

Universidade do Minho - Dep.^{to} Informática
Licenciatura em Engenharia Informática - 1º Semestre, 2022/2023
COMUNICAÇÕES POR COMPUTADOR
Teste Escrito – 12 janeiro
120 Minutos

PIN: 9408-3938-1000	PASSWORD: 05JH73TGG
NÚMERO :	
NOME :	
<div style="text-align: center;">Instruções</div> <ul style="list-style-type: none">• Intervenções consideradas desnecessárias durante o teste serão penalizadas com dedução de pontos à classificação.• A correção do teste será feita automaticamente tendo em conta as respostas assinaladas no formulário digital no computador.• Este documento é também um enunciado do teste e é também um formulário de <i>backup</i> onde deve indicar as suas respostas.• Para aceder ao formulário online disponível no computador à sua frente introduza o PIN e a PASSWORD indicados no cabeçalho acima e depois a sua identificação (número de aluno e nome completo).• O teste está dividido em 4 partes distintas, cada uma valendo 5 valores.• Cada parte tem 2 questões com 5 afirmações cada. Pede-se, em todas as questões, para se assinalar quais as afirmações que considera verdadeiras e as que considera falsas.• Pode ignorar as questões que não souber fazer ou as afirmações que não conseguir distinguir como verdadeiras ou falsas.• Cada afirmação bem identificada vale 0,5 pontos.• As respostas incorretas terão uma cotação que será deduzida ao total da cotação do teste. O montante a deduzir depende da gravidade da incorreção e pode ir de -0,1 a -0,4 pontos.• Inclua a sua identificação (número de aluno e nome completo) em todas as folhas.• Pode usar as páginas em branco deste documento para rascunho. Não serão fornecidas folhas adicionais para rascunho.• A duração do teste é de 120 minutos. Não será concedido qualquer tempo adicional (exceto a alunos com estatuto NEE) nem será permitida a ida ao WC (exceto a alunos com atestado médico).• Só poderá sair da sala no final dos 120 minutos.• Confirme que terminou e enviou a sua resolução digital antes de abandonar a sala.• No final, deixe este enunciado/formulário na sua secretária/mesa.	

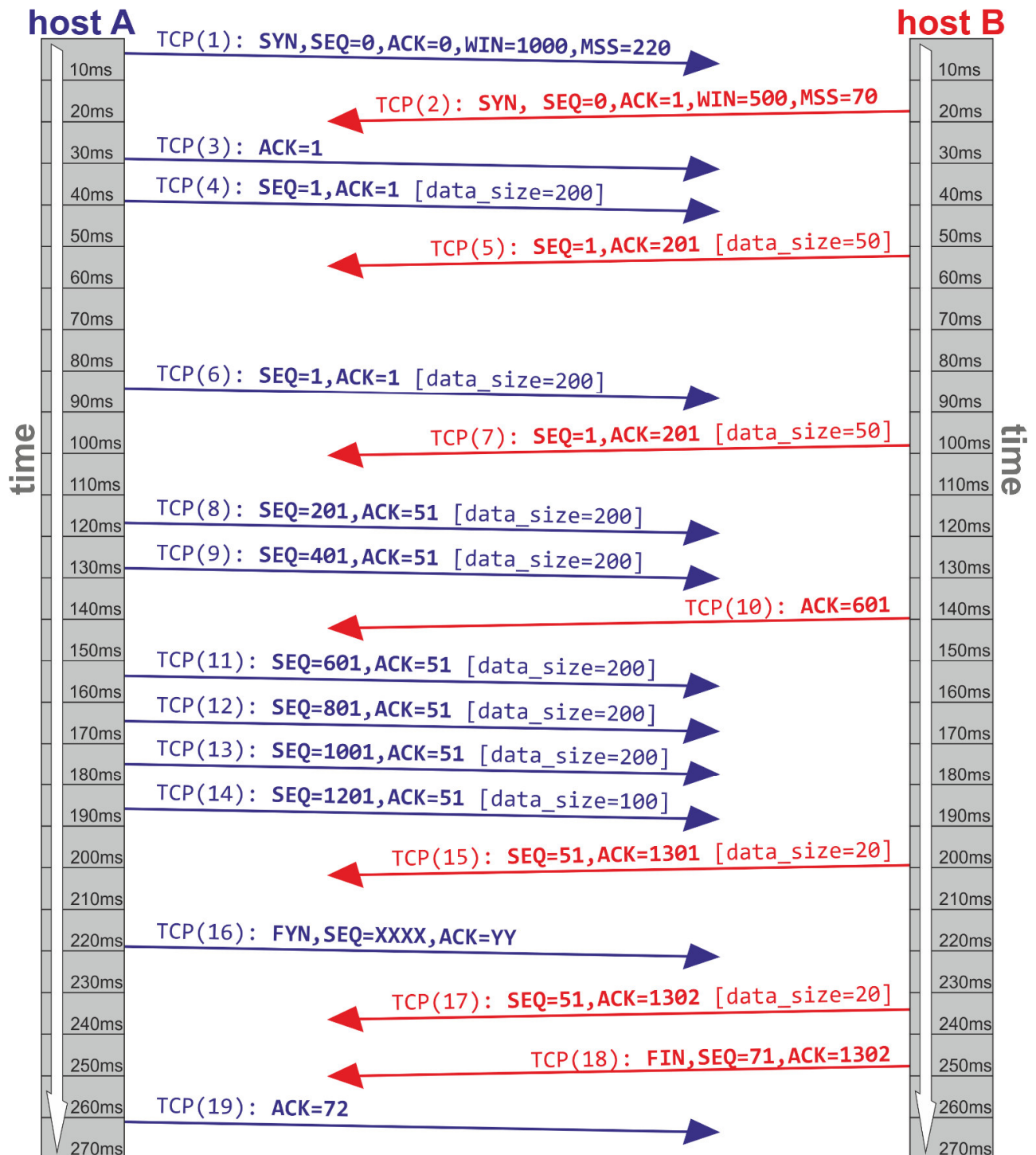
NÚMERO :**NOME :****PARTE I – Protocolos de Transporte na Internet**

