

Curso de Tecnologia em Sistemas de Computação

Disciplina: Redes de Computadores II

AD1 – 1º semestre de 2019 – GABARITO

Questão 1 20 pontos

O problema de encaminhamento consiste em determinar por qual interface de rede um pacote IP que chega a um roteador será transmitido. O encaminhamento dos pacotes é realizado de acordo com a tabela de roteamento. O objetivo desta questão é compreender como funciona este mecanismo na Internet.

Suponha que um roteador da Internet deva encaminhar seus pacotes de acordo com a tabela abaixo, cujas faixas de endereços IP está representada em binário.

Faixa de endereço destino	Interface
00111001 10111011 11000000 00000000 a	0
00111001 10111011 11001111 11111111 a	1
00111001 10111011 11001110 10000000 a	2
00111001 10111011 11001110 01111111 a	0
00111001 10111011 11001110 11111111 a	0
00111001 10111011 11001110 01101000 a	0
00111001 10111011 11001110 01101111 a	0
00111001 10111011 11000110 00000000 a	0
00111001 10111011 11000110 11111111 a	2
00111001 10111011 11001010 00000000 a	2
00111001 10111011 11001010 01111111 caso contrário	1

- (a) Construa a tabela de roteamento com base nas informações da tabela acima, isto é, determine o prefixo (em notação binária) correspondente a cada linha da tabela acima. Sua tabela deve indicar a interface de rede correspondente a cada prefixo.

Resposta:

Prefixo	Interface
00111001 10111011 1100	0
00111001 10111011 11001110 0	1
00111001 10111011 11001110 1	2
00111001 10111011 11001110 01101	0
00111001 10111011 11000110	0
00111001 10111011 11001010 0	2
-	1

- (b) Reescreva a tabela de roteamento encontrada acima utilizando a notação a.b.c.d/x.

Resposta:

Prefixo	Interface
57.187.192.0/20	0
57.187.206.0/25	1
57.187.206.128/25	2
57.187.206.104/29	0
57.187.198.0/24	0
57.187.202.0/25	2
0.0.0.0/0	1

- (c) Determine para qual interface de rede cada um dos pacotes abaixo, com os respectivos endereços de destino, será encaminhado.

- | | |
|---|---|
| i. 57.187.194.46 → interface 0 | vii. 57.187.206.106 → interface 0 |
| ii. 57.187.206.52 → interface 1 | viii. 57.187.206.224 → interface 2 |
| iii. 57.187.195.17 → interface 0 | ix. 57.187.206.71 → interface 1 |
| iv. 57.187.242.147 → interface 1 | x. 57.187.227.180 → interface 1 |
| v. 57.187.201.83 → interface 0 | xi. 57.187.206.79 → interface 1 |
| vi. 57.187.206.109 → interface 0 | xii. 57.187.206.209 → interface 2 |

- (d) Determine quais regras (ou seja, linhas) da tabela de roteamento acima podem ser removidas sem afetar o encaminhamento dos pacotes. Indique se isto não for o caso.¹

Resposta:

A regra do prefixo 57.187.198.0/24 é redundante.

Questão 2 15 pontos

Um problema fundamental em gerência de redes consiste em dividir uma determinada rede em múltiplas subredes. Resolver este problema nos permite alocar endereços IP para as estações pertencentes a cada uma destas subredes, visto que uma interface de rede de uma estação ou roteador pertence a exatamente uma subrede.

Considere um roteador que interconecta diferentes subredes associadas às suas interfaces. Cada item a seguir ilustra diferentes configurações para o roteador, indicando o endereço de rede que deve ser dividido, o número de subredes a serem criadas (uma para cada interface do roteador), e o número de estações em cada subrede.

Para cada configuração, determine os endereços das subredes na forma a.b.c.d/x para atender aos requisitos, ou indique não ser possível atendê-los (explique sua resposta neste caso). Repare que não há uma solução única.²

¹Dica: procure por faixas mais específicas.

