



RC-I 2024-2025	TRABALHO DE LABORATÓRIO	Número:	2.2
CAMADA DE APLICAÇÃO		Data:	
Aplicações Web e HTTP (Complementar)		Prazo:	

1. Introdução

O objectivo deste laboratório (complementar) é a familiarização com alguns conceitos básicos do protocolo HTTP. Serão observados os mecanismos de pedido/resposta, os formatos das mensagens HTTP e os mecanismos de redirecção. Os estudantes deverão seguir as instruções contidas neste guia e responder às questões colocadas.

As questões devem ser respondidas tendo em conta as informações apresentadas pela análise do tráfego na infraestrutura previamente configurada. O estudante deve instalar o Wireshark na máquina física e usar o browser ou o wget. Nessa experiência devem lançar Wireshark e o browser, que deve permanecer aberto até ao final. Abra o seguinte URL <http://localhost:8080> no browser que irá automaticamente contactar o website configurado. Desta forma, o browser deverá ser lançado antes de começar a captura de pacotes no Wireshark, por forma a ignorar este tráfego.

Cada uma das secções descreve uma experiência. Assim, para cada secção deve iniciar a captura de pacotes, utilizando o Wireshark, parar a captura e responder às questões colocadas, analisando os pacotes capturados.

2. Carregamento de uma página

2.1. Procedimento

- Limpa a cache do chrome (ctrl+shift+Delete)
- Aceda a <http://localhost:8080>

2.2. Questões

1. Qual o comando HTTP utilizado para pedir a página?

R : O comando é GET. O pedido no Frame 92 começa com: GET / HTTP/1.1

2. Qual o endereço IP do servidor?

R: O endereço IP do servidor é 127.0.0.1 (Loopback). O campo Destination Address (Endereço de Destino) no Frame 92 é 127.0.0.1.

3. Quantos pedidos HTTP foram efectuados?

R: 1 pedido principal. O procedimento pede para aceder uma vez à página (<http://localhost:8080>). O Frame 92 mostra este pedido GET.

4. Qual o código e mensagem de resposta enviados pelo servidor em resposta ao pedido do ficheiro transferido ?

R: 200 OK. O Frame 97 (resposta ao Frame 92) mostra: HTTP/1.1 200 OK.



5. Quais as linguagens indicadas pelo seu browser como preferidas? (faça um print da tela)

Wireshark screenshot showing network traffic. The packet list shows multiple HTTP requests from 127.0.0.1 to 127.0.0.1. The details pane shows the request headers, including 'Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7\r\n'.

R: **pt-PT, pt;q=0.9, en-US;q=0.8, en;q=0.7.**

O Frame 92 (o pedido) contém o cabeçalho Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7.

6. Qual o nome do software utilizado como servidor HTTP?

R: **Apache/2.4.7 (Ubuntu).** O Frame 97 contém o cabeçalho Server: Apache/2.4.7 (Ubuntu).

7. Qual o tamanho da página apresentada pelo browser?

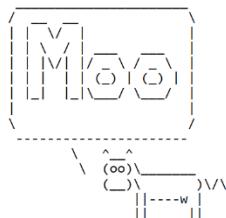
R: **173 bytes (comprimido) ou 420 bytes (descomprimido).**