1. Analise as seguintes afirmações genéricas sobre protocolos de transporte e indique (com **X**) se as considera verdadeiras (*true*) ou se as considera falsas (*false*). Se não souber não indique qualquer uma das hipóteses.

	V/T	F	
1.a)	X		Os protocolos de transporte não orientados à conexão (como o <i>User Datagram Protocol</i> – UDP) também permitem a multiplexagem e desmultiplexagem de aplicações distribuídas na Internet, tal como os protocolos orientados à conexão.
1.b)		X	Num <i>socket</i> TCP (<i>Transmission Control Protocol</i>) ativo num <i>host</i> (conexão corretamente aberta e aceite) é possível receber segmentos de dados de vários <i>hosts</i> desde que os segmentos venham de endereços IP diferentes entre si.
1.c)		X	Em aplicações de videoconferência é aconselhável usar-se um protocolo não orientado à conexão como o UDP para minimizar a perda de pacotes na comunicação.
1.d)		X	Uma aplicação Internet usando o protocolo de transporte TCP não pode implementar mecanismos próprios de controlo de fluxo da informação aplicacional, adicionais aos já existentes no TCP, apenas mecanismos adicionais de segurança.
1.e)		X	Tanto no TCP como no UDP, o tamanho do cabeçalho das respetivas unidades protocolares é de tamanho fixo, ainda que o cabeçalho do TCP seja maior e mais complexo do que o do UDP.

2. Considere o esquema da interação protocolar numa conexão TCP entre dois *hosts* (que estão em redes IP diferentes) da Figura 1. Este esquema foi deduzido através da análise dos segmentos TCP que foram observados a circular nas redes locais respetivas, ou seja, é possível garantir que os segmentos apresentados saíram dos respetivos *hosts* de origem, mas não é possível garantir que tenham chegado ao destino. Indique (com **X**) se as afirmações seguintes lhe parecem verdadeiras (*true*) ou falsas (*false*). Se não souber não indique qualquer uma das hipóteses.

	V/T	F	
2.a)	X		O <i>host</i> B aceitou os valores dos parâmetros de funcionamento da conexão (tamanho de janela e tamanho máximo de segmento) e propôs também os valores para esses parâmetros para o outro sentido da comunicação, e que foram aceites pelo <i>host</i> A.
2.b)	X		No segmento TCP(16), enviado pelo <i>host</i> A para o <i>host</i> B, o valor do número de sequência (<i>sequence number</i>) XXX deve ser igual a 1301 e o valor do número de confirmação (<i>acknowledgment number</i>) YYY deve ser igual a 51.
2.c)	X		Analisando a interação protocolar podemos concluir que o segmento identificado como TCP(5), enviado pelo <i>host</i> B, não chegou ao <i>host</i> A em tempo útil.
2.d)		X	A quantidade total de dados aplicacionais enviados pelo <i>host</i> A foi de 1600 bytes, mas a quantidade útil recebida pelo <i>host</i> B foi de 1400 bytes.
2.e)		X	Pela análise da interação protocolar podemos concluir que, na direção da comunicação do <i>host</i> B para o <i>host</i> A, é utilizado o mecanismo de controlo de congestão do TCP denominado de <i>slow start/congestion avoidance</i> .

