

# Trabalho Prático Nº2 – Protocolo IPv4 :: Datagramas IP e Fragmentação (1ª Parte)

Duração: 4h

Neste trabalho deve usar a máquina virtual XubunCORE\_7\_5 (TP0) para a Questão 1 e a máquina nativa para as Questões 2 e 3.

Nota importante: O trabalho é para ser realizado nas aulas PL correspondentes. Não serão aceites trabalhos "resolvidos em casa".

## 1. Objetivo

O principal objetivo deste trabalho é o estudo do Internet Protocol (IP) nas suas principais vertentes, nomeadamente: (i) estudo do formato de um pacote ou datagrama IP; (ii) fragmentação de pacotes IP; (iii) endereçamento IP; e (iv) encaminhamento IP.

Na primeira parte deste estudo é realizado o registo de datagramas IP enviados e recebidos através da execução do programa `traceroute`. São analisados os vários campos de um datagrama IP e detalhado o processo de fragmentação realizado pelo protocolo IP. Para tal, o computador de trabalho deve estar conectado à rede da sala de aula.

## 2. Captura de tráfego IP

Com o objetivo de obter um registo de tráfego IP, pretende-se usar o programa `traceroute` para descobrir uma rota IP, enviando pacotes de diferentes tamanhos para um determinado destino X.

O comando `traceroute` permite descobrir a rota (salto-a-salto) desde uma origem IP até um destino IP, tirando partido da escolha de valores adequados para o "tempo-de-vida" indicado no cabeçalho IP dos datagramas enviados. O `traceroute` opera da seguinte forma: inicialmente, é enviado um ou mais datagramas com o campo TTL (*Time-To-Live*) igual a 1; seguidamente, é enviado um ou mais datagramas com o TTL a 2; depois com o TTL a 3; e assim sucessivamente. Todos os pacotes são enviados para o mesmo destino, especificado no comando `traceroute`.

Recorda-se que cada *router* no percurso até ao destino deve decrementar de 1 o TTL de cada datagrama recebido. Se o TTL atinge o valor zero, o *router* descarta o datagrama e devolve uma mensagem de controlo ICMP (*Internet Control Message Protocol*) ao *host* de origem, indicando que o TTL foi excedido (ICMP Type=11 - *TTL exceeded*). Como resultado, o datagrama com o TTL=1 (enviado pelo *host* que executa o `traceroute`) faz com que o *router* a um salto de distância envie uma mensagem ICMP para a origem. O datagrama com TTL=2 provoca esse comportamento no *router* a dois saltos de distância e assim sucessivamente.

Desta forma, um *host* que execute o comando `traceroute` pode obter a identificação dos *routers* no percurso para o destino X, extraindo o endereço IP fonte dos datagramas que contenham mensagens ICMP do tipo TTL excedido.

## 3. Questões

1. Prepare uma topologia CORE para verificar o comportamento do `traceroute`. Na topologia deve existir: um *host* (pc) cliente designado *Lost*, cujo *router* de acesso é RA1; o *router* RA1 está simultaneamente ligado a dois *routers* no *core* da rede RC1 e RC2; estes estão conectados a um *router* de acesso RA2, que por sua vez, se liga a um *host* (servidor) designado *Found*. Ajuste o nome dos equipamentos atribuídos por defeito para o enunciado. Apenas nas ligações (links) da rede de core, estabeleça um tempo de propagação de 15ms. Após ativar a topologia, note que pode não existir conectividade IP imediata entre *Lost* e *Found* até que o anúncio de rotas entre *routers* estabilize.