3. Mecanismo de cache

3.1. Procedimento

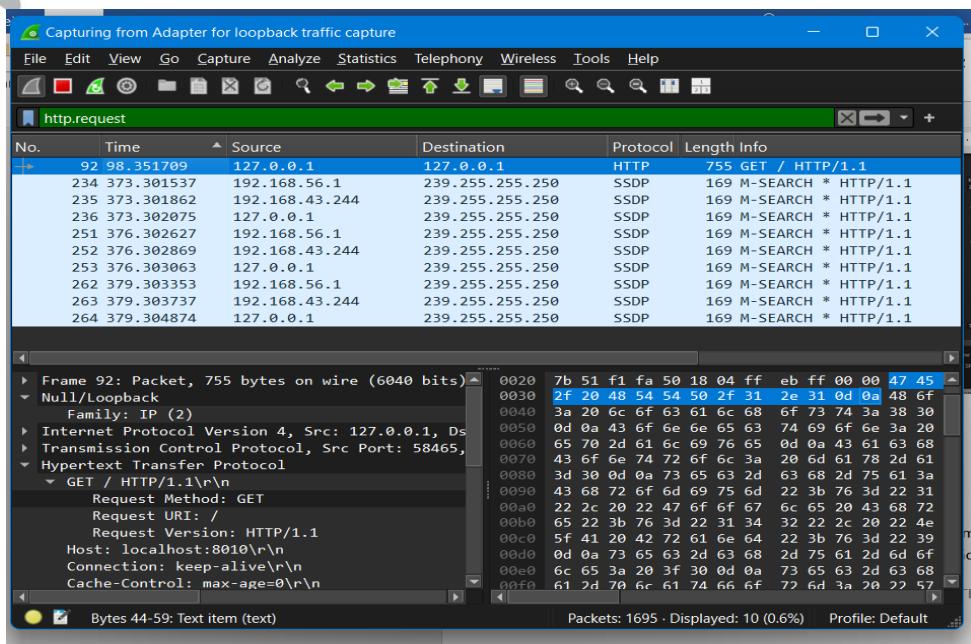
- Limpar a cache do browser
- Aceder a <http://localhost:8080> (repere na barra no fim)
- Imediatamente a seguir, carregar no botão de reload do browser.
- Esperar 2 minutos.
- Carregar novamente no botão de reload.



The screenshot shows a Windows desktop environment. In the center, there is a Microsoft Word document titled "RCI2024-Lab2-v0.1 - parte 2.doc [Modo de Compatibilidade]". Below it, a Wireshark window is open, capturing traffic from "Adapter for loopback traffic capture". The packet list pane shows several TCP and UDP packets between the source IP 127.0.0.1 and destination IP 127.0.0.1. The details and bytes panes provide detailed information about each packet, including flags like ACK, SYN, and FIN, and sequence numbers. The packet details pane also shows the raw hex and ASCII data for each packet.

