

Curso de Tecnologia em Sistemas de Computação
Disciplina: Redes de Computadores II
AD1 – 2º semestre de 2019 – GABARITO

Questão 1 10 pontos

O objetivo deste exercício é entender melhor a relação entre faixas de endereços IP, endereços de rede, e endereços IPs. Repare que todo endereço de rede define uma faixa de endereços IP e que todo endereço IP pertence a uma ou mais redes. O entendimento destes conceitos é fundamental e será necessário para resolver outras questões desta avaliação.

(a) Determine se o endereços IP a seguir pertencem ou não às redes correspondentes.

- i. O endereço 10.176.190.239 **não pertence** à rede 12.0.0.0/8
- ii. O endereço 61.242.215.32 **não pertence** à rede 61.242.212.128/25
- iii. O endereço 147.85.83.118 **pertence** à rede 147.85.80.0/22
- iv. O endereço 18.255.27.195 **pertence** à rede 18.255.0.0/18
- v. O endereço 74.245.56.207 **pertence** à rede 74.244.0.0/14
- vi. O endereço 84.103.159.207 **pertence** à rede 84.0.0.0/9
- vii. O endereço 1.78.149.201 **pertence** à rede 1.72.0.0/13
- viii. O endereço 49.247.184.30 **não pertence** à rede 49.244.0.0/17
- ix. O endereço 123.133.7.124 **pertence** à rede 123.133.0.0/17
- x. O endereço 112.5.147.149 **não pertence** à rede 100.0.0.0/6

(b) Nos itens a seguir, serão apresentados dois endereços, correspondentes a duas redes distintas. Determine, em cada caso, se uma das redes é uma subrede da outra ou não. Em caso positivo, identifique a subrede maior e a subrede menor.

- i. 59.0.0.0/11 **é disjunta de** 59.132.0.0/14
- ii. 90.90.192.0/19 **é disjunta de** 90.93.0.0/16
- iii. 132.0.0.0/9 **é disjunta de** 130.0.0.0/7
- iv. 89.30.192.0/18 **está contida em** 89.30.0.0/15
- v. 90.166.231.0/24 **é disjunta de** 90.166.228.0/27
- vi. 248.96.0.0/12 **é disjunta de** 252.0.0.0/7
- vii. 176.0.0.0/7 **contém** 176.0.0.0/10
- viii. 185.192.0.0/13 **contém** 185.197.128.0/18
- ix. 152.77.128.0/17 **é disjunta de** 152.79.160.0/19

x. 105.243.40.0/22 **é disjunta de** 105.243.37.128/26

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.

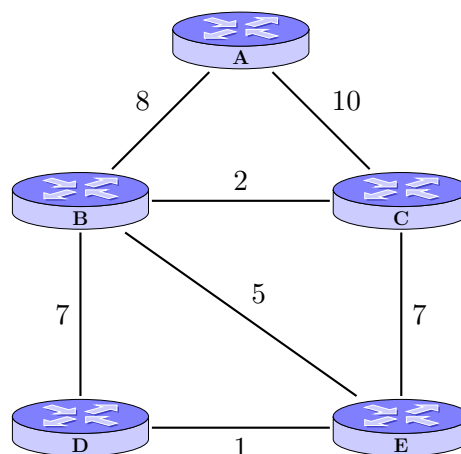
Considere um roteador que interconecta diferentes subredes associadas às suas interfaces. Cada item a seguir ilustra diferentes configurações para o roteador, indicando o endereço de rede que deve ser dividido, o número de subredes a serem criadas (uma para cada interface do roteador), e o número de 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.¹

i. R_1 : 96.0.0.0/7 com 3 subredes: R_{11} : 3000000 estações 96.128.0.0/10 R_{12} : 2000000 estações 96.192.0.0/11 R_{13} : 4000000 estações 96.0.0.0/9	ii. R_2 : 141.189.53.0/24 com 5 subredes: R_{21} : 20 estações R_{22} : 30 estações R_{23} : 60 estações R_{24} : 50 estações R_{25} : 60 estações Alocação impossível	iii. R_3 : 137.32.56.0/23 com 4 subredes: R_{31} : 100 estações 137.32.56.0/25 R_{32} : 70 estações 137.32.56.128/25 R_{33} : 40 estações 137.32.57.0/26 R_{34} : 30 estações 137.32.57.64/26
--	--	--

Questão 3 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.



