

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?
2. Qual o endereço IP do servidor?
3. Quantos pedidos HTTP foram efectuados?
4. Qual o código e mensagem de resposta enviados pelo servidor em resposta ao pedido do ficheiro transferido ?
5. Quais as linguagens indicadas pelo seu browser como preferidas? (faça um print da tela)
6. Qual o nome do software utilizado como servidor HTTP?
7. Qual o tamanho da página apresentada pelo browser?

### 3. Mecanismo de cache

#### 3.1. Procedimento

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

#### 3.2. Questões

1. Quantos pedidos HTTP foram efectuados?
2. Quantas ligações TCP foram utilizadas?
3. Qual a versão do protocolo HTTP utilizada?
4. Quais os cabeçalhos utilizados para manter uma ligação?
5. Quanto tempo tenta o cliente manter a ligação?
6. Quanto tempo permite o servidor que a ligação permaneça aberta sem ser utilizada?
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) ?
8. Após a espera de 2 minutos, o servidor respondeu com “304 Not Modified”, porque...

### 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?

### 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?
- Apágina é constituída por quantos objectos?

## 6. Códigos de erro

### 6.1. Procedimento

- Crie um subdirectório no servidor com o nome download e mova 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

### 6.2. Questões

- Quais os códigos de estado/erro enviados nas várias respostas?
- Qual a reacção do browser ao receber uma resposta com o código 301?

## 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.

## 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?



2. Numa rede de comutação de pacotes, quais os componentes do atraso que 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?
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?
  - b) Qual o tempo médio de acesso a uma página após a instalação do proxy?
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?
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 (2001:8a0:2104:ff:213:13:146:138)
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
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?
- b) Qual a porta (camada de transporte) utilizado pelo servidor?
- c) Qual o URL introduzido no browser?
- d) É a primeira vez que o utilizador visita este site (usando este browser)?
- e) São usadas ligações persistentes?

**Bom trabalho!**