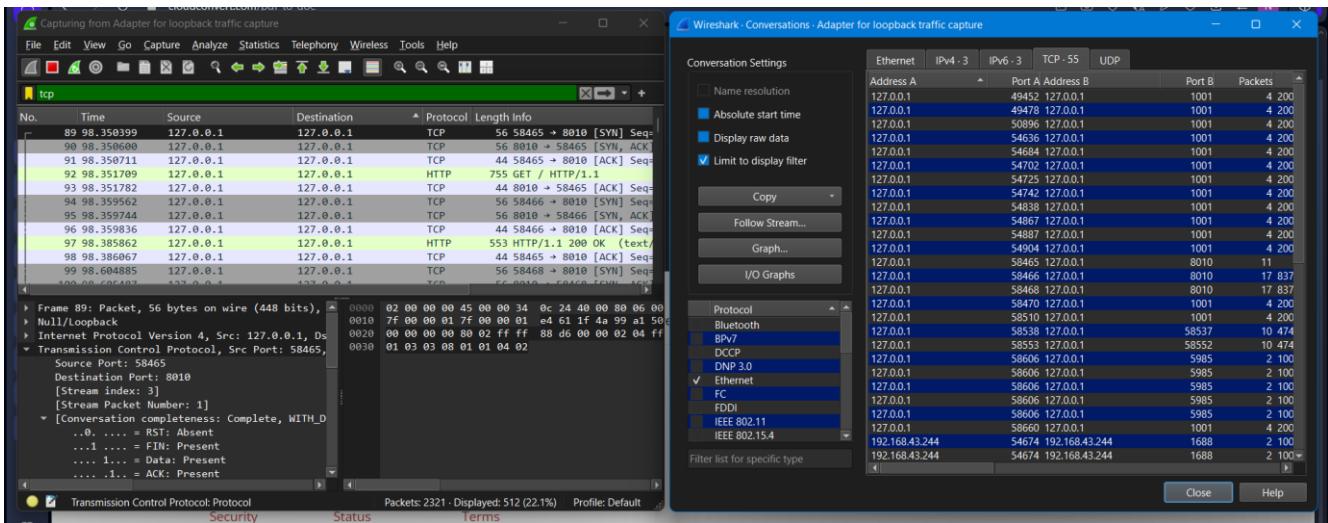
3.2. Questões

1. Quantos pedidos HTTP foram efectuados?



R: Foram efectuados 10 pedidos HTTP.

2. Quantas ligações TCP foram utilizadas?



R: Foram utilizadas 56 ligações TCP.

3. Qual a versão do protocolo HTTP utilizada?

R: A versão do protocolo usado é **HTTP/1.1**

4. Quais os cabeçalhos utilizados para manter uma ligação?



The screenshot shows the Wireshark interface with a selected stream. The ASCII dump pane displays the initial HTTP request and the server's response. The request is:

```
GET /19cb9892-fe64-469c-abd2-2cd23589c726 HTTP/1.1
Sec-WebSocket-Version: 13
Sec-WebSocket-Key: tiwHab2/fy86HyIpb1/lUg==
Connection: Upgrade
Upgrade: websocket
Host: 127.0.0.1:58741
```

The response is:

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: +6QgXe1cbvKLVynWMw6zP/dF94Y=
```

Below the ASCII dump, the status bar indicates "Packet 11961, 1 client pkt(s), 1 server pkt(s), 1 turn(s). Click to select."

R: Os cabeçalhos utilizados para manter a ligação foram:

Connection: Upgrade e Upgrade: websocket, que indicam a transição para o protocolo WebSocket e permitem manter a ligação TCP ativa.

5. Quanto tempo tenta o cliente manter a ligação?

R: O cliente não especifica um tempo através de Keep-Alive. Como a conexão foi atualizada para WebSocket (Connection: Upgrade / Upgrade: websocket), a ligação é mantida indefinidamente até ser encerrada por cliente ou servidor.

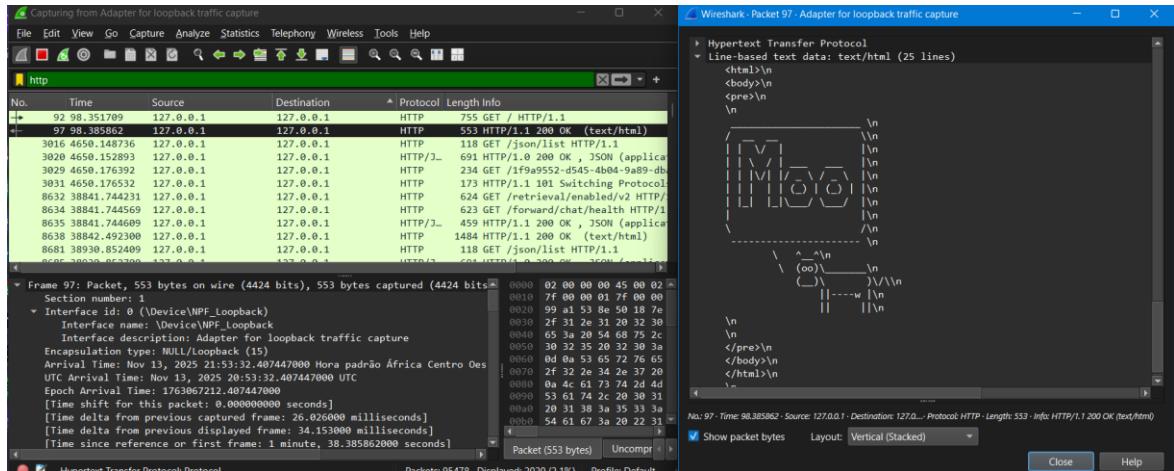
6. Quanto tempo permite o servidor que a ligação permaneça aberta sem ser utilizada?

The screenshot shows the Wireshark interface with a selected stream. The packet list pane shows a series of TCP packets between 127.0.0.1 and 127.0.0.1. The details pane shows the first few packets in detail, including the source port (58741), destination port (55973), sequence number (403522), and acknowledgment number (244). The bytes pane shows the raw hex and ASCII data for the selected packet.

R: O servidor permite que a ligação permaneça aberta aproximadamente 32 segundos sem qualquer atividade antes de a encerrar.



7. Qual o código e mensagem de resposta enviados pelo servidor em resposta ao segundo pedido do ficheiro <http://localhost:8080> (primeira utilização do botão reload) ?



R: O código de resposta é 200 e a mensagem de resposta é OK. Esta informação é encontrada na linha de Status do pacote de resposta HTTP (Frame 97).

8. Após a espera de 2 minutos, o servidor respondeu com “304 Not Modified”, porque...

R: Ao responder 304 Not Modified, o servidor não reenvia o corpo do ficheiro (economizando largura de banda), mas apenas instrui o navegador a carregar a página/ficheiro a partir do seu cache local.

4. Ficheiro de grandes dimensões

4.1. Procedimento

- Coloque o ficheiro jctest.docx no servidor mesmo path do ficheiro HTML

4.2. Observação

- Observe o grande número de mensagens enviadas pelo servidor. O tamanho do objecto a enviar exige o envio de várias mensagens por parte do servidor.

4.3. Questões

- Quantas das mensagens enviadas pelo servidor contém cabeçalhos HTTP?

R: Apenas 1 das mensagens enviadas pelo servidor contém os cabeçalhos HTTP.

5. Carregamento de uma página com imagens

5.1. Procedimento

- Adiciona duas imagens no ficheiro HTML
- Aceder a <http://localhost:8080> (repare na barra no fim)



5.2. Questões

- Quantos pedidos HTTP foram efectuados?
R: **3 pedidos HTTP.** O navegador faz um pedido para a página HTML principal e, em seguida, faz um pedido adicional para cada uma das **2 imagens** que encontra dentro do código HTML.
- A página é constituída por quantos objectos?
R: **3 objetos.** A página é constituída por **1 objeto base** (o ficheiro HTML) + **2 objetos** embutidos (as duas imagens).



6. Códigos de erro

6.1. Procedimento

- Crie um subdirectorio no servidor com o nome download e move o ficheiro jctest.docx para esse directório.
- Limpar a cache do browser
- Aceda o ficheiro jctest.docx usando a mesma URL do ponto 4
- Aceda o ficheiro jctest.docx com a URL correcta

The screenshot shows a Wireshark capture window titled "Capturing from Adapter for loopback traffic capture". The "http" tab is selected. The list pane displays 101933 total packets, with 2046 displayed (2.0% of the total). The table shows the following data:

Index	Time	Source	Destination	Protocol	Length	Info
1	...	127.0.0.1	127.0.0.1	HTTP	357	HTTP/1.1 200 OK (image/jpeg)
2	...	127.0.0.1	127.0.0.1	HTTP	740	GET /jctest.docx HTTP/1.1
3	...	127.0.0.1	127.0.0.1	HTTP	545	HTTP/1.1 404 Not Found (text/html)
4	...	127.0.0.1	127.0.0.1	HTTP	749	GET /download/jctest.docx HTTP/1.1
5	...	127.0.0.1	127.0.0.1	HTTP	554	HTTP/1.1 404 Not Found (text/html)
6	...	127.0.0.1	127.0.0.1	HTTP	775	GET /download/jctest.docx HTTP/1.1
7	...	127.0.0.1	127.0.0.1	HTTP	554	HTTP/1.1 404 Not Found (text/html)
8	...	127.0.0.1	127.0.0.1	HTTP	749	GET /download/jctest.docx HTTP/1.1
9	...	127.0.0.1	127.0.0.1	HTTP	553	HTTP/1.1 404 Not Found (text/html)
10	...	127.0.0.1	127.0.0.1	HTTP	775	GET /download/jctest.docx HTTP/1.1
11	...	127.0.0.1	127.0.0.1	HTTP	1035	HTTP/1.1 200 OK (application/vnd.openxmlformats-officedocume...)

The details pane shows the raw request and response data. The request is:

```
Upgrade-Insecure-Requests: 1\r\nUser-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.5778.90 Safari/537.36\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\nSec-Fetch-Site: none\r\nSec-Fetch-Mode: navigate\r\nSec-Fetch-User: ?1\r\nSec-Fetch-Dest: document\r\nAccept-Encoding: gzip, deflate, br, zstd\r\nAccept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7\r\n\r\n[Response in frame: 97]\r\n[Full request URI: http://localhost:8010/]
```

The bytes pane shows the hex and ASCII representation of the captured data.

6.2. Questões

- Quais os códigos de estado/erro enviados nas várias respostas?

R:

- **301 Moved Permanently** (Movido Permanentemente) - O servidor envia este código para indicar que o ficheiro `jctest.docx` **mudou de endereço de forma definitiva** (do path antigo para `/download/jctest.docx`).
- **200 OK** - O servidor envia este código após o navegador seguir o novo caminho (`/download/jctest.docx`) para indicar que o **ficheiro foi encontrado** no seu novo local e está a ser transferido com sucesso.
- **404 Not Found** (Não Encontrado) - Este código de erro **pode ser enviado** se o navegador tentar aceder ao ficheiro pelo **caminho antigo** após o 301 não ser aplicado corretamente, ou se o servidor for configurado para retornar um erro em vez de um 301. No entanto, o 301 é a resposta esperada.



- Qual a reacção do browser ao receber uma resposta com o código 301?

R: Ao receber uma resposta com o código de estado **301 Moved Permanently**, a reação padrão e automática do navegador é:

1. **Ler o Cabeçalho Location:** O navegador inspeciona os cabeçalhos HTTP da resposta 301. O servidor inclui obrigatoriamente um cabeçalho chamado **Location** que contém o novo URL do ficheiro (neste caso, `http://localhost:8080/download/jctest.docx`).
2. **Redirecionamento Automático:** O navegador **cancela a ligação atual** (a que resultou no 301) e, de imediato, **cria um novo pedido HTTP GET** direcionado ao URL especificado no cabeçalho Location.
3. **Atualização de Cache:** Como o 301 é um redirecionamento **permanente**, o navegador armazena esta nova localização em **cache**. Na próxima vez que tentar aceder ao URL antigo, ele irá diretamente para o novo URL, sem sequer perguntar ao servidor.

Em resumo, o código 301 faz com que o *browser* se **redirecione automaticamente e inicie um novo pedido** no novo endereço do recurso.

7. Programação com Socket

7.1. Procedimento

- Revise os conceitos fundamentais com exemplos práticos:
<https://docs.python.org/3/howto/sockets.html> .
- Modifique a infraestrutura para adicionar dois clientes. Assim, teremos um servidor e dois clientes.

7.2. Desafio

- Crie um servidor UDP que espera conexões de cliente e realiza o seguinte: (1) O cliente solicita adesão, enviando tuplo <IP, PortTCP>, ao servidor. (2) O servidor gera uma SeqNum, SeqNum é um identificador único (gerado sequencialmente) para cliente, armazena o tuplo <IP, PortTCP, SeqNum> do cliente e envia para o cliente o tuplo <Status, SeqNum>, em que o Status pode ser OK ou NOK e o SeqNum pode ser respectivamente o numero de sequência ou -1, enviar o tuplo <IP, PortTCP, SeqNum> de cada cliente conectado a rede e termina a ligação.
- Os clientes conectam-se via TCP usando os tuplos recebidos do servidor via UDP para trocarem mensagens de texto simple.

R: O desafio encontra-se no github

(<https://github.com/NzolaKiampava/RC1Projectos>), e na pasta Desafios subpasta **UDP_TCP**.



8. Problema prático

1. Considere uma rede de comutação de circuitos que usa TDM com S slots. O tempo de estabelecimento de um circuito é T s, o débito das ligações é R b/s. Qual o débito alcançado na transmissão de um ficheiro de L Bytes?

R: Refect = $8L/(T + (8L/R))$ [b/s]



2. Numa rede de comutação de pacotes, quais os componentes do atraso que dependem do tamanho do pacote?

Os componentes do atraso que dependem do tamanho do pacote são:

- **Atraso de transmissão:** $d_{trans} = L/R$
- **Atraso de enfileiramento**

Os atrasos de propagação e de processamento **não dependem** do tamanho do pacote.

3. Considere que acede a uma página web constituída pelo HTML e uma imagem. Qual o tipo de ligação HTTP que deve ser usado para garantir o melhor desempenho?

R: Para garantir o **melhor desempenho** ao carregar uma página web com HTML e uma imagem, deve-se usar:

HTTP/1.1 com conexões persistentes ou melhor ainda, HTTP/2.

4. Uma rede local fornece conectividade a várias estações de trabalho. O router que liga essa rede à Internet usa uma ligação de 1Mbps. A rede local tem um débito de 1Gbps. Para aumentar o desempenho no acesso à WWW, é instalado um proxy web. Em média, cada página web tem 1MB. O proxy web tem um hit ratio de 50%.

- a) Qual o tempo médio de acesso a uma página antes de ser instalado o proxy?

R: **Tempo médio antes do proxy: 8 s**

- b) Qual o tempo médio de acesso a uma página após a instalação do proxy?

R: **Tempo médio após o proxy: 4 s**

5. Considere uma aplicação de partilha de ficheiros P2P. Um ficheiro de L Byte é inicialmente partilhado por um peer. Existem N peers na rede, cada um com U b/s de capacidade de upload e D b/s de download. Qual o tempo mínimo para partilhar o ficheiro?

R: **Tmin = max (8L/D, 8L/U)**

6. Considere o detalhe de um pacote capturado no wireshark.

