

Rede de Computadores

Relatório
Trabalho1



Universidade do Porto

Faculdade de Engenharia

FEUP

Mestrado Integrado em Engenharia Informática e Computação

Redes de Computadores

Hugo Ari Rodrigues Drumond — 201102900 — hugo.drumond@fe.up.pt

José Pedro Pereira Amorim — 201206111 — ei12190@fe.up.pt

João Ricardo Pintas Soares — 201200740 — ei12039@fe.up.pt

Faculdade de Engenharia da Universidade do Porto
Rua Roberto Frias, 4200-65 Porto, Portugal

23 de Dezembro de 2014

1 Introdução

Este trabalho laboratorial, desenvolvido no âmbito da Unidade Curricular de Redes de Computadores (RCOM), teve como objetivo implementar um protocolo de ligação de dados, do tipo acknowledged connection-oriented, e testá-lo em diversas situações de stress de modo a verificar a sua robustez. Ao longo deste relatório, serão descritos os aspetos fundamentais do referido trabalho, permitindo obter um conhecimento detalhado deste. Será apresentada a arquitetura, estrutura do código, casos de uso principais, protocolo de ligação lógica e de aplicação. No mesmo sentido, serão apresentadas a validação dos resultados e os elementos de valorização.

2 Parte 1

3 Parte 2

3.1 Experiência 1

3.1.1 What are the ARP packets and what are they used for?

ARP (Address Resolution Protocol) é um protocolo usado para obter um endereço da link layer, a partir de um endereço da network layer. Normalmente usado para obter o endereço MAC a partir de um endereço IP. O emissor difunde em broadcast um pacote ARP Request contendo o endereço IP do alvo e aguarda resposta do mesmo. O host alvo após recepção do pacote ARP Request responde em unicast, enviando um pacote ARP Reply contendo o seu endereço MAC. ARP é usado para construir e manter a ARP Table, contendo as relações de endereços da link layer e endereços da network layer, normalmente relações entre um endereço MAC a um endereço IP. Entradas dinâmicas desta são frequentemente guardadas com um timeout de até 20 minutos, o que significa que após um host ter obtido um endereço MAC através do ARP Request, vai necessitar de o voltar a enviar após 20 minutos. Normalmente um host envia ARP Requests a todos os endereços listado na ARP Table periodicamente para renovar as entradas.

3.1.2 What are the MAC and IP addresses of ARP packets and why?

Um pacote ARP contém o endereço MAC e o endereço IP do emissor assim como os do alvo. Num pacote ARP Request o emissor envia o seu endereço MAC, e IP e o endereço IP do alvo deixando o endereço MAC do alvo em branco. Num pacote ARP Reply todos endereços são preenchidos.

3.1.3 What packets does the ping command generate?

O comando ping envia para o alvo um pacote Echo Request, recebendo depois um pacote Echo Reply proveniente do host alvo.

3.1.4 What are the MAC and IP addresses of the ping packets?

Os endereços MAC do emissor e do alvo estão localizados no header do Ethernet Frame e os endereços IP do emissor e alvo na header do pacote.

3.1.5 How to determine if a receiving Ethernet frame is ARP, IP, ICMP?

O campo type da header do Ethernet frame indica o tipo de pacote, o seu valor é 0x0806 para pacotes do tipo ARP e 0x0800 para pacotes do tipo IP. Os pacotes IP com o campo protocol com valor 1 são pacotes ICMP.

6. How to determine the length of a receiving frame? Se o campo type do Ethernet Frame tiver valor igual ou inferior a 1500, então esse valor indica o tamanho do payload, senão indica o tipo de pacote.

3.1.6 What is the loopback interface and why is it important?

A interface loopback é uma interface (virtual) especial porque permite que um computador comunique com ele mesmo. Um dos exemplos mais comuns é poder-se ligar a um servidor que está em execução numa máquina local, nomeadamente: servidor dns, unbound; servidor http, lighttpd; servidor de filestorage, git-annex; servidor web search, yacy; servidor de impressão, cups; servidor de encriptação de dns, dnscrypt; etc. Como é uma interface virtual permite que as aplicações e utilizadores comuniquem com um servidor (localmente) mesmo que não haja conexão internet. Para além disto, é usado para: testar configurações; fazer simulações; resolução de problemas entre aplicações de modo rápido. Pode haver uma rede de loopback interfaces 127.0.0.0/8, o que torna fácil correr servidores que corram numa mesma porta facilmente.

