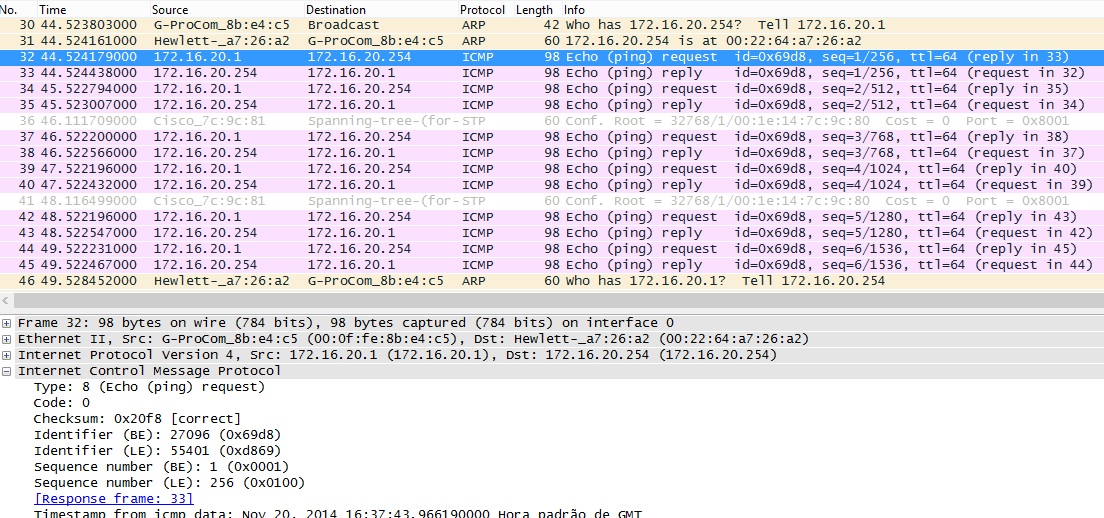
>> *access-list 1 permit 172.16.21.0 0.0.0.7*

>> *ip route 0.0.0.0 0.0.0.0 172.16.2.254*

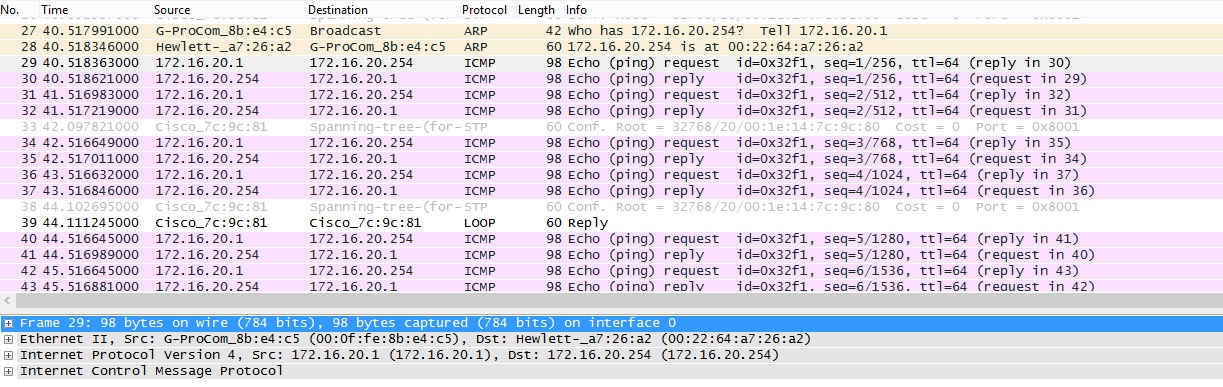
>> *ip route 172.16.20.0 255.255.255.0 172.16.21.253*