```
Internet Protocol Version 6, Src: 2001:8a0:7409:ac01:2015:a80e:8916:b0c2 (2001:8a0:7409:ac01:2015:a80e:8916:b0c2), Dst: 2001:8a0:2104:ff:213:13:146:138 (2
Transmission Control Protocol, Src Port: 53894 (53894), Dst Port: 80 (80), Seq: 1740208792, Ack: 4286080053, Len: 746
Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
Host: www.sapo.pt\r\n
Connection: keep-alive\r\n
Cache-Control: max-age=0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.90 Safari/537.36\r\n
Accept-Encoding: gzip, deflate, sdch\r\n
Accept-Language: en-US,en;q=0.8,pt-PT;q=0.6,pt;q=0.4\r\n
[truncated]Cookie: uu=ac3dd41dc03f4d7794a670f73b41430b; _shp_refresh=1445817600000; _swa_lpvt=1445885614286; _swa_uv=938242371445885614; _swa_huv=55388
\r\n
```

- a) Qual a versão do protocolo de rede utilizada?



R: A versão do protocolo de rede é o **IPv6** (Internet Protocol Version 6).

- b) Qual a porta (camada de transporte) utilizado pelo servidor?

R: PORTA 80

- c) Qual o URL introduzido no browser?

R: <http://www.sapo.pt/>

- d) É a primeira vez que o utilizador visita este site (usando este browser)?

R: **NÃO**.

- e) São usadas ligações persistentes?

R: **Sim**. As ligações persistentes (*Keep-Alive*) são usadas.

Bom trabalho!