3.2 Experiência 2

3.2.1 How to configure vlans?

Foram criadas duas vlans de modo a isolar as maquinas tux1 e tux4 da tux2. A interface eth0 da tux1 (172.16.y0.1) e tux4 (172.16.y0.254) foram ligadas ao switch e depois cada uma das portas foi adicionada a uma vlan y0; a tux2 (172.16.y0.2) foi configurada de igual modo só que a respetiva porta foi adicionada a outra vlan y1. Verificámos que não é possível comunicar com a máquina tux2, como seria de esperar.

Comandos usados para criar cada uma das vlans

```
configure terminal
vlan x0
end
show vlan id x0
```

Comandos para adicionar uma dada porta a uma vlan

```
configure terminal
interface fastEthernet 0/porta
switchport mode access
switchport access vlan x0
end
show running-config interface fastEthernet 0/porta
show interfaces fastEthernet 0/porta switchport
```

50	VLAN0050	active	Fa0/1, Fa0/2
51	VLAN0051	active	Fa0/3

3.2.2 How many broadcast domains are there? How can you conclude it from the logs?

Existem dois domínios de broadcast, devido a existir duas vlans (sem interligação entre elas). Através dos logs verificamos que quando fazemos broadcast no tux1 só o tux4 responde. E quando fazemos broadcast no tux2 nenhuma máquina responde, porque não chega nenhum pedido às outras máquinas.

3.3 Experiência 3

3.3.1 What routes are there in the tuxes? What are their meaning?

No tux1 existem duas rotas.

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
172.16.50.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
172.16.51.0	172.16.50.254	255.255.255.0	UG	0	0	0	eth0

A primeira tem o significado de aceitar o tráfego proveniente de 172.16.50.0/24. A outra faz com que haja um redirecionamento dos ips to tipo 172.16.51.0/24 para o gateway router 172.16.50.254(eth0 tux4).

No tux2 existem duas rotas e a tabela é semelhante à do tux1.

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
172.16.50.0	172.16.51.253	255.255.255.0	UG	0	0	0	eth0
172.16.51.0	0.0.0.0	255.255.255.0	Ui	0	0	0	eth0

Aceita os ips provenientes da rede 172.16.51.0/24. E faz forward dos ips 172.16.50.0/24 para o gateway router 172.16.51.253(eth1 tux4).

I tux4 funciona como um gateway router, porque redireciona de uma interface (eth0) para outra (eth1).

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
172.16.50.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
172.16.51.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1

Aceita o tráfego proveniente destas redes e não manda para nenhum gateway visto que os clientes estão ligados diretamente, isto é, não há um decremento do ttl.

3.3.2 What information does an entry of the forwarding table contain?

Uma vez que a nossa routing table só tem static routes a forwarding table irá resultar nas mesmas routes. Porque não é feita uma descoberta da rede, através de routing protocols para construir a routing table, e aplicado o algoritmo de Dijkstra para só depois construir a forwarding table, com o resultado desse processo. Uma forwarding table contém os seguintes campos: Destination; Gateway; Genmask; Flags; Metric; Ref; Use; e, Iface.

3.3.3 What ARP messages, and associated MAC addresses, are observed and

why?

Uma vez que a ethernet usa MAC addresses e a ARP table foi limpa, é necessário saber descobrir qual o mac address para um determinado IP. Que é a função do protocolo ARP. O tux1 vai precisar de saber qual o mac address da interface eth0 do tux4, o tux4 vai precisar de saber qual o mac do tux3, o tux3 o mac do eth1 tux4, e o tux4 o mac do tux1. Uma vez descoberto o mac address, o mesmo, é escrito na trama. Protocolo simplificado, pedido -> broadcast ARP com ip alvo e ip de retorno; reply -> ip alvo com mac address.

HWaddr 00:21:5a:c3:78:70, é o mac address da interface eth0(172.16.50.254) do tux4.

HWaddr 00:c0:df:08:d5:b0, é o mac address da interface eth1(172.16.51.253) do tux4.

HWaddr 00:0f:fe:8b:e4:a7, é o mac address da interface eth0(172.16.50.1) do tux1.

HWaddr 00:21:5a:61:2f:d6, é o mac address da interface eth0(172.16.51.1) do tux2.

3.3.4 What ICMP packets are observed and why?

São observados pacotes ICMP de request(tipo 8) e reply(tipo 0). Porque foi usado o comando ping que trata de fazer os pedidos e que espera por uma resposta. Acontece uma coisa um pouco estranha que é o facto do tux3 parecer estar a fazer reply sem antes saber o mac do tux4 eth1 (através dos logs do wireshark). Só depois de algum tempo é que se vê o broadcast ARP a partir do tux3. Pensamos que isto é devido ao tux3 ter respondido ao broadcast do tux4, e de ter ficado de modo implícito com o mac do tux4 eth1 em cache, sendo só depois feito o pedido formal. O mesmo acontece do tux4 para o tux1.

