



FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE NOVA DE LISBOA

Departamento de Informática

Licenciatura em Engenharia Informática  
2º TESTE – Redes de Computadores  
1º Semestre, 2015/2016 (9/dezembro/2015)

REF-A-RC-T2-2015-2016

Leia com atenção cada questão antes de responder. A interpretação do enunciado é da inteira responsabilidade do estudante.

A duração do teste é 1 hora e 30 minutos com 15 minutos de tolerância.

Não pode usar calculadoras, *tablets*, computadores ou telemóveis.

O enunciado contém 10 questões e 6 páginas, incluindo a de rosto, que devem ser entregues agrafadas como resposta ao teste.

Pode usar uma folha A4 manuscrita com anotações suas feitas durante o seu estudo.

As respostas erradas em questões de resposta múltipla descontam, podendo resultar numa cotação nula com várias respostas certas e erradas.

NOME: \_\_\_\_\_ Nº Aluno: \_\_\_\_\_

**Copie no fim para este quadro as suas respostas – só estas respostas serão consideradas para a sua classificação, exceto as indicadas para responder numa caixa**

1 a)

1 b)

1 c)

2 a)

2 b)

3)

4 a)

5)

7)

9)

Responda às questões 4 b), 6), 8) e 10) nas caixas de texto a seguir às perguntas.

1) Um cliente HTTP (por exemplo um *Browser*) acede a uma página HTML num servidor. Depois de obter essa página, o cliente deduz que a mesma tem 4 imagens e que as mesmas devem ser obtidas igualmente, a partir desse mesmo servidor, para mostrar o conteúdo total ao utilizador. O tempo de trânsito ida e volta (RTT) entre o cliente e o servidor é de 50 milissegundos. O cliente não tem nenhuma conexão aberta para o servidor antes de começar a aceder à página, mas já conhece o endereço IP do servidor. O tempo necessário para transmitir os pacotes com o ou os comandos, a página ou as imagens são negligenciáveis e cada mensagem HTTP (*Request ou Reply*) é transmitida de uma vez dado caber sempre num só segmento TCP.

Em cada uma das questões abaixo selecione a opção contornando o valor que mais se aproxima da resposta certa.

a) Qual o tempo necessário (**em milissegundos**) para o cliente obter a página e as imagens usando apenas a versão 1.0 do protocolo HTTP e SEM usar conexões paralelas?

100 200 300 400 500 600 700 800 900 1000 1100 1200 1300  
1400 1500 1600 1700 1800 1900 2000

b) Qual o tempo necessário (**em milissegundos**) para o cliente obter a página com e as imagens usando a versão 1.1 do protocolo HTTP SEM usar *pipelining* e SEM usar conexões paralelas?

100 200 300 400 500 600 700 800 900 1000 1100 1200 1300  
1400 1500 1600 1700 1800 1900 2000

c) Qual o menor tempo necessário (**em milissegundos**) para o cliente obter a página e as imagens usando o protocolo HTTP (qualquer versão e com ou sem *pipelining*) sabendo que o cliente abre uma conexão para obter a página inicial e a seguir a essa obtenção, mais 3 ficando então com 4 conexões paralelas?

100 200 300 400 500 600 700 800 900 1000 1100 1200 1300  
1400 1500 1600 1700 1800 1900 2000

2) Um servidor acessível via o serviço HTTP distribui segmentos de filmes com a dimensão, sempre constante, de 128 x 1400 bytes  $\approx$  180.000 bytes. O servidor aceita pedidos em HTTP 1.0 ou em HTTP 1.1 mas em qualquer dos casos responde sempre com uma mensagem contendo as seguintes campos no cabeçalho da resposta: "Accept-Ranges: none" e "Connection: close".

Todas as conexões TCP do servidor usam um MSS (*Maximum Segment Size*) de 1460 bytes pelo que pode considerar que em cada segmento TCP se transferem geralmente 1400 bytes de filme. As transferências nunca estão limitadas pelo débito de nenhum canal na rede pois os mesmos são todos de muito alto débito. A dimensão dos *buffers* do TCP é sempre de 256 K bytes no cliente e no servidor. As conexões começam sempre a transmitir dados com o valor da *ConWnd* (*Congestion Window*) igual a um segmento.

Em cada uma das questões abaixo selecione a opção que mais se aproxima da resposta certa.

a) Quanto tempo (**em milissegundos**) leva um cliente a obter cada um dos segmentos de filme usando o protocolo HTTP 1.0, sem conexões paralelas, sabendo que o RTT do cliente ao servidor é 25 ms?

100 200 300 400 500 600 700 800 900 1000 1100 1200 1300  
1400 1500 1600 1700 1800 1900 2000 milissegundos

b) Quanto tempo (**em milissegundos**) leva um cliente a obter um cada um dos segmentos de filme nas condições da **a)** mas usando o protocolo HTTP 1.1?