¹Dica: aloque os endereços das subredes em ordem decrescente de seus tamanhos (i.e., maior primeiro).

- (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	8	∞	∞	∞	via	A	8	∞	∞	∞	via	A	10	∞	∞	∞	via	B	∞	7	∞	∞	via	B	∞	5	∞	∞
	C	∞	10	∞	∞		C	∞	2	∞	∞		B	∞	2	∞	∞		E	∞	∞	∞	1		C	∞	∞	7	∞
					D		∞	∞	7	∞	E		∞	∞	∞	7						D	∞		∞	∞	1		
					E		∞	∞	∞	5																			

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	8	10	∞	∞	8	0	2	7	5	10	2	0	∞	7	∞	7	∞	0	1	∞	5	7	1	0

- (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	8	10	15	13	via	A	8	18	∞	∞	via	A	10	18	∞	∞	via	B	15	7	9	12	via	B	13	5	7	12
	C	12	10	∞	17		C	12	2	∞	9		B	10	2	9	7		E	∞	6	8	1		C	17	9	7	∞
					D		∞	∞	7	8	E		∞	12	8	7						D	∞		8	∞	1		
					E		∞	12	6	5																			

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	8	10	15	13	8	0	2	6	5	10	2	0	8	7	15	6	8	0	1	13	5	7	1	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	8	10	14	13	via	A	8	18	23	21	via	A	10	18	25	23	via	B	15	7	9	12	via	B	13	5	7	11
	C	12	10	18	17		C	12	2	10	9		B	10	2	8	7		E	14	6	8	1		C	17	9	7	15
					D		22	15	7	8	E		20	12	8	7						D	16		7	9	1		
					E		18	12	6	5																			

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	8	10	14	13	8	0	2	6	5	10	2	0	8	7	14	6	8	0	1	13	5	7	1	0

Passo 3:

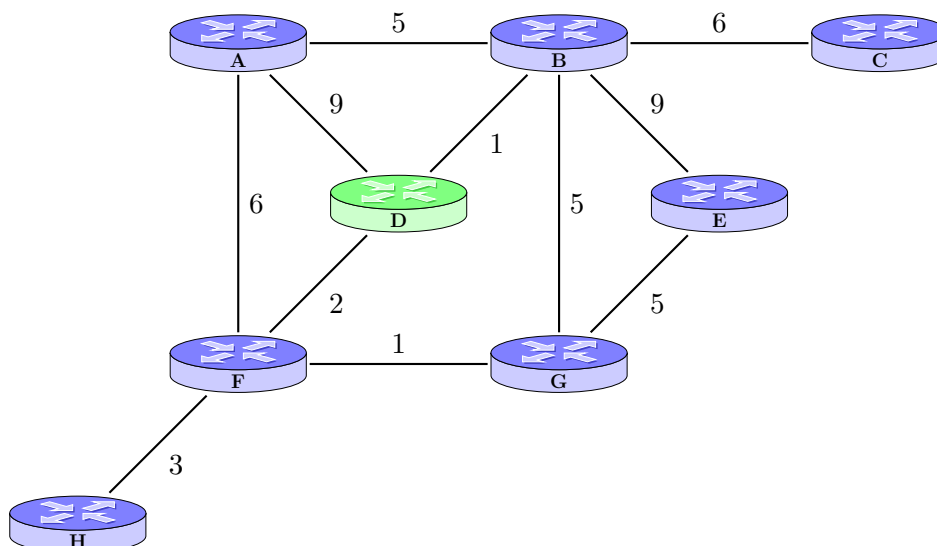
A		custo até				B	custo até				C	custo até				D	custo até				E	custo até							
		B	C	D	E		A	C	D	E		A	B	D	E		A	B	C	E		A	B	C	D				
via	B	8	10	14	13	via	A	8	18	22	21	via	A	10	18	24	23	via	B	15	7	9	12	via	B	13	5	7	11
	C	12	10	18	17		C	12	2	10	9		B	10	2	8	7		E	14	6	8	1		C	17	9	7	15
					D		21	15	7	8	E		20	12	8	7						D	15		7	9	1		
					E		18	12	6	5																			

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	8	10	14	13	8	0	2	6	5	10	2	0	8	7	14	6	8	0	1	13	5	7	1	0

Questão 4..... 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.