**Notas:**

WIN=tamanho de janela, MSS=tamanho máximo de segmento.

Só são apresentados os campos/flags relevantes ou que não se alteram de segmento para segmento enviado pelo mesmo host.

Figura 1: Interação protocolar TCP no exercício 2.

NÚMERO :**NOME :****PARTE II – Serviço DNS na Internet**

3. Analise as seguintes afirmações genéricas sobre o serviço de resolução de nomes da Internet (*Domain Name Resolution* – DNS) e indique (com **X**) se as considera verdadeiras (*true*) ou se as considera falsas (*false*). Se não souber não indique qualquer uma das hipóteses.

	V/T	F	
3.a)		X	O serviço DNS da Internet é uma espécie de base de dados distribuída em que os servidores autoritativos para cada domínio são responsáveis pelo registo e manutenção da parte da base de dados referente ao seu domínio e subdomínios.
3.b)	X		A organização hierárquica dos domínios de DNS não está relacionada com a organização do endereçamento das redes IP, nem com a forma como os <i>hosts</i> , equipamentos e <i>routers</i> se organizam em topologias de rede.
3.c)	X		Os dois modos de funcionamento da pesquisa/procura de informação no DNS, i.e., modo iterativo e modo recursivo, garantem o mesmo resultado a uma determinada <i>query</i> DNS, ainda que os tempos de resposta possam ser diferentes.
3.d)	X		Uma resposta a uma <i>query</i> DNS só pode incluir entradas de DNS (<i>resource records</i>) integrais (i.e., conforme aparecem na base de dados do domínio respetivo) nos campos RESPONSE VALUES, AUTHORITIES VALUES e EXTRA VALUES.
3.e)		X	Um servidor primário dum domínio de DNS deixa de ser um servidor autoritativo para esse domínio se a sua cópia da base de dados da informação desse domínio tiver expirado completamente.

4. Uma aplicação cliente envia a *query* “sp.medium.big. A” a um servidor *resolver* (SR). Parta do princípio que todos os domínios existem e que os servidores primários são identificados/registados no DNS com o nome “sp.nome_do_dominio” (por exemplo, sp.big é o nome do servidor primário (SP) do domínio de topo big) e que os servidores secundários (SS) são identificados/registados como “ssN.nome_do_dominio” (por exemplo, ss1.big é o nome dum servidor secundário do domínio de topo big). Admita que o único servidor de topo (ST) registado no SR é identificado pelo endereço 10.10.10.10, todos os servidores funcionam no modo iterativo e nenhum tem, inicialmente, entradas úteis em *cache*. Tendo em consideração os aspetos mais importantes das primeiras ações relevantes do serviço DNS, assinale como verdadeiras (*true*) ou falsas (*false*) as afirmações:
(Nota: RV=”RESPONSE VALUES”, AV=”AUTHORITIES VALUES”, EV=”EXTRA VALUES”.)

	V/T	F	
4.a)		X	i) No início, o SR reenvia a <i>query</i> para o ST 10.10.10.10 que responde com campo RV nulo, no campo AV inclui apenas os nomes dos servidores secundários para big e no campo EV inclui os endereços IP desses servidores.
4.b)	X		ii) Depois do SR receber a resposta do ST referida em i), guarda as entradas dos campos AV e EV em <i>cache</i> e reenvia a <i>query</i> inicial para o endereço do servidor ss1.big ou dos outros endereços dos SS do domínio big recebidos.
4.c)	X		iii) O SS contactado em ii) responde ao SR com campo RV nulo, no campo AV inclui os nomes dos servidores autoritativos para medium.big e no campo EV inclui os endereços IP desses servidores.
4.d)	X		iv) O SR recebe a resposta referida em iii), guarda as entradas dos campos AV e EV em <i>cache</i> e responde à <i>query</i> inicial enviando uma resposta ao cliente pois já tem informação suficiente em <i>cache</i> para conseguir responder.
4.e)		X	Por razões de segurança e eficiência é natural e desejável que os domínios de topo (como o domínio big deste exemplo) tenham mais do que um servidor primário ativo e registado nos servidores de topo.