>> *end*

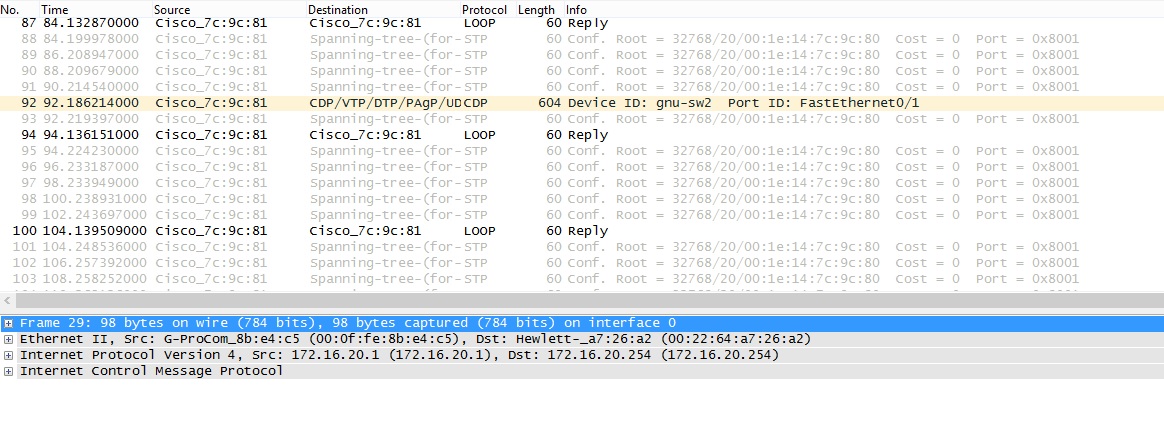
**ANEXO B – *Logs* capturados no *Wireshark***

1. Experiência 1

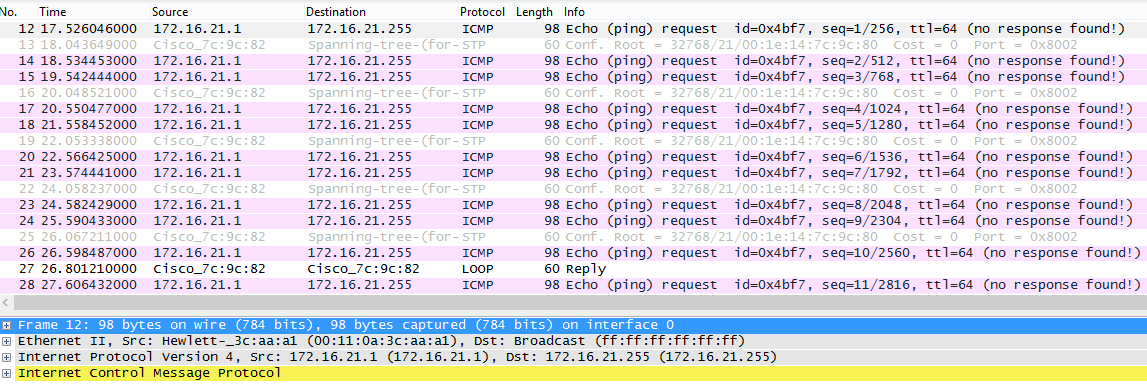
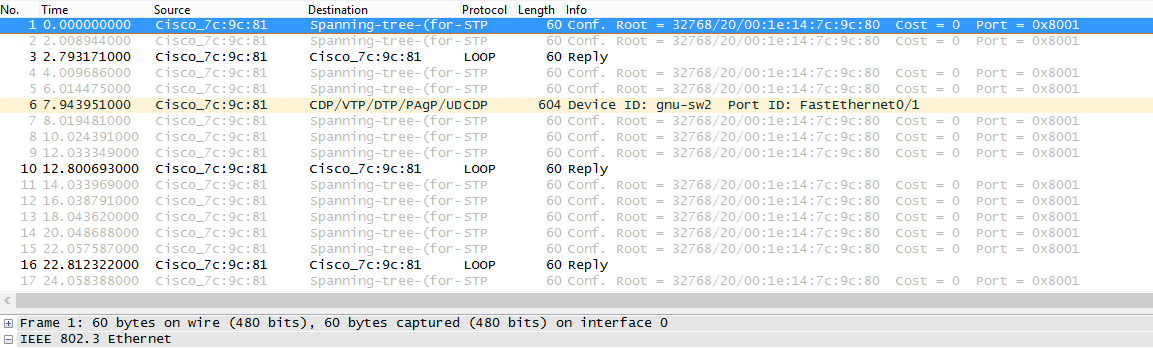
1 - Log resultante do ping do tux 21 para o tux 24. Note-se os pacotes ARP, responsáveis pela informação a inserir na tabela ARP, bem como, neste caso, do tipo de trama ICMP (igual a 8 para “echo request”)