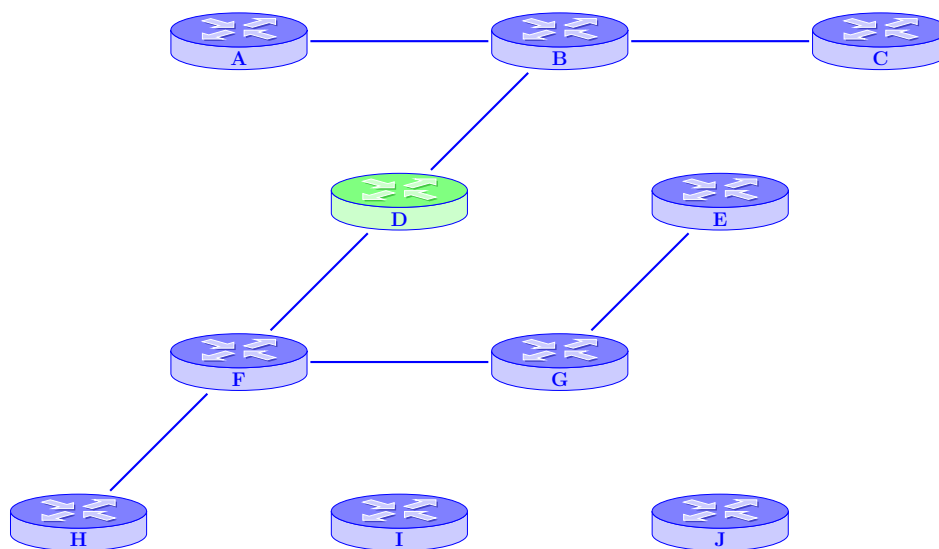
- (a) Utilizando o algoritmo de Dijkstra, calcule os caminhos mais curtos a partir do nó D, destacado em verde, para todos os outros nós da rede. Construa uma tabela igual à mostrada em aula que mostra o funcionamento do algoritmo de forma iterativa.

Resposta:

	N'	d _A PA	d _B PB	d _C PC	d _E PE	d _F PF	d _G PG	d _H PH
0	D	9 D	1 D	∞ -	∞ -	2 D	∞ -	∞ -
1	DB	6 B		7 B	10 B	2 D	6 B	∞ -
2	DBF	6 B		7 B	10 B		3 F	5 F
3	DBFG	6 B		7 B	8 G			5 F
4	DBFGH	6 B		7 B	8 G			
5	DBFGHA			7 B	8 G			
6	DBFGHAC				8 G			
7	DBFGHACE							

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

Resposta:



FALTA A TABELA DE ROTEAMENTO!!

Questão 5 20 pontos

Considere um mecanismo NAT cujo endereço IP na rede pública é 163.98.0.146 e que gerencia as conexões da rede privada, que ocupa a faixa 172.16.0.0/12. Suponha que o NAT possui a seguinte tabela de tradução de endereços, onde cada regra é identificada por um número:

	(IP, porta) da estação local	(IP, porta) da estação remota	Porta pública no NAT
(1)	172.16.0.1, 20337	78.87.118.77, 29010	16039
(2)	172.16.0.1, 10940	106.166.4.149, 26212	25841
(3)	172.16.0.1, 15223	103.124.93.77, 32432	1028
(4)	172.16.0.1, 2389	165.69.139.82, 1923	24910
(5)	172.16.0.2, 5580	57.239.78.89, 32416	26683
(6)	172.16.0.1, 14244	64.123.243.180, 6333	31933
(7)	172.16.0.2, 19502	175.81.48.30, 15451	16010
(8)	172.16.0.2, 8854	202.225.146.157, 14525	20770
(9)	172.16.0.1, 25710	139.219.155.129, 22196	24619
(10)	172.16.0.2, 16558	147.201.42.142, 30046	18236

- (a) Considere que o NAT irá receber uma sequência de pacotes provenientes da rede pública (cuja estação de destino está na rede privada), cujos endereços e portas de origem e destino estão identificados a seguir. Determine se estes pacotes serão encaminhados à rede privada e, em caso positivo, quais serão os endereços e portas de origem e destino que o pacote conterá quando for encaminhado.