3.3.5 What are the IP and MAC addresses associated to ICMP packets and

why? Os endereços do Network Layer presentes no protocolo IP são os ips de destino e fonte. E os endereços do Data Link Layer necessários para o protocolo Ethernet II são os macs de destino e fonte, sendo estes dependentes das máquinas por onde passam. É necessário haver estes dois tipos de endereços porque é necessário saber de que máquina veio a mensagem(de forma unívoca) mas também o ip fonte de origem; do mesmo modo para o destino. E também porque os protocolos assim o dizem.

3.4 Experiência 4

3.4.1 How to configure a static route in a commercial router?

É feito de modo semelhante ao que se faz nas máquinas que correm linux.

Cria um default gateway,
ip route 0.0.0.0 0.0.0.0 172.16.1.254

Faz um redirecionamento,
ip route 172.16.50.0 255.255.255.0 172.16.51.253

3.4.2 What are the paths followed by the packets in the experiments carried out

and why? Através do algoritmo das forwarding tables(escolha de linhas) e das entradas presentes em cada máquina. Acontece o seguinte:

Tux1 -> net,
172.16.50.1(tux1,eth0) -> 172.16.50.254(tux4, eth0) -> 172.16.51.254(routerCisco, eth0) -> 172.16.1.254(routerCisco, eth1)

Tux2 -> net,
172.16.51.1(tux1,eth0) -> 172.16.51.254(routerCisco, eth0) -> 172.16.1.254(routerCisco, eth1)

Tux4 -> net,
172.16.51.254(tux4,eth1) -> 172.16.51.254(routerCisco, eth0) -> 172.16.1.254(routerCisco, eth1)

Tux1 -> Tux2,
172.16.50.1(tux1) -> 172.16.50.254(tux4, eth0) -> 172.16.51.1(tux2)

Tux2 -> Tux1,
172.16.51.1(tux2) -> 172.16.51.253(tux4,eth1) -> 172.16.50.1(tux1)

Isto acontece devido às seguintes routes:

```

Tux1,
  Destination  Gateway      Genmask      Flags  Metric  Ref  Use  Iface
  0.0.0.0      172.16.50.254  0.0.0.0      UG      0        0    0    eth0
  172.16.50.0  0.0.0.0       255.255.255.0 U      0        0    0    eth0

Tux2,
  Destination  Gateway      Genmask      Flags  Metric  Ref  Use  Iface
  0.0.0.0      172.16.51.254  0.0.0.0      UG      0        0    0    eth0
  172.16.50.0  172.16.51.253  255.255.255.0 UG      0        0    0    eth0
  172.16.51.0  0.0.0.0       255.255.255.0 U      0        0    0    eth0

Tux4,
  Destination  Gateway      Genmask      Flags  Metric  Ref  Use  Iface
  0.0.0.0      172.16.51.254  0.0.0.0      UG      0        0    0    eth1
  172.16.50.0  0.0.0.0       255.255.255.0 U      0        0    0    eth0
  172.16.51.0  0.0.0.0       255.255.255.0 U      0        0    0    eth1

Router Cisco,
ip route 0.0.0.0 0.0.0.0 172.16.1.254
ip route 172.16.50.0 255.255.255.0 172.16.51.253

```

3.4.3 How to configure NAT in a commercial router?

Usamos o template presente no guião de modo a fazer esta configuração.

```

conf t
interface gigabitethernet 0/0
ip address 172.16.51.254 255.255.255.0
no shutdown
ip nat inside # Interface do lado lan
exit
interface gigabitethernet 0/1
ip address 172.16.1.59 255.255.255.0
no shutdown
ip nat outside # Interface do lado "wan"
exit
ip nat pool ovrlld 172.16.1.59 172.16.1.59 prefix 24
ip nat inside source list 1 pool ovrlld overload
# Esta permissão não deixa que o tux4 tenha acesso à net
# 0.0.0.255 para as permissões serem iguais para os hosts
access-list 1 permit 172.16.50.0 0.0.0.7
access-list 1 permit 172.16.51.0 0.0.0.7
# Default gateway do router com nat
ip route 0.0.0.0 0.0.0.0 172.16.1.254
# Para falar com os hosts do outro lado do router
ip route 172.16.50.0 255.255.255.0 172.16.51.253
end