1. Experiência 2

2 - Log resultante do ping pelo tux 21, visto deste. O log resultante, visto do tux24, obtém resultado idêntico.

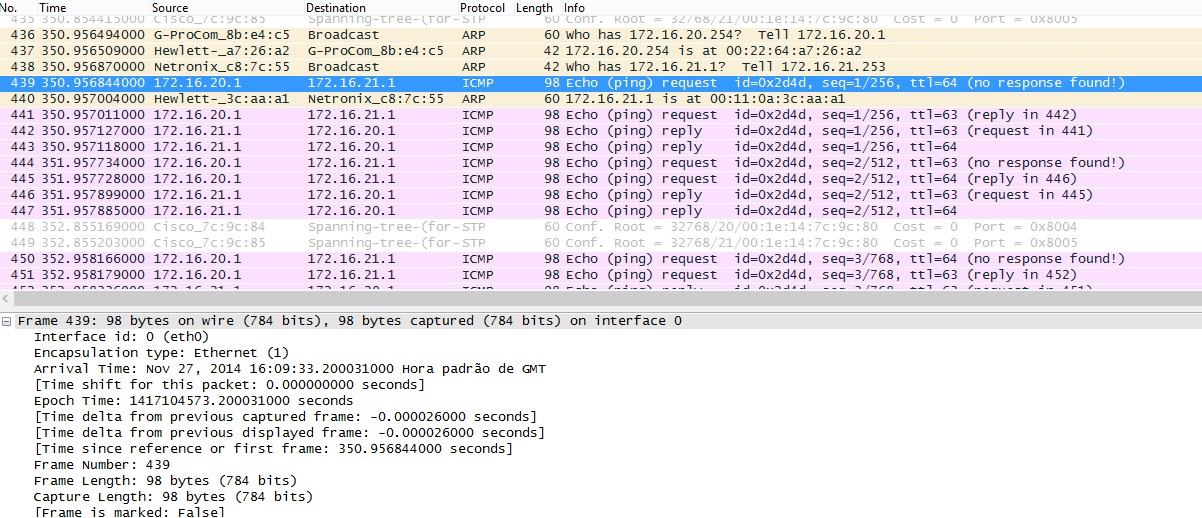


3 – Log resultante do ping pelo tux21, visto no tux22.