- i. Origem: 103.124.93.77, 32432; Destino: 163.98.0.146, 24619
descartado
- ii. Origem: 56.152.130.124, 27475; Destino: 163.98.0.146, 26683
descartado
- iii. Origem: 0.182.119.32, 14846; Destino: 163.98.0.146, 24910
descartado
- iv. Origem: 103.124.93.77, 32432; Destino: 163.98.0.146, 1028
→ **Origem: 103.124.93.77, 32432; Destino: 172.16.0.1, 15223**
- v. Origem: 147.201.42.142, 30046; Destino: 163.98.0.146, 24619
descartado
- vi. Origem: 21.8.150.247, 19631; Destino: 163.98.0.146, 24910
descartado
- vii. Origem: 202.225.146.157, 14525; Destino: 163.98.0.146, 20770
→ **Origem: 202.225.146.157, 14525; Destino: 172.16.0.2, 8854**
- viii. Origem: 64.123.243.180, 6333; Destino: 163.98.0.146, 31933
→ **Origem: 64.123.243.180, 6333; Destino: 172.16.0.1, 14244**
- ix. Origem: 41.10.177.147, 16288; Destino: 163.98.0.146, 31933
descartado
- x. Origem: 139.219.155.129, 22196; Destino: 163.98.0.146, 24619
→ **Origem: 139.219.155.129, 22196; Destino: 172.16.0.1, 25710**

- (b) Considere agora a seguinte sequência de pacotes TCP que chegam, nesta ordem, ao NAT provenientes da rede privada (cuja estação de destino está na rede pública). Determine quais destes pacotes levarão à criação de novas entradas na tabela de tradução. Determine também os endereços e portas, de origem e de destino, de todos os pacotes após eles serem encaminhados à rede pública.

- i. Origem: 172.16.0.1, 25710; Destino: 139.219.155.129, 32078

Cria nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(11)	172.16.0.1, 25710	139.219.155.129, 32078	1024

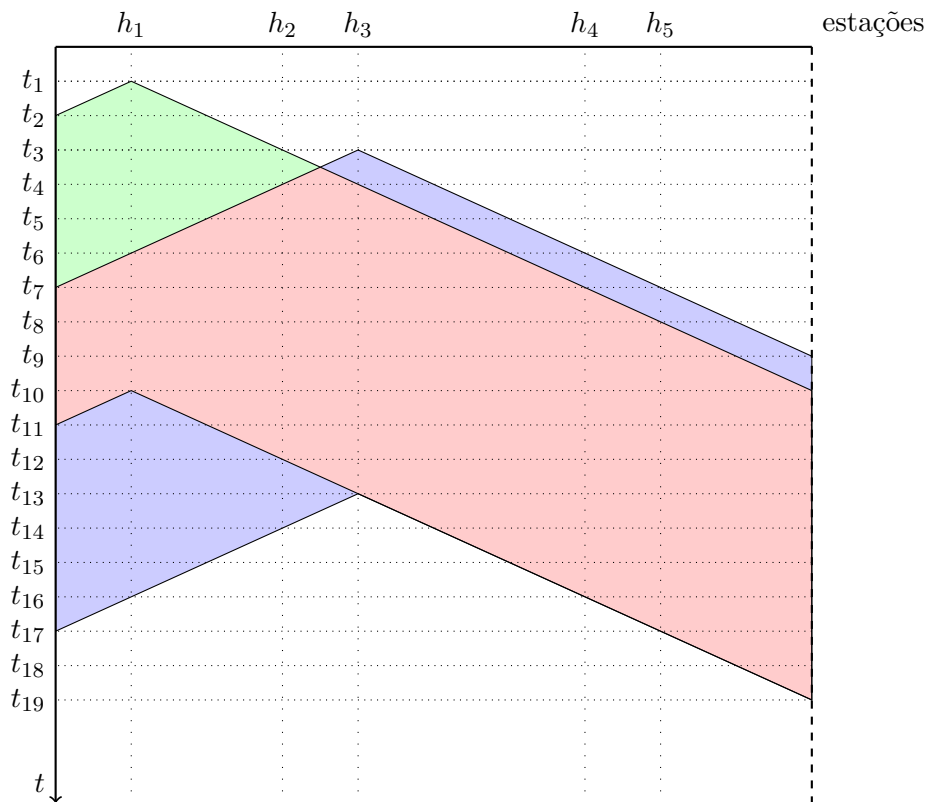
→ **Origem: 163.98.0.146, 1024; Destino: 139.219.155.129, 32078**