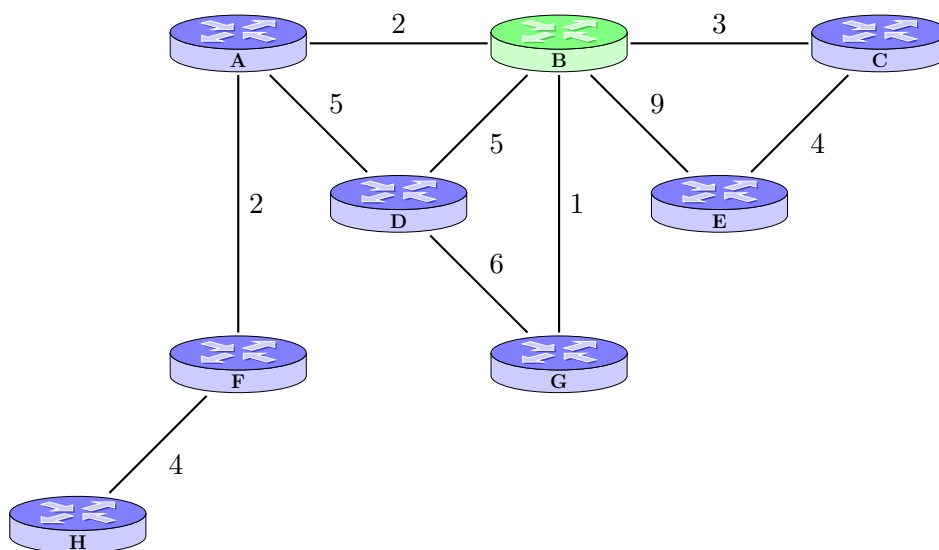
²Dica: aloque os endereços das subredes em ordem decrescente de seus tamanhos (i.e., maior primeiro).

i. R_1 : 159.71.102.128/25 com 5 subredes: R_{11} : 25 estações R_{12} : 4 estações R_{13} : 22 estações R_{14} : 29 estações R_{15} : 29 estações Alocação impossível	ii. R_2 : 172.0.0.0/6 com 4 subredes: R_{21} : 4000000 estações 173.128.0.0/10 R_{22} : 2000000 estações 173.192.0.0/10 R_{23} : 8000000 estações 173.0.0.0/9 R_{24} : 15000000 estações 172.0.0.0/8	iii. R_3 : 66.203.240.0/20 com 4 subredes: R_{31} : 800 estações 66.203.244.0/22 R_{32} : 300 estações 66.203.252.0/23 R_{33} : 1000 estações 66.203.240.0/22 R_{34} : 600 estações 66.203.248.0/22
--	---	--

Questão 3 20 pontos

O algoritmo de Dijkstra — que leva o nome do professor holandês Edsger W. Dijkstra, que o desenvolveu em 1956 — é um dos algoritmos mais utilizados para encontrar os caminhos mais curtos em uma rede onde as arestas (enlaces) possuem pesos não negativos. Este algoritmo é utilizado, por exemplo, na Internet, para que roteadores, conhecendo a topologia da rede em questão, possam calcular rotas ótimas e encaminhar pacotes segundo estas rotas. Algoritmos de roteamento que seguem esta ideia são ditos algoritmos do tipo estado de enlace (*link state*). O objetivo desta questão é entender como funciona o algoritmo de Dijkstra.

Considere a rede ilustrada abaixo, onde os enlaces estão anotados com seus respectivos custos.