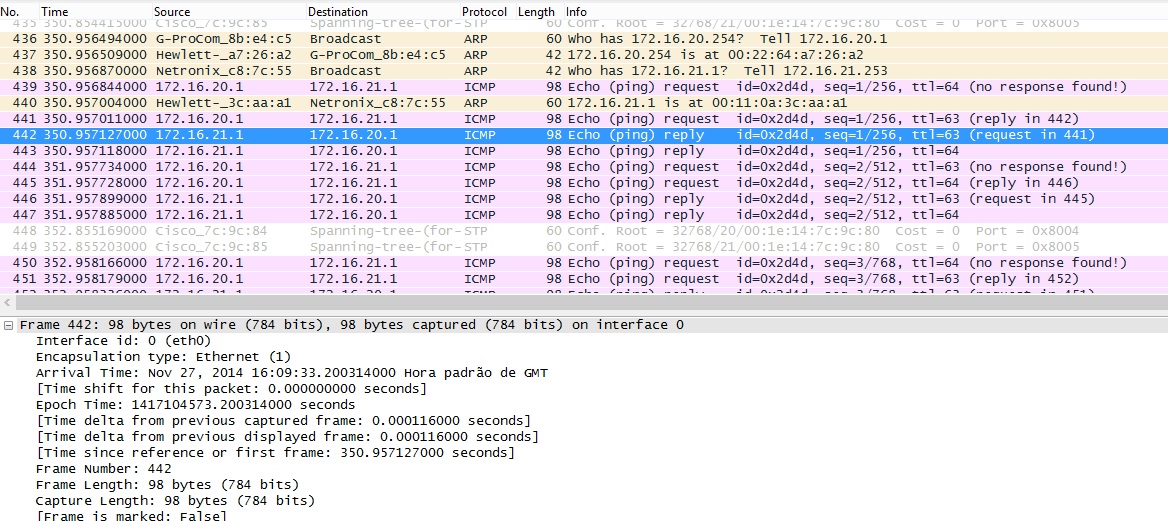
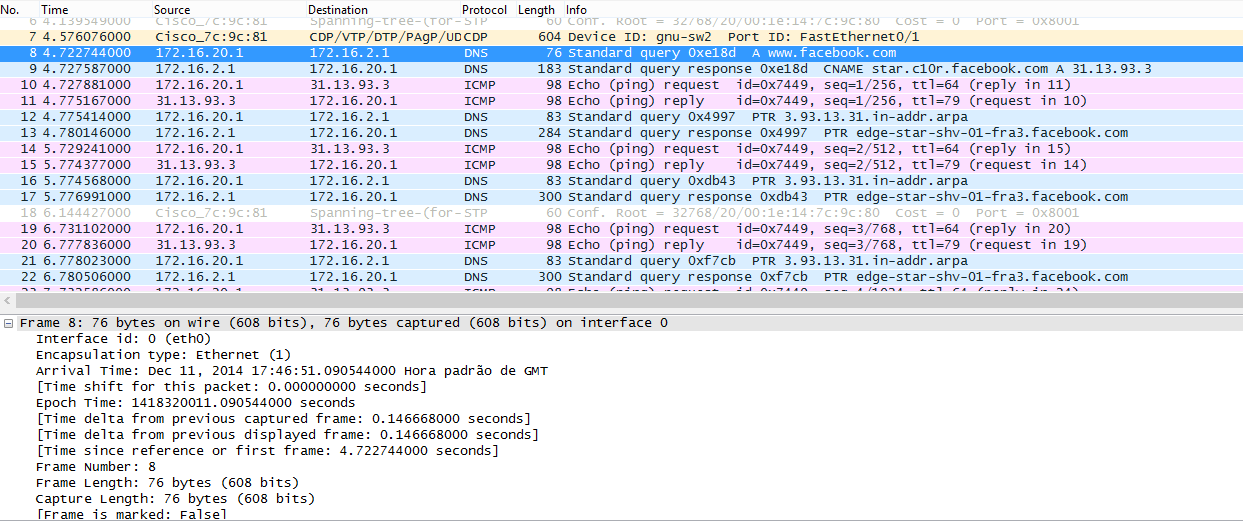
5 – Log resultante do ping pelo tux22, visto no tux24. O log resultante do mesmo ping, visto pelo tux21, obtém o mesmo resultado.

4 – Log resultante do ping pelo tux22, visto no tux22.

1. Experiência 3

6 – Log resultante do ping pelo tux21, visto no tux24, relativa à interface eth0.

6 – Log resultante do ping pelo tux21, visto no tux24, relativa à interface eth1.

1. Experiência 4
2. Experiência 5

7 – Log resultante do ping a www.facebook.com