- ii. Origem: 172.16.0.2, 16558; Destino: 147.201.42.142, 30046
Não cria nova entrada
→ **Origem: 163.98.0.146, 18236; Destino: 147.201.42.142, 30046**
- iii. Origem: 172.16.0.1, 14244; Destino: 64.123.243.180, 6333
Não cria nova entrada
→ **Origem: 163.98.0.146, 31933; Destino: 64.123.243.180, 6333**
- iv. Origem: 172.16.0.1, 14244; Destino: 64.123.243.180, 6333
Não cria nova entrada
→ **Origem: 163.98.0.146, 31933; Destino: 64.123.243.180, 6333**
- v. Origem: 172.16.0.2, 5580; Destino: 57.239.78.89, 12990
Cria nova entrada:
- | | (IP, porta) local | (IP, porta) destino | Porta pública no NAT |
|------|-------------------|---------------------|----------------------|
| (12) | 172.16.0.2, 5580 | 57.239.78.89, 12990 | 1025 |
- **Origem: 163.98.0.146, 1025; Destino: 57.239.78.89, 12990**
- vi. Origem: 172.16.0.1, 14244; Destino: 64.123.243.180, 10365
Cria nova entrada:
- | | (IP, porta) local | (IP, porta) destino | Porta pública no NAT |
|------|-------------------|-----------------------|----------------------|
| (13) | 172.16.0.1, 14244 | 64.123.243.180, 10365 | 1026 |
- **Origem: 163.98.0.146, 1026; Destino: 64.123.243.180, 10365**
- vii. Origem: 172.16.0.1, 10940; Destino: 106.166.4.149, 1983
Cria nova entrada:
- | | (IP, porta) local | (IP, porta) destino | Porta pública no NAT |
|------|-------------------|---------------------|----------------------|
| (14) | 172.16.0.1, 10940 | 106.166.4.149, 1983 | 1027 |
- **Origem: 163.98.0.146, 1027; Destino: 106.166.4.149, 1983**
- viii. Origem: 172.16.0.1, 30619; Destino: 103.124.93.77, 32432
Cria nova entrada:
- | | (IP, porta) local | (IP, porta) destino | Porta pública no NAT |
|------|-------------------|----------------------|----------------------|
| (15) | 172.16.0.1, 30619 | 103.124.93.77, 32432 | 1029 |
- **Origem: 163.98.0.146, 1029; Destino: 103.124.93.77, 32432**
- ix. Origem: 172.16.0.1, 20337; Destino: 78.87.118.77, 29010
Não cria nova entrada
→ **Origem: 163.98.0.146, 16039; Destino: 78.87.118.77, 29010**
- x. Origem: 172.16.0.1, 25710; Destino: 139.219.155.129, 22196
Não cria nova entrada
→ **Origem: 163.98.0.146, 24619; Destino: 139.219.155.129, 22196**

Questão 6 20 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_1 iniciou sua transmissão primeiro, utilizando o meio entre os instantes de tempo t_1 e t_{10} . Depois disto, a estação h_3 transmite dados entre t_3 e t_{13} .