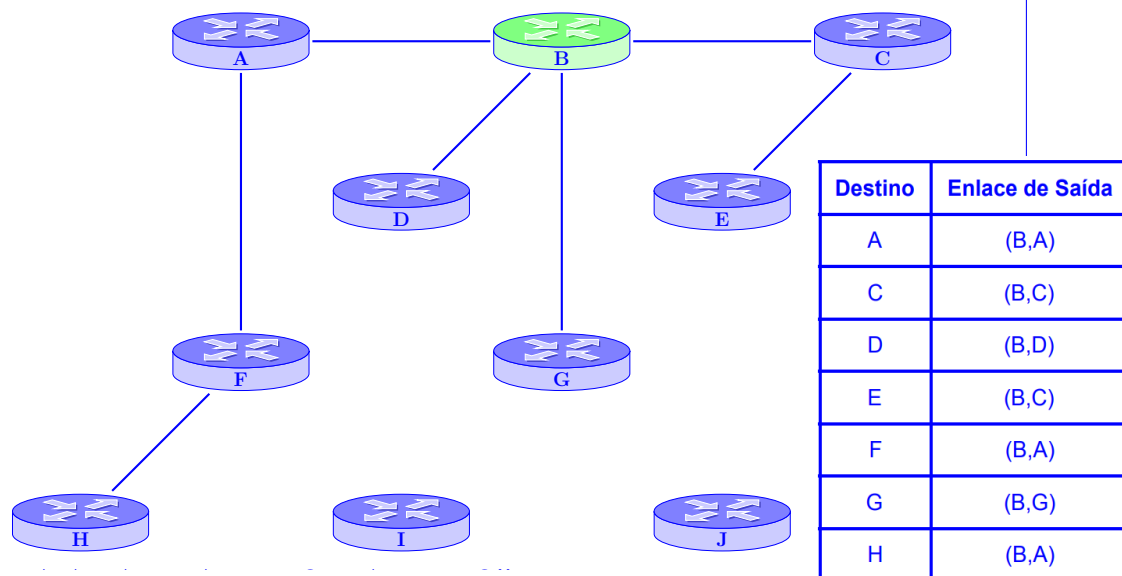


- (a) Utilizando o algoritmo de Dijkstra, calcule os caminhos mais curtos a partir do nó B, destacado em verde, para todos os outros nós da rede. Construa uma tabela igual à mostrada em aula que mostra o funcionamento do algoritmo de forma iterativa.

Resposta:

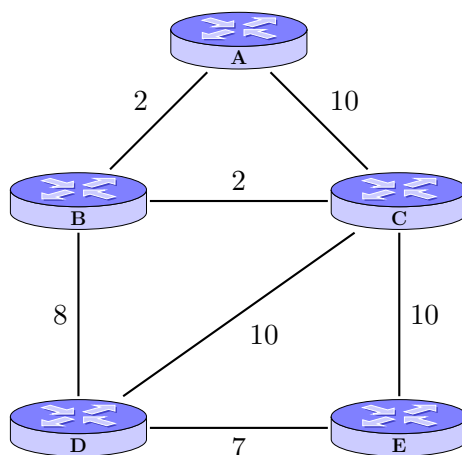
	N°	d _A p _A	d _C p _C	d _D p _D	d _E p _E	d _F p _F	d _G p _G	d _H p _H
0	B	2 B	3 B	5 B	9 B	∞ -	1 B	∞ -
1	BG	2 B	3 B	5 B	9 B	∞ -		∞ -
2	BGA		3 B	5 B	9 B	4 A		∞ -
3	BGAC			5 B	7 C	4 A		∞ -
4	BGACF			5 B	7 C			8 F
5	BGACFD				7 C			8 F
6	BGACFDE							8 F
7	BGACFDEH							

- (b) A partir do resultado do item anterior, construa a árvore de caminhos mínimos a partir de B calculada pelo algoritmo. Construa também a tabela de roteamento de B.

Resposta:

Questão 4 15 pontos

Algoritmos de roteamento do tipo *distance vector* são algoritmos distribuídos para calcular os caminhos mais curtos em uma rede com custos. Nestes algoritmos, os nós da rede não possuem conhecimento da topologia da rede e atualizam seus caminhos mínimos em função de mensagens trocadas com seus vizinhos — isto é, os nós devem continuamente cooperar uns com os outros, através da troca de informações, para que todos descubram seus caminhos mínimos. O objetivo desta questão é compreender como funcionam estes algoritmos.



- (a) Considere a rede ilustrada acima, onde os enlaces estão anotados com seus respectivos custos. Vamos assumir que os nós da rede executam uma versão síncrona do algoritmo *distance vector*, de forma que cada passo do algoritmo é executado por todos os nós simultaneamente, antes de todos avançarem para o passo seguinte.

Antes da execução do algoritmo, todos os nós sabem apenas o custo dos enlaces que os liga a seus vizinhos. Determine as tabelas de distância iniciais de cada nó. A partir destas tabelas de distância, determine também os vetores de distância iniciais de cada nó. Este é o “passo 0” do algoritmo.