```

3.4.4 What does NAT do?

Esta técnica permite que se possa utilizar os mesmos ips em diferentes redes, chamadas redes privadas(10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16). Isto é possível porque os ips privados são traduzidos para um público. Este mapeamento é feito através de uma tabela chamada tabela de tradução de endereços de rede, que possibilita que o router saiba para que hosts são destinados certos pacotes vindos da internet. Cada entrada da tabela tem uma parte para o lado WAN e

outra para o lado LAN, cada um deles tem dois campos: um para o ip e a outra para a porta. A porta é o que permite fazer a distinção entre destinos de pacotes. É possível reencaminhar todos os pacotes wan com destino a uma dada porta (por exemplo, porta default para o server http, 80) para uma dada porta de um host, que terá um processo em modo listening associado a essa porta.

3.5 Experiência 5

3.5.1 How to configure the DNS service at an host?

Para configurar o serviço DNS alterou-se o ficheiro resolv.conf localizado no diretório /etc do host. Nesta experiência o conteúdo deste ficheiro foi alterado para:

```
search lixa.netlab.fe.up.pt
nameserver 172.16.1.1
```

172.16.1.1, é o endereço do servidor de nomes para onde se fazem os pedidos. lixa.netlab.fe.up.pt, é o único elemento da lista de procura de hostnames.

3.5.2 What packets are exchanged by DNS and what information is transported?

Existem três tipos de DNS packets: Queries, Responses e Updates. Os pacotes de DNS queries e reponses(replies) têm um formato comum, para além dos 12 bytes do cabeçalho: identificação; flags; número de perguntas; número de respostas de RRs; número de registos de servidores authoritative; e, número adicional de RRs. São enviadas: perguntas com os campos nome e tipo(A, NS, CNAME e MX) seleccionados; respostas a uma pergunta, RRs completos; registo de servidores authoritative; e, informação adicional útil. O pacote de update contém à semelhança dos outros um header com: uma identificação; flags; e, número de registo de recursos(RRs) para cada tipo da secção seguinte: zone entry; prerequisite resource records; update resource records; e, additional resource records.

3.6 Experiência 6

3.6.1 How many TCP connections are opened by your ftp application?

São abertas duas conexões TCP pela aplicação FTP, uma de controlo e outra de dados. Primeiramente a aplicação cria uma conexão TCP de controlo que conecta um porto sem privilégios (porto > 1023) ao porto FTP do servidor (porto 21) que vai continuar aberta durante a transferência. Foi utilizado nesta aplicação o modo passivo por oposição ao modo activo, sendo o cliente a abrir a conexão TCP de dados. No modo activo o cliente envia para o servidor um número de porto aleatório e sem privilégios, para se conectar e é o próprio servidor a abrir a conexão TCP de dados, ligando a porto do cliente recebido e o seu porto 20. No modo passivo o servidor envia para o cliente um número de porto aleatório e sem privilégios, e é o cliente que abre a conexão TCP de dados.

3.6.2 In what connection is transported the FTP control information?

A informação de controlo FTP é transportada na conexão de controlo que liga um porto aleatório do cliente e o porto 21 do servidor. É também transportado as mensagens de acknowledgement (ACK).

3.6.3 What are the phases of a TCP connection?

3.6.4 How does the ARQ TCP mechanism work? What are the relevant TCP fields? What relevant information can be observed in the logs?

3.6.5 How does the TCP congestion control mechanism work? What are the relevant fields. How did the throughput of the data connection evolve along the time? Is it according the TCP congestion control mechanism?

3.6.6 Is the throughput of a TCP data connections disturbed by the appearance of a second TCP connection? How?

4 Conclusões

O projeto desenvolvido resultou numa aplicação capaz de transferir ficheiros e mensagens através duma ligação com a porta série RS-232 (comunicação assíncrona). Ligação esta customizada pelo utilizador através de argumentos na linha de comandos. A separação e encapsulamento das várias partes do projeto facilitou o seu desenvolvimento, uma vez que tornou a linklayer completamente independente da applayer, e a applayer do parser. Um dos pontos que queríamos garantir já de partida. A elaboração deste trabalho contribuiu para a aprendizagem dos conhecimentos básicos de funcionamento de um protocolo de dados e de como o implementar de maneira robusta. O trabalho foi realizado com sucesso, dado que passou todos os testes feitos na demonstração e porque foram cumpridos todos os elementos de valorização apresentados pelo professor.