Documente e justifique todas as respostas às seguintes alíneas:

- Active o Wireshark no *host Lost*. Numa *shell* de *Lost* execute o comando `traceroute -I` para o endereço IP do *Found*. Registe e analise o tráfego ICMP enviado pelo sistema *Lost* e o tráfego ICMP recebido como resposta. Explique os resultados obtidos tendo em conta o princípio de funcionamento do `traceroute`.
- Qual deve ser o valor inicial mínimo do campo TTL para alcançar o servidor *Found*? Esboce um esquema com o valor do campo TTL à chegada a cada um dos *routers* percorridos até ao servidor *Found*. Verifique na prática que a sua resposta está correta.
- Calcule o valor médio do tempo de ida-e-volta (RTT - *Round-Trip Time*) obtido no acesso ao servidor. Por modo a obter uma média mais confiável, poderá alterar o número pacotes de prova com a opção `-q`.
- O valor médio do atraso num sentido (*One-Way Delay*) poderia ser calculado com precisão dividindo o RTT por dois? O que torna difícil o cálculo desta métrica numa rede real?

2. Pretende-se agora usar o traceroute na sua máquina nativa e gerar datagramas IP de diferentes tamanhos.

Windows. O programa `tracert` disponibilizado no Windows não permite mudar o tamanho das mensagens a enviar. Como alternativa, o programa `pingplotter` (ou equivalente) na sua versão livre ou *shareware* (<http://www.pingplotter.com>) permite maior flexibilidade para efetuar traceroute. Descarregue, instale e experimente o `pingplotter` face ao objectivo pretendido.

O tamanho da mensagem a enviar (*ICMP Echo Request*) pode ser estabelecido no `pingplotter` no menu `Edit -> Options -> Default Settings -> Engine`. Uma vez enviado um conjunto de pacotes com valores crescentes de TTL, o programa recomeça com TTL=1, após um determinado intervalo. Tanto o valor do intervalo de tempo como o número de intervalos podem ser configurados.

Linux/Unix. O comando `traceroute` permite indicar o tamanho do pacote ICMP (opção `-l`) através da linha de comando, a seguir ao `host` de destino (ver `man traceroute`).

Exemplo: `% traceroute -l router-di.uminho.pt 512`

Documente as suas respostas com a impressão do(s) output(s) (e.g. pacote(s)) que as suportam.

Procedimento a seguir:

Usando o `Wireshark` capture o tráfego gerado pelo traceroute para os seguintes tamanhos de pacote: situação (i) sem especificar, i.e., usando o tamanho do pacote de prova por defeito; e situação (ii)  $(3500 + X)$  bytes, em que  $X$  é o número do grupo de trabalho. Utilize como destino o `host` `marco.uminho.pt`. Pare a captura. Com base no tráfego capturado, identifique os pedidos *ICMP Echo Request* e o conjunto de mensagens devolvidas como resposta.

Usando o `Wireshark` capture o tráfego gerado pelo traceroute sem especificar o tamanho do pacote, i.e., quando é usado o tamanho do pacote de prova por defeito. Utilize como máquina destino o `host` `marco.uminho.pt`. Pare a captura. Com base no tráfego capturado, identifique os pedidos *ICMP Echo Request* e o conjunto de mensagens devolvidas como resposta.

Selecione a primeira mensagem ICMP capturada e centre a análise no nível protocolar IP e, em particular, do cabeçalho IP (expanda o *tab* correspondente na janela de detalhe do `Wireshark`).

Documente e justifique todas as respostas às seguintes alíneas:

- a. Qual é o endereço IP da interface ativa do seu computador?
- b. Qual é o valor do campo `protocol`? O que permite identificar?
- c. Quantos bytes tem o cabeçalho IPv4? Quantos bytes tem o campo de dados (`payload`) do datagrama? Como se calcula o tamanho do `payload`?
- d. O datagrama IP foi fragmentado? Justifique.
- e. Analise a sequência de tráfego ICMP gerado a partir do endereço IP atribuído à interface da sua máquina. Para a sequência de mensagens ICMP enviadas pelo seu computador, indique que campos do cabeçalho IP variam de pacote para pacote. Justifique estas mudanças.
- f. Observa algum padrão nos valores do campo de Identificação do datagrama IP e TTL?
- g. Ordene o tráfego capturado por endereço destino e encontre a série de respostas ICMP TTL *Exceeded* enviadas ao seu computador.
  - i. Qual é o valor do campo TTL recebido no seu computador? Esse valor permanece constante para todas as mensagens de resposta ICMP TTL *Exceeded* recebidas no seu computador? Porquê?
  - ii. Por que razão as mensagens de resposta ICMP TTL *Exceeded* são sempre enviadas na origem com um valor relativamente alto?
- h. A informação contida no cabeçalho ICMP poderia ser incluída no cabeçalho IPv4? Se sim, quais seriam as suas vantagens/desvantagens?