Resposta:

Passo 0:

A		custo até				B	custo até				C	custo até				D	custo até				E	custo até								
		B	C	D	E			A	C	D	E			A	B	D	E			A	B	C	E			A	B	C	D	
via	B	2	∞	∞	∞	via	A	2	∞	∞	∞	via	A	10	∞	∞	∞	via	B	∞	8	∞	∞	via	C	∞	∞	10	∞	
	C	∞	10	∞	∞		C	∞	2	∞	∞		C	∞	2	∞	∞		C	∞	∞	10	∞		D	∞	∞	∞	7	

- (b) Na primeira iteração do algoritmo, todos os nós enviam aos seus vizinhos os vetores de distância calculados no passo 0. Cada nó utiliza os vetores recebidos para compor uma nova tabela de distâncias, e utiliza esta nova tabela para atualizar o seu próprio vetor de distâncias. Determine as novas tabelas de distância de cada nó e, a partir delas, calcule os novos vetores de distância de cada nó. Isto conclui o passo 1 do algoritmo.

Resposta:

Passo 1:

A		custo até				B	custo até				C	custo até				D	custo até				E	custo até							
		B	C	D	E		A	C	D	E			A	B	D	E			A	B	C	E			A	B	C	D	
via	B	2	4	10	∞	via	A	2	12	∞	∞	via	A	10	12	∞	∞	via	B	10	8	10	∞	via	C	20	12	10	20
	C	12	10	20	20		C	12	2	12	12		B	4	2	10	∞		C	20	12	10	20		D	∞	15	17	7
							D	∞	18	8	15		D	∞	18	10	17		E	∞	∞	17	7						
							E						E	∞	∞	17	10												

vetor de A					vetor de B					vetor de C					vetor de D					vetor de E				
A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
0	2	4	10	20	2	0	2	8	12	4	2	0	10	10	10	8	10	0	7	20	12	10	7	0

- (c) As iterações seguintes do algoritmo procedem da mesma forma que a primeira, mas utilizando os vetores de distância calculados no passo anterior. Esta dinâmica irá continuar até a *convergência* do algoritmo, isto é, até que chegue um passo em que o vetor de distâncias de todos os nós não se modifique. Repita o item anterior — isto é, determine as novas tabelas de distância e recalcule os vetores de distância de todos os nós — até que ocorra a convergência do algoritmo. Em quantos passos ocorreu esta convergência?

Resposta:

O algoritmo irá convergir em apenas 3 passos.

Passo 2:

A		custo até				B	custo até				C	custo até				D	custo até				E	custo até							
		B	C	D	E		A	C	D	E			A	B	D	E			A	B	C	E			A	B	C	D	
via	B	2	4	10	14	via	A	2	6	12	22	via	A	10	12	20	30	via	B	10	8	10	20	via	C	14	12	10	20
	C	12	10	20	20		C	6	2	12	12		B	4	2	10	14		C	14	12	10	20		D	17	15	17	7
							D	18	18	8	15		D	20	18	10	17		E	27	19	17	7						
							E						E	30	22	17	10												

vetor de A					vetor de B					vetor de C					vetor de D					vetor de E				
A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
0	2	4	10	14	2	0	2	8	12	4	2	0	10	10	10	8	10	0	7	14	12	10	7	0

Passo 3:

A		custo até				B	custo até				C	custo até				D	custo até				E	custo até							
		B	C	D	E		A	C	D	E		A	B	D	E		A	B	C	E		A	B	C	D				
via	B	2	4	10	14	via	A	2	6	12	16	via	A	10	12	20	24	via	B	10	8	10	20	via	C	14	12	10	20
	C	12	10	20	20		C	6	2	12	12		B	4	2	10	14		C	14	12	10	20		D	17	15	17	7
							D	18	18	8	15		D	20	18	10	17		E	21	19	17	7						
							E						E	24	22	17	10												

vetor de A					vetor de B					vetor de C					vetor de D					vetor de E				
A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
0	2	4	10	14	2	0	2	8	12	4	2	0	10	10	10	8	10	0	7	14	12	10	7	0

Questão 5..... 15 pontos

Considere o problema de tradução de endereços entre a rede privada e a rede pública que um NAT precisa resolver. Este problema é resolvido utilizando uma tabela de tradução de endereços que é atualizada quando uma nova conexão é estabelecida entre uma estação na rede privada e outra na rede pública. O objetivo desta questão é compreender exatamente como funciona este mecanismo.