PARTE III – Protocolo de Transporte HTTP

5. Analise as seguintes afirmações genéricas sobre o protocolo de transporte aplicacional *Hypertext Transfer Protocol* (HTTP) e indique (com X) se as considera verdadeiras (*true*) ou falsas (*false*).

	V/T	F	
5.a)	X		O método DELETE introduzido na versão 1.1 do HTTP introduz problemas de segurança pelo que os servidores HTTP devem ter cuidados acrescidos quando recebem pedidos DELETE dos clientes.
5.b)		X	Uma das características melhoradas na evolução do HTTP 1.0 para o HTTP 1.1 foi a capacidade dos clientes poderem usar várias conexões TCP em paralelo/simultâneo para o mesmo servidor HTTP.
5.c)	X		Apesar de ser um protocolo encapsulado no TCP, o HTTP é um protocolo assíncrono, sem estados (<i>stateless</i>), o que dificulta a implementação de aplicações distribuídas no paradigma clássico de cliente-servidor.
5.d)		X	O método de <i>pipelining</i> do HTTP permite uma troca de informação mais rápida, mas também exige mais largura de banda da rede do servidor HTTP.
5.e)	X		Os <i>proxy</i> HTTP diminuem a quantidade de dados obtida diretamente dos servidores remotos (fora da rede local servida pelos <i>proxy</i>) apenas para as entradas de objetos que estejam na sua <i>cache</i> .

6. Considere uma interação HTTP 1.1 em que um cliente A tenta obter uma página *web* dum servidor B. O RTT (*Round Trip Time*) médio, usando TCP (sem conexões paralelas e sem perda de dados), entre A e B é de 10 ms e o tempo que um segmento TCP demora a ir de A para B é o mesmo que o tempo que demora a ir de B para A, i.e., RTT/2. Assuma que: i) o tempo de processamento dos pacotes e o tempo de execução de qualquer tarefa computacional associada ao serviço prestado pelo servidor é irrelevante; ii) o tempo de transmissão dos segmentos TCP para a rede é irrelevante, tanto em A como em B; iii) o tamanho de janela do TCP nos dois sentidos será igual a 5 Kbytes e o MSS (*Maximum Segment Size*) nos dois sentidos será igual a 512 bytes; iv) é utilizado o mecanismo de *slow start*, nos dois sentidos, iniciando a janela de congestão com um MSS e com um *threshold* de 8*MSS; v) o cliente A não tem a capacidade de usar várias conexões TCP em paralelo; vi) os dados estão disponíveis para o cliente A assim que todos os bytes de todos os objetos da página chegam (i.e., para o tempo que o cliente A demora a obter os dados não se considera o tempo utilizado para fechar a última conexão TCP se depois disso já não há dados para receber); vii) a página *web* é constituída por um ficheiro HTML de 300 bytes e duas imagens que ocupam 1300 bytes cada uma. Indique (com X) se acha as afirmações verdadeiras (*true*) ou falsas (*false*).

	V/T	F	
6.a)	X		Se for usado o modo persistente sem método de <i>pipelining</i> , o tempo mínimo que se pode esperar para o cliente A poder obter os três objetos da página referida é de 50 ms.
6.b)	X		Se não for usado o modo persistente nem o método de <i>pipelining</i> , o cliente A pode obter os três objetos da página referida num tempo mínimo que é de 100 ms.
6.c)	X		Se for usado o modo persistente com método de <i>pipelining</i> , o cliente A pode obter os três objetos da página referida num tempo mínimo que é de 40 ms.
6.d)		X	Neste caso, se fosse usado HTTP 2.0 não poderíamos esperar qualquer diminuição do tempo de acesso à página HTML por parte do cliente A quando comparado com o acesso por HTTP 1.1 em modo persistente e com método de <i>pipelining</i> .
6.e)		X	Se entre o cliente A e o servidor B introduzirmos um <i>proxy</i> estaremos sempre a aumentar o tempo de acesso do cliente a essa página, mas a diminuir a quantidade de dados na rede de acesso do cliente A, desde que a página esteja na <i>cache</i> do <i>proxy</i> .

