

2º TESTE – Redes de Computadores

1º Semestre, 2018/2019 (19/dezembro/2018)

1 - **VERSÃO A)** O servidor DNS primário do domínio **utopia.com** tem o seguinte registo:

verdade.utopia.com. 800 IN A 100.100.100.100

No momento t_1 , esse registo foi atualizado para:

verdade.utopia.com. 400 IN A 100.100.100.200

A partir de que momento, garantidamente, a consulta do mesmo RR (resource record) devolve o novo endereço IP, seja qual for o servidor a que essa consulta é feita? Escolha apenas uma das seguintes opções:

- a) t_1+400 segundos
- b) t_1+800 milissegundos
- c) t_1+800 segundos
- d) t_1+1200 segundos
- e) $t_1+10000$ segundos
- f) t_1+0 segundos
- g) nenhuma destas

2 - **VERSÃO A)** O tempo médio necessário para que um computador, ligado à rede da sua faculdade, execute uma interação *request / reply* com o servidor DNS *caching only* dessa rede, é desprezável. Mas o tempo médio necessário para que esse servidor execute a mesma interação com um servidor DNS externo, é em média de 100 ms. Na *cache* do servidor DNS *caching only* da sua faculdade existe a seguinte informação:

bestshop.com	NS	dns1.bestshop.com
bestshop.com	NS	dns2.bestshop.com
dns1.bestshop.com	A	88.10.1.24
dns2.bestshop.com	A	88.10.2.56

mas não existe nenhuma informação sobre o domínio “.com”, nem mais informação sobre o domínio “bestshop.com”. Qual o tempo mínimo necessário para um computador ligado à rede da sua faculdade, e que usa o servidor *caching only* da mesma, obter o endereço IP de www.bestshop.com?

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

3 - VERSÃO A) 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 25 milissegundos.

Hipóteses: 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 de transmissão dos pacotes com o ou os comandos, a página ou as imagens são negligenciáveis e qualquer mensagem HTTP (*Request ou Reply*) é transmitida de uma só vez dado caber sempre num só segmento TCP.

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

- a) Qual o tempo mínimo 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?

25 50 75 100 125 150 175 200 225 250

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

25 50 75 100 125 150 175 200 225 250

- c) Qual o tempo mínimo 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 logo 4 conexões paralelas para aceder ao servidor?

25 50 75 100 125 150 175 200 225 250

4 - VERSÃO A) Um servidor acessível via o serviço HTTP distribui ficheiros. Um cliente pretende obter um ou mais ficheiros todos com 1.000.000 bytes.

Todas as conexões TCP do servidor usam um MSS (*Maximum Segment Size*) tal que em cada segmento TCP se transferem geralmente 1000 bytes do ficheiro.

Hipóteses: as transferências nunca estão limitadas pelo débito de nenhum canal. As janelas e os *buffers* máximos do TCP são sempre de 500 x MSS no cliente e no servidor. As conexões começam sempre a transmitir dados com o valor da *ConWnd* (*Congestion Window*) igual a um MSS.

Um cliente pretende obter 100 desses ficheiros.

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 um dos ficheiros usando o protocolo HTTP 1.0, sabendo que que o RTT do cliente ao servidor é 100 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 o último ficheiro, depois de ter obtido 99 através do protocolo HTTP 1.1, sabendo que que o RTT do cliente ao servidor é 100 ms?

100 200 300 400 500 600 700 800 900 1000 1100 1200 1300

1400 1500 1600 1700 1800 1900 2000

5 - VERSÃO A) Um conjunto de computadores com interfaces IEEE 802.3 (com fios) estão interligados através de um equipamento central numa configuração em estrela. Algumas das interfaces têm o débito de 10 Mbps e outras de 100 Mbps, mas todas comunicam perfeitamente entre si. Qual das seguintes respostas está certa?

- a) O equipamento central é um *hub* Ethernet (grosso modo um repetidor de sinal)
- b) O equipamento central é um *switch* Ethernet (um comutador de pacotes ou *frames*)

6 - VERSÃO A) Um protocolo de controlo de acesso a um meio de transmissão partilhado multi-ponto do tipo CSMA/CA usado num canal sem fios (802.11), apresenta um *overhead* superior ao de um do tipo CSMA/CD usado num canal com fios (802.3), mesmo quando ambos usam o mesmo débito de transmissão. Num contexto em que muitas interfaces estão ligadas ao mesmo canal, quais dos seguintes fatores contribuem mais significativamente para essa diferença?

- a) Detecção da portadora (*Carrier Sense*)
- b) Os mecanismos de reserva do canal sem fios (RTS, CTS).
- c) As diferenças no cabeçalho dos *frames* entre 802.3 e 802.11
- d) O mecanismo de arbitragem inicial do acesso ao canal.
- e) A ausência de mecanismo de detecção das colisões na versão 802.11
- f) O tempo de transmissão dos *frames*.

7 - VERSÃO A) Considere o algoritmo de encaminhamento de pacotes por inundação (*flooding*) numa rede em que todos os canais são bi-direcionais. Qual ou quais das seguintes afirmações são verdadeiras?

- a) O algoritmo de inundação implementa a difusão de mensagens para todas as interfaces (de saída) ligadas à rede.
- b) O algoritmo de encaminhamento por inundação é muito ineficiente e por isso nunca é usado na prática
- c) Numa rede estruturada em malha (com ciclos) a forma de suprimir a introdução de pacotes duplicados pela inundação é usar um TTL adequado.
- d) Numa rede estruturada em árvore, a inundação encaminha sempre pelo melhor caminho.
- e) Numa rede completamente conexa (cada nó tem pelo menos um caminho para os outros) o algoritmo de inundação com TTL finito assegura sempre que todas as mensagens chegam a todos os nós.
- f) Caso a topologia da rede não tenha ciclos, a inundação é um processo aceitável de implementar o encaminhamento porque nunca enviará mensagens em duplicado nem inúteis.