100 200 300 400 500 600 700 800 900 1000 1100 1200 1300  
1400 1500 1600 1700 1800 1900 2000

3) Dois utilizadores A e B estão a realizar uma conversa telefónica interativa, sem imagem, sobre a Internet, usando o protocolo RTP/UDP (*Real Time Transmission Protocol* sobre UDP). Verifica-se que a ligação da rede entre A e B apresenta as seguintes características de qualidade de serviço: taxa de perda de pacotes de 1%, débito extremo a extremo variando entre 1 Mbps e 1,5 Mbps, tempo de trânsito de 98% dos pacotes inferior a 100 ms mas os restantes 2% podem ter um tempo de trânsito até 600 ms. Indique qual das seguintes soluções A e B podem usar para manter a sua conversa telefónica interativa (escolha uma só opção):

- 1 – CODEC MPEG para transmissão de voz e imagem com uma resolução média de 1 Mbps e um *playout delay* de 1 segundo
- 2 – CODEC PCM (Pulse Coded Modulation) para transmissão de voz a 64 Kbps e um *playout delay* de 1 segundo
- 3 – CODEC PCM (Pulse Coded Modulation) para transmissão de voz a 64 Kbps e um *playout delay* de 50 milissegundos
- 4 – CODEC MPEG para transmissão de voz e imagem com uma resolução média de 1 Mbps e um *playout delay* de 600 milissegundos

4) O servidor DNS primário do domínio “paradise.com” tem na sua tabela, há mais de um ano, a indicação de que o servidor de nome “true.paradise.com” tem o endereço IP 100.100.100.100 com um TTL associado de 1000 segundos. Num determinado momento foi feita uma atualização dessa informação e o mesmo servidor DNS passou a indicar que o endereço IP associado ao nome “true.paradise.com” é agora 100.100.100.200 mas com o TTL de 2000 segundos.

a) Quanto tempo leva a que qualquer interrogação ao DNS sobre o nome “true.paradise.com” feita em qualquer ponto da Internet devolva como resposta o novo IP (100.100.100.200) e não o antigo (100.100.100.100)? Escolha apenas uma opção.

- |                                     |                   |                           |
|-------------------------------------|-------------------|---------------------------|
| 1 – nunca                           | 2 – imediatamente | 3 – 3000 segundos         |
| 4 – 2000 segundos                   | 5 – 1000 segundos | 6 – mais de 3000 segundos |
| 7 – geralmente cerca de 10 segundos |                   |                           |

b) Você é o gestor do servidor e do domínio acima. Para diminuir a probabilidade de que na altura em que procede à troca do endereço, se corra o risco de alguns clientes não conseguirem aceder ao serviço, na véspera resolve mudar um parâmetro DNS associado ao nome do servidor. Que parâmetro é esse e que valor ele deve tomar? Justifique a sua resposta.

5) Pretende-se utilizar o método de controlo de acesso ao meio CSMA/CD, para controlar o acesso a um canal multiponto *half-duplex*, baseado em difusão, do tipo Ethernet clássica, com a velocidade de transmissão de **10 Mbps** e um tempo máximo de propagação de extremo a extremo de **50 micro segundos**. Num canal desse tipo define-se *collision slot* como sendo o período durante o qual o emissor tem de emitir continuamente bits e ao fim do qual este sabe que se não houve colisão até então, já não existirá nenhuma colisão durante a transmissão corrente.

Indique qual a **dimensão em bits** do menor *frame* Ethernet que tem de ser emitido nesse canal. (Escolha das opções a seguir a que está mais próxima da resposta certa):

100	200	300	400	500	600	700	800	900	1000	1100
1200	1300	1400	1500	1600	1700	1800	1900	2000	2100	2200
2400	2500	2600								

6) Numa rede IP com suporte de difusão (*broadcasting*) de *frames* Ethernet existe um servidor DHCP, o servidor DHCP1, que afeta endereços IP no prefixo IP 100.100.100.0/24 aos computadores que se ligam a essa rede, com um *lease time* (duração máxima da afetação) de 1000 segundos. O servidor DHCP avariou-se e foi substituído imediatamente por um novo, o servidor DHCP2, que não conhece, no momento do arranque, quais os endereços afetados pelo servidor DHCP1 que ainda estão a ser usados.

DHCP2 afeta endereços no mesmo prefixo que DHCP1 e tem agora o problema de quando recebe um novo pedido de endereço, decidir que endereço afetar. DHCP2 poderia correr o risco de afetar endereços aleatoriamente na esperança de que não houvessem colisões com endereços afetados por DHCP1 que ainda estivessem a ser usados. Poderia também pedir a todos os computadores que devolvessem os endereços em uso e pedissem um novo, mas o protocolo DHCP não suporta essa hipótese.