Considere um NAT cujo endereço IP na rede pública é 189.7.100.156 e que gerencia as conexões da rede privada, que ocupa a faixa 192.168.0.0/16. Inicialmente o NAT em questão possui a seguinte tabela de tradução, onde cada regra é identificada por um número:

	(IP, porta) da estação local	(IP, porta) da estação remota	Porta pública no NAT
(1)	192.168.0.1, 32296	129.37.161.112, 26234	1028
(2)	192.168.0.1, 9106	77.84.74.36, 19233	32748
(3)	192.168.0.2, 13481	73.218.61.210, 8960	1025
(4)	192.168.0.2, 21753	224.68.4.253, 24350	13241
(5)	192.168.0.2, 14275	16.137.217.244, 24869	29395
(6)	192.168.0.2, 5084	5.204.194.151, 14906	23900
(7)	192.168.0.1, 12475	159.143.167.135, 13196	10780
(8)	192.168.0.1, 18702	125.213.219.56, 6713	23552
(9)	192.168.0.3, 14855	13.119.49.173, 30962	23000
(10)	192.168.0.1, 10458	189.58.214.98, 18440	29648

Suponha que todos os fluxos nesta questão são fluxos TCP, identificados unicamente pelos endereços e portas das duas estações envolvidas na conexão.

- (a) Considere que o NAT irá receber uma sequência de pacotes provenientes da rede pública (cuja estação de destino está na rede privada). Em cada item a seguir, identificamos os endereços e portas, de origem e destino, de um destes pacotes ao ser enviado pela estação que o gerou. Para cada pacote, determine os endereços e portas, de origem e de destino, que serão colocados no pacote quando o NAT realizar a tradução de endereços e encaminhá-lo à rede privada. Se o NAT descartar o pacote em vez de encaminhá-lo, indique isto em sua resposta.

- i. Origem: 159.143.167.135, 13196; Destino: 189.7.100.156, 10780
 → **Origem: 159.143.167.135, 13196; Destino: 192.168.0.1, 12475**
- ii. Origem: 125.213.219.56, 6713; Destino: 189.7.100.156, 23552
 → **Origem: 125.213.219.56, 6713; Destino: 192.168.0.1, 18702**
- iii. Origem: 31.234.37.84, 17880; Destino: 189.7.100.156, 13241
descartado
- iv. Origem: 224.68.4.253, 24350; Destino: 189.7.100.156, 23900
descartado
- v. Origem: 73.218.61.210, 8960; Destino: 189.7.100.156, 1028
descartado
- vi. Origem: 5.204.194.151, 14906; Destino: 189.7.100.156, 23900
 → **Origem: 5.204.194.151, 14906; Destino: 192.168.0.2, 5084**
- vii. Origem: 73.218.61.210, 8960; Destino: 189.7.100.156, 1025
 → **Origem: 73.218.61.210, 8960; Destino: 192.168.0.2, 13481**
- viii. Origem: 77.84.74.36, 19233; Destino: 189.7.100.156, 1028
descartado
- ix. Origem: 77.84.74.36, 19233; Destino: 189.7.100.156, 13241
descartado
- x. Origem: 118.228.76.66, 6700; Destino: 189.7.100.156, 1025
descartado

- (b) Considere agora, que o NAT irá receber uma sequência de pacotes provenientes da rede privada (cuja estação de destino está na rede pública). Os endereços e portas, de origem e destino, destes pacotes são identificados nos itens a seguir.

Para cada pacote, determine se o NAT precisará ou não criar uma nova entrada em sua tabela de tradução ao encaminhá-lo para a rede pública. Se a nova entrada for necessária, determine o seu conteúdo; se não, identifique a entrada já existente que o NAT irá utilizar para encaminhar o pacote. Considere que somente portas a partir da 1024 estão disponíveis para o NAT. Note que há mais de uma solução correta.³

- i. Origem: 192.168.0.2, 5084; Destino: 5.204.194.151, 28836

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(11)	192.168.0.2, 5084	5.204.194.151, 28836	1024

- ii. Origem: 192.168.0.2, 5084; Destino: 5.204.194.151, 24888

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(12)	192.168.0.2, 5084	5.204.194.151, 24888	1026

- iii. Origem: 192.168.0.2, 13481; Destino: 207.125.189.159, 1778

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(13)	192.168.0.2, 13481	207.125.189.159, 1778	1027

- iv. Origem: 192.168.0.1, 9106; Destino: 77.84.74.36, 19233

Encaminhado segundo a entrada (2)

