

Universidade do Minho - Dep.^{to} Informática
Licenciatura em Engenharia Informática - 1º Semestre, 2022/2023
COMUNICAÇÕES POR COMPUTADOR
Exame de Recurso Escrito – 31 janeiro
120 Minutos

PIN :	PASSWORD :
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 ou quando todos os alunos terminarem.• 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 :

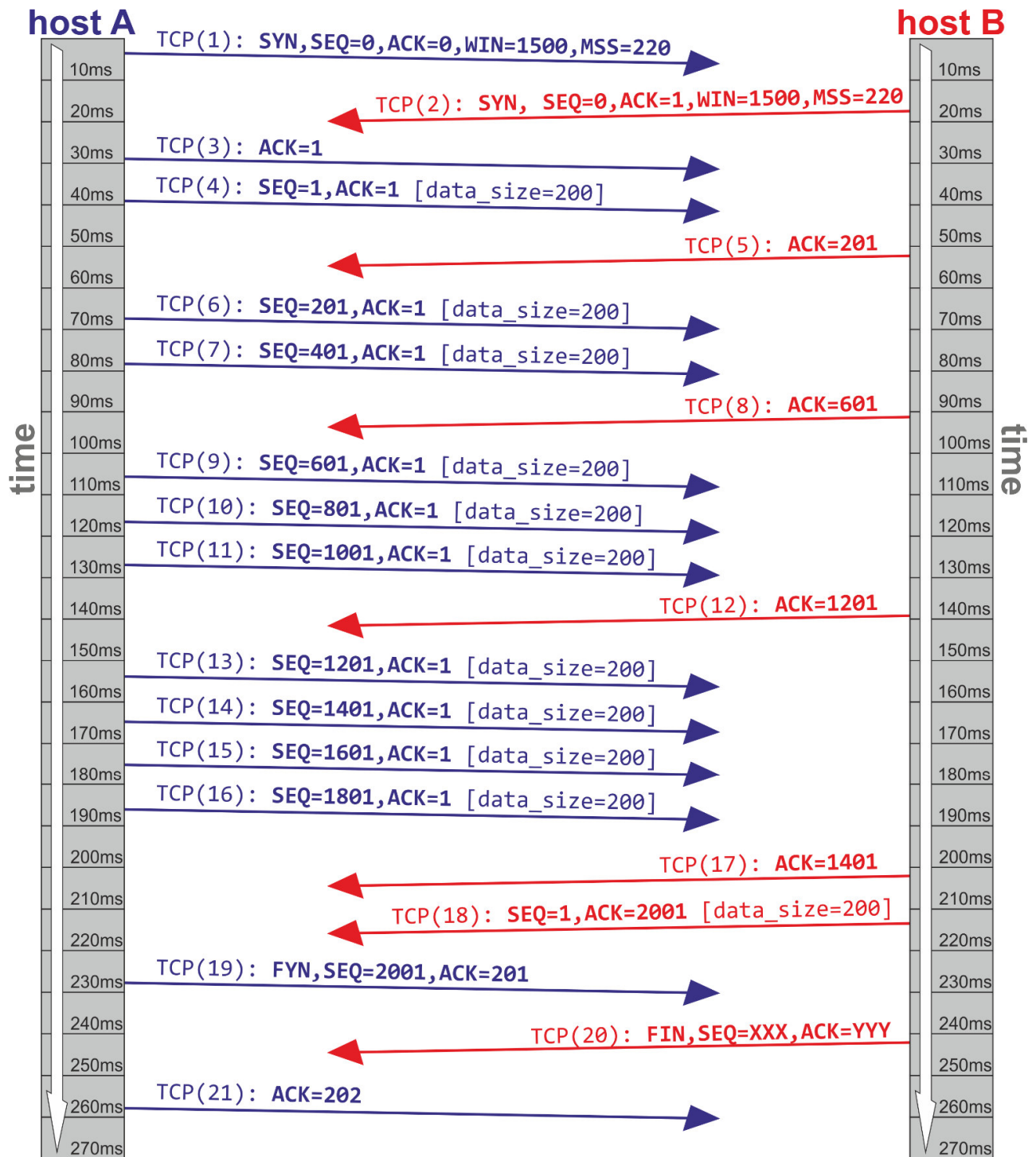
PARTE I – Protocolos de Transporte na Internet

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)			As aplicações que usam protocolos de transporte não orientados à conexão (como o <i>User Datagram Protocol</i> – UDP) têm obrigatoriamente de implementar, elas próprias, mecanismos de controlo de fluxo e de prevenção de congestão na rede
1.b)			Nunca é possível conectar/ligar um <i>socket</i> TCP (<i>Transmission Control Protocol</i>) dum <i>host</i> B a um <i>socket</i> UDP num <i>host</i> A.
1.c)			Para serviços implementados sobre redes que permitem ritmos de dados muito elevados é sempre aconselhável usar-se um protocolo orientado à conexão como o TCP para potenciar ainda mais a velocidade de transferência de dados.
1.d)			Uma aplicação Internet usando o protocolo de transporte TCP pode definir o valor dos parâmetros que definem o controlo de fluxo da informação, como o tamanho de janela de transmissão nos dois sentidos ou o tamanho máximo de segmento.
1.e)			Apesar do tamanho máximo dum datagrama UDP ser igual ao tamanho máximo dum segmento TCP, um datagrama UDP pode incluir mais dados aplicacionais que um segmento TCP do mesmo tamanho.