- (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_1	Transmissão de h_3
h_2	$t_3 - t_{12}$	$t_4 - t_{14}$
h_4	$t_7 - t_{16}$	$t_6 - t_{16}$
h_5	$t_8 - t_{17}$	$t_7 - t_{17}$

- (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:

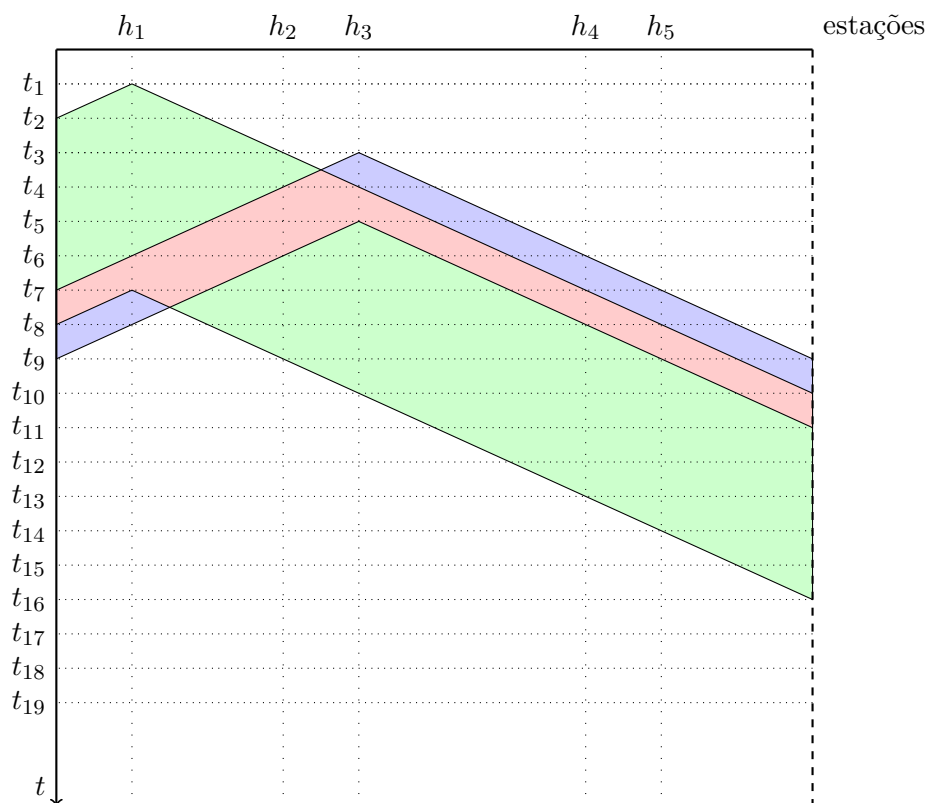
Estação	Vê meio ocupado entre:	Detecta colisão em:
h_1	$t_1 - t_{16}$	t_6
h_2	$t_3 - t_{14}$	t_4
h_3	$t_3 - t_{13}$	t_4
h_4	$t_6 - t_{16}$	t_7
h_5	$t_7 - t_{17}$	t_8

- (e) Considere os instantes de tempo t_2 , t_5 e t_{16} . Quais estações podem iniciar uma transmissão nestes instantes de tempo?

Resposta:

	Estações:
t_2	h_2, h_3, h_4, h_5
t_5	h_4, h_5
t_{16}	h_2, h_3

- (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_1 e h_3 transmitiram. Desta vez, a estação h_1 transmitiu entre os instantes de tempo t_1 e t_7 , e a estação h_3 transmitiu entre os instantes de tempo t_3 e t_5 . As demais estações recebem estas transmissões conforme a tabela a seguir:

Estação	Transmissão de h_1	Transmissão de h_3
h_2	$t_3 - t_9$	$t_4 - t_6$
h_4	$t_7 - t_{13}$	$t_6 - t_8$
h_5	$t_8 - t_{14}$	$t_7 - t_9$

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	$t_1 - t_8$	t_6
h_2	$t_3 - t_9$	t_4
h_3	$t_3 - t_{10}$	t_4
h_4	$t_6 - t_{13}$	t_7
h_5	$t_7 - t_{14}$	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_2	h_2, h_3, h_4, h_5
t_5	h_4, h_5
t_{16}	h_1, h_2, h_3, h_4, h_5

- (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	$t_8 - t_{16}$
h_2	$t_9 - t_{14}$
h_3	$t_{10} - t_{13}$
h_4	$t_{13} - t_{16}$
h_5	$t_{14} - t_{17}$