- v. Origem: 192.168.0.2, 5618; Destino: 5.204.194.151, 28836

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(14)	192.168.0.2, 5618	5.204.194.151, 28836	1029

- vi. Origem: 192.168.0.1, 18702; Destino: 125.213.219.56, 21045

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(15)	192.168.0.1, 18702	125.213.219.56, 21045	1030

- vii. Origem: 192.168.0.3, 14855; Destino: 13.119.49.173, 10653

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(16)	192.168.0.3, 14855	13.119.49.173, 10653	1031

- viii. Origem: 192.168.0.2, 14275; Destino: 16.137.217.244, 24869

Encaminhado segundo a entrada (5)

- ix. Origem: 192.168.0.2, 14566; Destino: 207.125.189.159, 1778

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(17)	192.168.0.2, 14566	207.125.189.159, 1778	1032

- x. Origem: 192.168.0.2, 13481; Destino: 207.125.189.159, 16638

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(18)	192.168.0.2, 13481	207.125.189.159, 16638	1033

- (c) Utilizando sua resposta para o item (b), determine os endereços e portas, de origem e destino, que o NAT irá colocar em cada um dos pacotes anteriores quando for encaminhá-lo à rede pública.

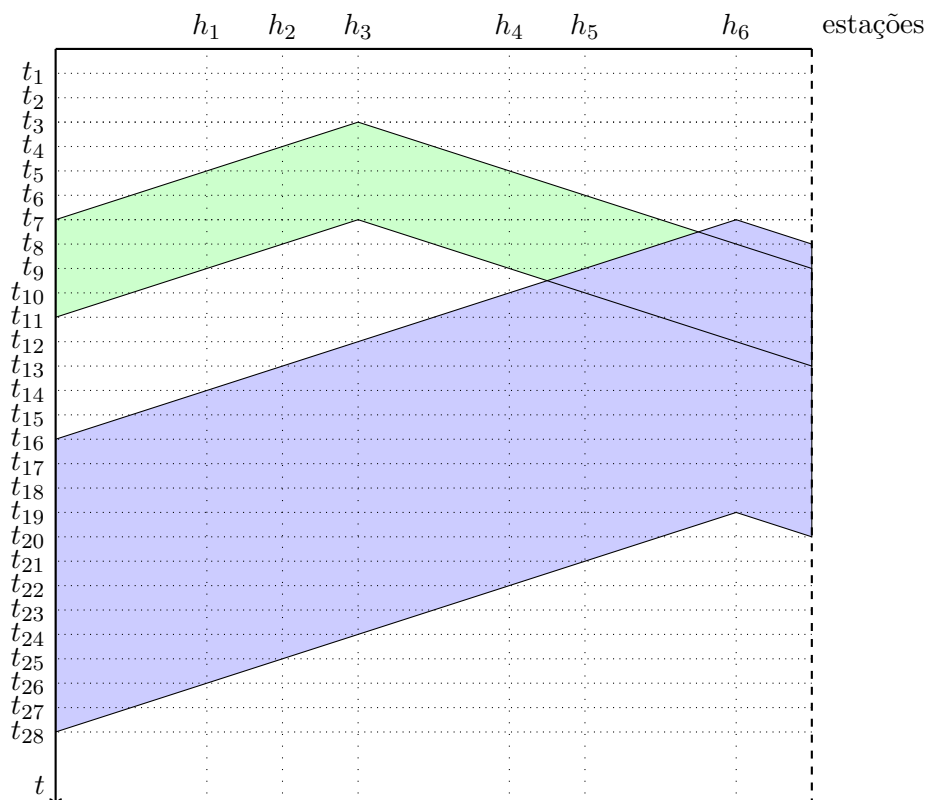
³Dica 1: cuidado com as portas já utilizadas. Dica 2: a entrada que um pacote criar pode ser utilizada pelos que forem enviados depois.