NÚMERO :**NOME :****PARTE IV – Encaminhamento IP**

7. Estude as seguintes afirmações genéricas sobre algoritmos e protocolos de encaminhamento IP e indique (com **X**) se as considera verdadeiras (*true*) ou se as considera falsas (*false*). Se não souber não indique qualquer uma das hipóteses.

Nota: LS – algoritmos estado das ligações; DV – algoritmos vetores de distância.

	V/T	F	
7.a)		X	Os protocolos de encaminhamento dinâmico IP são usados para gerir a informação do plano de controlo numa forma totalmente passiva e transparente para o plano de dados.
7.b)		X	Um protocolo de encaminhamento dinâmico específico e normalizado é definido para usar um algoritmo de encaminhamento do tipo LS ou do tipo DV ou dos dois tipos em simultâneo.
7.c)	X		Um dos principais fatores de complexidade da implementação dos algoritmos de encaminhamento é o dinamismo da topologia das redes quando as ligações entre os <i>routers</i> ficam ativas ou inativas com frequência.
7.d)	X		Um fator que complica a implementação dos protocolos de encaminhamento IP é o facto dos destinos finais usados nas tabelas de encaminhamento serem redes (ou sub-redes ou grupos de redes) e não <i>routers</i> individuais.
7.e)		X	Um algoritmo DV não pode ser utilizado com topologias de rede em que o custo da ligação entre dois <i>routers</i> é diferente nos dois sentidos da comunicação ou quando existe mais do que uma ligação direta entre dois <i>routers</i> .

8. Considere uma rede com a seguinte topologia: $C(u,y)=7$, $C(u,z)=3$, $C(z,w)=5$, $C(z,y)=1$ e $C(w,y)=3$; em que $C(a,b)$ indica o custo de usar a ligação direta entre o *router* a e o *router* b . O custo é igual nos dois sentidos da comunicação e coincide com o tempo (em ms) que leva uma mensagem a ser transmitida entre os *routers*. Assuma que este tempo é independente da quantidade de dados em cada mensagem, quer sejam mensagens LSA (*Link-State Announcement/Advertisement*) ou mensagens de anúncios de tabelas DV. Assuma também que o tempo de processamento de mensagens e execução de tarefas associadas aos algoritmos de encaminhamento é irrelevante e nunca há perda de mensagens. Indique (com **X**) se acha as seguintes afirmações verdadeiras (*true*) ou falsas (*false*). Não responda se não sabe a resposta. Nota: LS – algoritmos estado das ligações; DV – algoritmos vetores de distância.

	V/T	F	
8.a)	X		Se for usado um algoritmo LS a topologia de rede tem de ser conhecida em todos os <i>routers</i> antes do algoritmo LS ser executado. Usando difusão LSA, o tempo mínimo para que todos os <i>routers</i> tenham a informação da topologia estabilizada é de 20 ms.
8.b)	X		Se for usado o algoritmo Dijkstra, depois de conhecida a topologia de rede por todos os <i>routers</i> , obtém-se uma tabela LS para o <i>router</i> u igual à Tabela 1 (em anexo).
8.c)		X	Se for usado o algoritmo Bellman-Ford, na primeira iteração o <i>router</i> w recebe a seguinte informação dos vizinhos: $V_z=\{(u,3),(y,1),(w,4)\}$ e $V_y=\{(u,4),(z,1),(w,3)\}$; em que V_R são as melhores entradas (custo mais baixo) da tabela DV do <i>router</i> R .
8.d)	X		A tabela de encaminhamento, resultante da execução dos algoritmos, tem de ser igual em todos os <i>routers</i> , desde que a topologia de rede seja estável, independentemente do tipo de algoritmo ser LS ou DV.
8.e)		X	Em geral, as tabelas de encaminhamento convergem/estabilizam mais rapidamente em algoritmos LS porque os <i>routers</i> sabem a topologia completa da rede logo de início, antes do algoritmo começar a calcular a tabela de encaminhamento.

Tabela 1: Informação LS do *router* u do exercício 8.

<i>Step</i>	<i>N'</i>	<i>D(z), last hop before z</i>	<i>D(y), last hop before y</i>	<i>D(w), last hop before w</i>
0	u	3, z*	7, y*	∞, w^*
1	uz	-	4, z	8, z
2	uzy	-	-	7, y
3	uzyw	-	-	-

Nota*: no primeiro passo também se pode considerar o próprio router u como *last hop before R*.