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)			O <i>host</i> B aceitou os valores dos parâmetros de funcionamento da conexão (tamanho de janela e tamanho máximo de segmento) propostos pelo <i>host</i> A para os dois sentidos da comunicação.
2.b)			No segmento TCP(20), enviado pelo <i>host</i> B para o <i>host</i> A, o valor do número de sequência (<i>sequence number</i>) XXX deve ser igual a 202 e o valor do número de confirmação (<i>acknowledgment number</i>) YYY deve ser igual a 2001.
2.c)			Analisando a interação protocolar podemos garantir que o segmento identificado como TCP(14), enviado pelo <i>host</i> A ao <i>host</i> B, demorou mais de 30ms a chegar ao <i>host</i> B.
2.d)			Pela análise da interação protocolar podemos garantir que, na direção da comunicação do <i>host</i> A para o <i>host</i> B, o valor do <i>timeout</i> usado é superior a 40ms.
2.e)			Pela análise da interação protocolar podemos concluir que, na direção da comunicação do <i>host</i> A para o <i>host</i> B, não 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)			O serviço DNS da Internet utiliza comunicação assíncrona na implementação da comunicação entre servidores e clientes, sendo que as <i>queries</i> DNS são sempre encapsuladas em segmentos TCP.
3.b)			Na organização hierárquica dos domínios de DNS, os nomes que podem ser utilizados para domínios só têm restrições para os domínios de topo (<i>top-level domains</i>). Para subdomínios dos domínios de topo a escolha dos nomes não sofre qualquer limitação.
3.c)			No modo recursivo de funcionamento da pesquisa/procura de informação no DNS, ainda que um servidor de topo aceite esse modo, não é garantido que esse modo seja utilizado pelos servidores seguintes necessários para se obter uma resposta final.
3.d)			Se um cliente fizer duas <i>queries</i> iguais, consecutivamente, ao mesmo servidor DNS, se a resposta for válida, o cliente irá receber mais rapidamente a resposta à segunda <i>query</i> , desde que algum dos servidores envolvidos na iteração implemente <i>cache</i> .
3.e)			Um servidor secundário dum domínio de DNS pode ter a sua base de dados desatualizada durante algum tempo (diferente da base de dados do respetivo servidor primário) e, mesmo assim, ser um servidor autoritativo para esse domínio.

4. Uma aplicação cliente envia a *query* “*sss.mmm.big. MX*” 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_domínio*” (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_domínio*” (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)			i) No início, o SR reenvia a <i>query</i> para o ST <i>10.10.10.10</i> que responde com o campo RV nulo, no campo AV inclui as entradas NS respetivas, <i>sp.big</i> , <i>ss1.big</i> e <i>ss2.big</i> , e no campo EV inclui o endereço IP desses servidores.
4.b)			ii) O SR recebe a resposta do ST, guarda as entradas dos campos AV e EV em <i>cache</i> e reenvia a <i>query</i> inicial para <i>sp.big</i> que responde com RV nulo, em AV inclui as entradas NS <i>sp.mmm.big</i> e <i>ss1.mmm.big</i> , e no campo EV os seus endereços IP.
4.c)			iii) O SP contactado em ii) responde com o campo RV nulo, no campo AV inclui os nomes dos servidores autoritativos para <i>sss.mmm.big</i> e no campo EV inclui os respetivos endereços IP.
4.d)			iv) O SR recebe a resposta referida em iii), guarda as entradas dos campos RV, AV e EV em <i>cache</i> e reenvia a resposta para o cliente.
4.e)			Por razões de segurança e eficiência é natural e desejável que os subdomínios dos domínios de topo (como o domínio <i>mmm.big</i> deste exemplo) tenham pelo menos um servidor secundário na mesma rede local do servidor primário do seu domínio 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)			Os métodos PUT e DELETE introduzidos na versão 1.1 do HTTP são métodos perfeitamente seguros desde que os clientes HTTP tenham cuidados acrescidos quando enviam estes tipos de pedidos aos servidores.
5.b)			Uma das características melhoradas na evolução do HTTP 1.1 para o HTTP 2.0 foi a capacidade dum cliente poder usar uma única conexão TCP para pedir todos os objetos duma página a um servidor HTTP.
5.c)			Apesar de ser um protocolo assíncrono e seguindo um paradigma de comunicação sem estados (<i>stateless protocol</i>), o HTTP inclui a implementação do conceito de sessão através de <i>cookies</i> , o que permite ultrapassar parte das suas limitações.
5.d)			Quando uma página tem múltiplos objetos, é preferível o cliente utilizar o método de <i>pipelining</i> do HTTP numa única conexão TCP com o servidor, do que usar várias conexões TCP para o mesmo servidor sem o método <i>pipelining</i> .
5.e)			Os <i>proxy</i> HTTP servem para acelerar o acesso a objetos que estejam na sua <i>cache</i> mas não são úteis para serviços em que os objetos/informação no servidor esteja em constante atualização, como é o caso do acesso a serviços de videoconferência.