Que pode fazer DHCP2 para diminuir a probabilidade de afetar endereços que ainda estejam a ser usados por computadores ligados à sua rede? Justifique a sua resposta.

7) Considere o algoritmo de encaminhamento de pacotes por inundação (*flooding*). Qual ou quais das seguintes afirmações são verdadeiras no contexto desse algoritmo de encaminhamento?

- 1 – Uma maneira de suprimir a introdução de pacotes duplicados através de inundação numa rede estruturada em malha (com ciclos) é usar um TTL adequado.
- 2 – Numa rede completamente conexa (cada nó tem pelo menos um caminho para os outros) o algoritmo de inundação assegura que todas as mensagens chegam a todos os nós
- 3 – O algoritmo de encaminhamento por inundação é muito ineficiente e por isso nunca é usado na prática
- 4 – Caso a topologia da rede não tenha ciclos, a inundação é um processo razoavelmente eficiente de implementar o encaminhamento porque nunca enviará mensagens em duplicado nem inúteis.
- 5 – O algoritmo de inundação implementa naturalmente a difusão de mensagens para todas as interfaces ligadas à rede.
- 6 – Numa rede estruturada em árvore, a inundação encaminha sempre pelo melhor caminho.

8) Um computador tem uma interface Ethernet com o endereço IP 100.100.0.1/24 e recebe um pacote com endereço de destino 100.100.0.100. A tabela de ARP do computador tem o seguinte conteúdo:

100.100.0.1	00:00:AA:BB:00:10
100.100.0.4	00:00:AA:BB:00:02
100.100.0.10	00:00:AA:BB:00:01
100.100.0.67	00:00:AA:BB:00:55

Preencha na 2ª coluna da seguinte tabela o tipo ou conteúdo e qual o quais *frames* Ethernet que o computador deve enviar para encaminhar o pacote IP até ao seu destino (podem ser 1, 2 ou 3 *frames* distintos)

Tipo ou conteúdo do <i>frame</i> 1:	
Endereço MAC origem do <i>frame</i> 1:	
Endereço MAC destino do <i>frame</i> 1:	
Tipo ou conteúdo do <i>frame</i> 2	
Endereço MAC origem do <i>frame</i> 2:	
Endereço MAC destino do <i>frame</i> 2:	
Tipo ou conteúdo do <i>frame</i> 3:	
Endereço MAC origem do <i>frame</i> 3:	
Endereço MAC destino do <i>frame</i> 3:	

9) Na seguinte tabela quais dos seguintes endereços IP estão incluídos no prefixo 100.100.0.192/26?

1	100.100.0.193
2	100.100.0.254
3	100.100.0.1
4	100.100.1.110
5	100.100.0.100

**10)** No quadro do encaminhamento pela técnica conhecida como *link state routing*, um comutador recebeu de um vizinho a que está diretamente ligado um *link state announcement* (LSA) que o vizinho lhe enviou. Essa mensagem, é um objeto da classe LSA, que tem o seguinte construtor:

LSA (String comutador, String vizinho, long sequência, byte[] data)

Em que “comutador” é o identificador do comutador que cria o LSA; “vizinho” é o comutador que transmitiu o LSA; “sequência” é o número de sequência (único e sempre crescente no contexto de cada comutador) do LSA; e “data” contém a informação de estado que “comutador” quer transmitir aos outros.

A mensagem (objeto LSA) pode ser manipulada pelos seguintes métodos:

String comutador()	devolve o identificador do <i>router</i> que gerou inicialmente este LSA
String vizinho()	devolve o identificador do <i>router</i> que enviou este LSA
long sequência()	devolve o número de sequência do LSA
void flood()	faz a inundação ( <i>flooding</i> ) do LSA para os outros comutadores vizinhos, exceto o vizinho que o enviou ao comutador que invoca <i>flood</i> .
void enviarACK()	envia um ACK ao comutador vizinho que o enviou ao comutador que invoca enviarACK.
void enviarNACK(LSA lsa)	envia um NACK ao comutador vizinho que o enviou ao comutador que invoca enviarNACK, assim como a versão mais recente do mesmo LSA

Em cada comutador, a tabela db, da classe LSADaBase, contém o estado da rede, isto é, o conjunto de LSAs mais recentes recebidos. Esta tabela pode ser manipulada pelos seguintes métodos:

LSA get(String c)	devolve o mais recente LSA gerado pelo comutador c ou <i>null</i> se não existe nenhum enviado por c
void put(String c, LSA lsa)	insere lsa na tabela associando-o ao comutador c; note que se tem de assegurar sempre que c == lsa.comutador().

Escreva o código executado por um comutador quando recebe um LSA de um vizinho de forma a assegurar a difusão fiável desse LSA por todos os comutadores da rede.

```
void processarLSArecebido (LSA lsa, String vizinho) { // chegou o LSA lsa de vizinho, processá-lo:
```

```
}
```