- i. Origem: 192.168.0.2, 5084; Destino: 5.204.194.151, 28836
→ **Origem: 189.7.100.156, 1024; Destino: 5.204.194.151, 28836**
- ii. Origem: 192.168.0.2, 5084; Destino: 5.204.194.151, 24888
→ **Origem: 189.7.100.156, 1026; Destino: 5.204.194.151, 24888**
- iii. Origem: 192.168.0.2, 13481; Destino: 207.125.189.159, 1778
→ **Origem: 189.7.100.156, 1027; Destino: 207.125.189.159, 1778**
- iv. Origem: 192.168.0.1, 9106; Destino: 77.84.74.36, 19233
→ **Origem: 189.7.100.156, 32748; Destino: 77.84.74.36, 19233**
- v. Origem: 192.168.0.2, 5618; Destino: 5.204.194.151, 28836
→ **Origem: 189.7.100.156, 1029; Destino: 5.204.194.151, 28836**
- vi. Origem: 192.168.0.1, 18702; Destino: 125.213.219.56, 21045
→ **Origem: 189.7.100.156, 1030; Destino: 125.213.219.56, 21045**
- vii. Origem: 192.168.0.3, 14855; Destino: 13.119.49.173, 10653
→ **Origem: 189.7.100.156, 1031; Destino: 13.119.49.173, 10653**
- viii. Origem: 192.168.0.2, 14275; Destino: 16.137.217.244, 24869
→ **Origem: 189.7.100.156, 29395; Destino: 16.137.217.244, 24869**
- ix. Origem: 192.168.0.2, 14566; Destino: 207.125.189.159, 1778
→ **Origem: 189.7.100.156, 1032; Destino: 207.125.189.159, 1778**
- x. Origem: 192.168.0.2, 13481; Destino: 207.125.189.159, 16638
→ **Origem: 189.7.100.156, 1033; Destino: 207.125.189.159, 16638**

Questão 6 15 pontos

O CSMA é um dos mecanismos mais utilizados para acessar o meio compartilhado, fazendo parte do padrão Ethernet. Uma de suas principais características é que, antes de iniciarem suas transmissões, as estações escutam o meio para detectar transmissões que estejam em andamento, minimizando (mas não evitando) as colisões. O objetivo desta questão é compreender melhor o funcionamento deste mecanismo.

Considere o cenário de transmissão ilustrado na figura a seguir, onde o posicionamento das estações é apresentado no eixo horizontal, e o tempo no eixo vertical. Responda às perguntas utilizando a figura.



- (a) Quais estações transmitiram? Em que instantes de tempo cada uma destas estações iniciou e terminou sua transmissão?

Resposta:

Duas estações transmitiram: a estação h_3 iniciou sua transmissão primeiro, utilizando o meio entre os instantes de tempo t_3 e t_7 . Depois disto, a estação h_6 transmite dados entre t_7 e t_{19} .

- (b) Considere todas as estações que não transmitiram. Determine o instante de tempo que cada uma delas começa e termina de receber cada uma das transmissões.

Resposta:

Estação	Transmissão de h_3	Transmissão de h_6
h_1	$t_5 - t_9$	$t_{14} - t_{26}$
h_2	$t_4 - t_8$	$t_{13} - t_{25}$
h_4	$t_5 - t_9$	$t_{10} - t_{22}$
h_5	$t_6 - t_{10}$	$t_9 - t_{21}$

- (c) Para cada estação, determine o instante de tempo em que ela detecta a colisão.
- (d) Para cada estação, determine o período de tempo em que ela percebe o meio como ocupado.

Resposta:

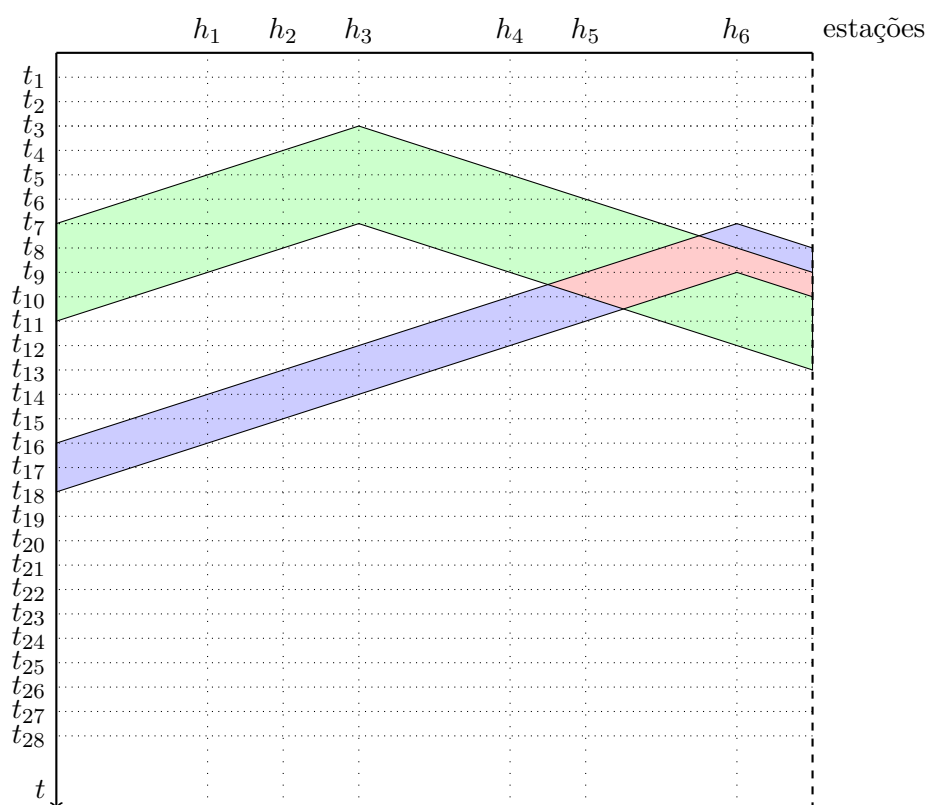
Estação	Vê meio ocupado entre:	Detecta colisão em:
h_1	$t_5 - t_{26}$	t_{14}
h_2	$t_4 - t_{25}$	t_{13}
h_3	$t_3 - t_{24}$	t_{12}
h_4	$t_5 - t_{22}$	t_{10}
h_5	$t_6 - t_{21}$	t_9
h_6	$t_7 - t_{19}$	t_8

- (e) Considere os instantes de tempo t_4 , t_8 e t_{24} . Quais estações podem iniciar uma transmissão nestes instantes de tempo?

Resposta:

	Estações:
t_4	h_1, h_4, h_5, h_6
t_8	—
t_{24}	h_4, h_5, h_6

- (f) Considere agora o mesmo cenário de transmissão acima, mas com o uso do protocolo CSMA/CD, conforme ilustrado na figura a seguir. Repita os itens (a) a (e) para este cenário.



Resposta:

Novamente, as estações h_3 e h_6 transmitiram. Desta vez, a estação h_3 transmitiu entre os instantes de tempo t_3 e t_7 , e a estação h_6 transmitiu entre os instantes de tempo t_7 e t_9 . As demais estações recebem estas transmissões conforme a tabela a seguir:

Estação	Transmissão de h_3	Transmissão de h_6
h_1	$t_5 - t_9$	$t_{14} - t_{16}$
h_2	$t_4 - t_8$	$t_{13} - t_{15}$
h_4	$t_5 - t_9$	$t_{10} - t_{12}$
h_5	$t_6 - t_{10}$	$t_9 - t_{11}$

As estações irão receber transmissões e detectar a colisão das mesmas nestes instantes de tempo:

Estação	Vê meio ocupado entre:	Detecta colisão em:
h_1	$t_5 - t_{16}$	t_{14}
h_2	$t_4 - t_{15}$	t_{13}
h_3	$t_3 - t_{14}$	t_{12}
h_4	$t_5 - t_{12}$	t_{10}
h_5	$t_6 - t_{11}$	t_9
h_6	$t_7 - t_{12}$	t_8

Por fim, nos instantes de tempo assinalados, as estações que enxergam o meio como livre (e, portanto, podem transmitir) são as seguintes:

	Estações:
t_4	h_1, h_4, h_5, h_6
t_8	—
t_{24}	$h_1, h_2, h_3, h_4, h_5, h_6$

- (g) Compare os períodos de tempo em que as estações percebem o meio como ocupado nos dois casos. Qual foi o ganho de tempo trazido pelo CSMA/CD para cada estação?

Resposta:

O ganho de tempo, para cada estação, é dado pelo período em que a estação veria o meio como ocupado com o protocolo CSMA, mas como livre com o protocolo CSMA/CD. Este período de tempo é indicado nesta tabela:

Estação	Ganho de tempo
h_1	$t_{16} - t_{26}$
h_2	$t_{15} - t_{25}$
h_3	$t_{14} - t_{24}$
h_4	$t_{12} - t_{22}$
h_5	$t_{11} - t_{21}$
h_6	$t_{12} - t_{19}$