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 20 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 oito MSS (*Maximum Segment Size*) e o MSS nos dois sentidos será igual a um Kbyte; iv) é utilizado o mecanismo de *slow start*, nos dois sentidos, iniciando a janela de congestão com um MSS e com um *threshold* de quatro 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 500 bytes, uma imagem A com 3500 bytes (1ª referência a aparecer na página) e outra B com 1100 bytes (2ª referência a aparecer). Indique se acha as afirmações verdadeiras ou falsas.

	V/T	F	
6.a)			i) 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 80 ms.
6.b)			ii) Se não for usado o modo persistente nem o método de <i>pipelining</i> , o cliente A irá obter os três objetos da página referida num tempo mínimo que é maior do que o do método referido em i) porque teria de fazer três conexões TCP para o servidor.
6.c)			iii) 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 menor do que o do método referido em i) apenas se a imagem B for transferida em primeiro lugar.
6.d)			Neste caso, se entre o cliente A e o servidor B introduzirmos um <i>proxy</i> , estaremos sempre a diminuir o tempo de acesso do cliente a essa página desde que, pelo menos, uma das imagens ou a própria página esteja já na <i>cache</i> do <i>proxy</i> .
6.e)			A versão do protocolo HTTP é irrelevante para quem cria conteúdo <i>web</i> para ser publicado num servidor HTTP. Ou seja, do ponto de vista comunicacional, o HTTP não se importa com a versão do HTML (ou outra linguagem) dos objetos a transferir.

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)			Os protocolos de encaminhamento dinâmico IP são usados para gerir a informação do plano de controlo e precisam de regras de comunicação entre os <i>routers</i> que têm de seguir as próprias tabelas de encaminhamento criadas por eles próprios.
7.b)			Um protocolo de encaminhamento dinâmico normalizado inclui um algoritmo de encaminhamento, regras de comunicação entre os <i>routers</i> e outros mecanismos necessários para a completa implementação do plano de controlo em redes IP.
7.c)			Tanto os protocolos de encaminhamento interno como os protocolos de encaminhamento externo podem utilizar algoritmos de encaminhamento LS ou DV na sua implementação.
7.d)			Um <i>router</i> pode, ao mesmo tempo, implementar um protocolo de encaminhamento interno e um protocolo de encaminhamento externo. É o caso dos <i>routers</i> fronteira dos sistemas autónomos na Internet.
7.e)			Ao contrário dos algoritmos DV, os algoritmos LS não podem ser utilizados com topologias de rede em que o custo da ligação é diferente nos dois sentidos numa ligação/conexão direta entre dois <i>routers</i> .

8. Considere uma rede com a seguinte topologia: $C(u,w)=5$, $C(u,z)=2$, $C(z,w)=2$, $C(z,y)=5$ e $C(w,y)=2$; 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)			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 12 ms.
8.b)			Se for usado o algoritmo de 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)			Se for usado o algoritmo de Bellman-Ford, na primeira iteração o <i>router</i> u recebe a seguinte informação dos vizinhos: $V_z=\{(u,2),(y,5),(w,2)\}$ e $V_w=\{(u,5),(z,2),(y,2)\}$; em que V_R são as melhores entradas (custo mais baixo) da tabela DV do <i>router</i> R .
8.d)			Se for usado o algoritmo de Bellman-Ford, depois da primeira iteração em que o <i>router</i> u recebe a informação dos <i>routers</i> vizinhos z e w , a tabela DV do <i>router</i> u é igual à Tabela 2 (em anexo).
8.e)			As tabelas de encaminhamento nos algoritmos DV precisam de ser deduzidas das tabelas DV, enquanto que nos algoritmos LS as tabelas de encaminhamento não precisam de ser deduzidas das tabelas LS, são imediatas.

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

<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	2, z*	∞, y^*	5, w*
1	uz	-	7, z	4, z
2	uzw	-	6, w	-
3	uzwy	-	-	-

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

Tabela 2: Informação DV do *router* u do exercício 8.d)

<i>Distance to Destination</i>	<i>First hop is z</i>	<i>First hop is w</i>
z	2	7
y	7	7
w	4	5