3. Pretende-se agora analisar a fragmentação de pacotes IP. Usando o Wireshark, capture e observe o tráfego gerado depois do tamanho de pacote ter sido definido para  $(3800 + X)$  bytes, em que  $X$  é o número do grupo de trabalho (e.g.,  $X=22$  para o grupo PL22). De modo a poder visualizar os fragmentos, aceda a Edit -> Preferences -> Protocols e em IPv4 desative a opção "Reassemble fragmented IPv4 datagrams".

Nota: Como alternativa para geração do tráfego pode usar o comando `ping <opção> <bytes> marco.uminho.pt`, onde a opção `-l` (Windows) ou `-s` (Linux, Mac) permite definir o número de bytes enviados no campo de dados do pacote ICMP.

Documente e justifique todas as respostas às seguintes alíneas:

- Localize a primeira mensagem ICMP. Porque é que houve necessidade de fragmentar o pacote inicial?
- Imprima o primeiro fragmento do datagrama IP segmentado. Que informação no cabeçalho indica que o datagrama foi fragmentado? Que informação no cabeçalho IP indica que se trata do primeiro fragmento? Qual é o tamanho deste datagrama IP?
- Imprima o segundo fragmento do datagrama IP original. Que informação do cabeçalho IP indica que não se trata do 1º fragmento? Existem mais fragmentos? O que nos permite afirmar isso?
- Quantos fragmentos foram criados a partir do datagrama original? Como se detecta o último fragmento correspondente ao datagrama original? Estabeleça um filtro no Wireshark que permita listar apenas o último fragmento do primeiro datagrama IP segmentado.
- Indique, resumindo, os campos que mudam no cabeçalho IP entre os diferentes fragmentos, e explique a forma como essa informação permite reconstruir o datagrama original.
- Estime teoricamente o número de fragmentos gerados e o número de bytes transportados em cada um dos fragmentos. Apresente todos os cálculos efetuados, incluindo os campos do cabeçalho IP relevantes para cada um dos fragmentos.
- Por que razão apenas o primeiro fragmento de cada pacote é identificado pelo Wireshark como sendo um pacote ICMP? Justifique a sua resposta com base no conceito de Fragmentação apresentado nas aulas teóricas.
- Com que valor é o tamanho do datagrama comparado a fim de se determinar se este deve ser fragmentado? Quais seriam os efeitos na rede ao aumentar/diminuir este valor?
- Sabendo que no comando `ping` a opção `"-f"` (Windows), `"-M do"` (Linux) ou `"-D"` (Mac) ativa a *flag* "Don't Fragment" (DF) no cabeçalho do IPv4, usando `ping <opção DF> <opção pkt_size> SIZE marco.uminho.pt`, (opção `pkt_size` = `-l` (Windows) ou `-s` (Linux, Mac)), determine o valor máximo de `SIZE` sem que ocorra fragmentação do pacote? Justifique o valor obtido.

(Fim da Parte I)

Nota: Os alunos devem entregar um único relatório, incluindo a resolução das partes I e II (a publicar).

## Bibliografia

Internetworking - Protocolo IP (Notas de Apoio das Aulas Teóricas)

traceroute: <http://tools.ietf.org/html/rfc2151> (secção 3.4)

Internet Protocol (IP): <http://tools.ietf.org/html/rfc791>

Internet Message Control Protocol (ICMP): <http://tools.ietf.org/html/rfc792>