

Curso de Tecnologia em Sistemas de Computação
Disciplina: Redes de Computadores II
AD1 – 1º semestre de 2015 – GABARITO

Questão 0 5 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.

Resposta:

Para resolver este exercício, você deve perceber que o endereço de rede apresentado determina um prefixo em notação binária. Os endereços IP desta rede são precisamente os que possuem este mesmo prefixo.

i. O endereço 132.192.35.137 **não pertence** à rede 132.192.16.0/20

Resposta:

O endereço IP da rede apresentada, 132.192.16.0, possui forma binária:
10000100 11000000 00010000 00000000

A máscara de rede /20 indica que os primeiros 20 bits formam o prefixo de subrede, ou seja, temos o prefixo:
10000100 11000000 0001

O endereço 132.192.35.127 possui forma binária:
10000100 11000000 00100011 01111111

cujo prefixo de subrede (primeiros 20 bits) é:
10000100 11000000 0010

Note que os prefixos de subrede deste endereço e do endereço da rede são diferentes, logo o endereço não pertence à rede apresentada.
A resolução dos itens restantes é análoga.

ii. O endereço 140.171.187.181 **não pertence** à rede 140.171.0.0/17

iii. O endereço 96.147.58.113 **não pertence** à rede 96.147.0.0/19

iv. O endereço 246.253.82.89 **pertence** à rede 246.253.0.0/16

v. O endereço 255.39.65.53 **pertence** à rede 255.0.0.0/8

vi. O endereço 10.86.185.248 **pertence** à rede 10.0.0.0/9

- vii. O endereço 14.96.5.230 **pertence** à rede 14.0.0.0/9
 - viii. O endereço 242.42.2.40 **não pertence** à rede 242.44.0.0/16
 - ix. O endereço 84.69.2.72 **não pertence** à rede 84.69.1.0/25
 - x. O endereço 139.181.28.192 **pertence** à rede 139.0.0.0/8
- (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.

Resposta:

Como no item anterior, os endereços de cada rede determinam dois prefixos de subrede. Se o prefixo mais curto (rede maior) for igual ao início do prefixo mais longo (rede menor), então todos os endereços da rede menor também pertencem à rede maior, ou seja, uma rede é subrede da outra. Caso contrário, nenhum endereço da rede menor pertence à rede maior, ou seja, as duas redes são disjuntas.

- i. 252.0.0.0/8 **é disjunta de** 253.16.0.0/12

Resposta:

A primeira rede, 252.0.0.0/8, possui prefixo:

11111100

enquanto a segunda rede, 253.16.0.0/12, possui prefixo:

11111101 0001

Note que o prefixo da primeira rede não corresponde ao início do prefixo da segunda, logo elas são redes disjuntas.

Novamente, a resolução dos itens restantes é análoga.

- ii. 250.0.0.0/7 **contém** 251.128.0.0/9
- iii. 43.163.62.0/29 **é disjunta de** 43.163.60.128/25
- iv. 10.0.0.0/14 **é disjunta de** 10.18.0.0/18
- v. 237.70.118.128/25 **é disjunta de** 237.70.116.0/28
- vi. 20.179.128.0/17 **contém** 20.179.136.0/21
- vii. 179.0.0.0/8 **está contida em** 176.0.0.0/6
- viii. 85.106.144.0/20 **está contida em** 85.106.0.0/16
- ix. 85.159.64.0/19 **é disjunta de** 85.159.135.0/24
- x. 53.148.0.0/22 **é disjunta de** 53.148.12.0/24

Questão 1 15 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
11011100 01000000 00000000 00000000 a	0
11011100 01111111 11111111 11111111 11011100 01000000 00000000 00000000 a	1
11011100 01001111 11111111 11111111 11011100 01000011 10000000 00000000 a	2
11011100 01000011 11111111 11111111 11011100 01111000 00000000 00000000 a	0
11011100 01111111 11111111 11111111 11011100 01010000 00000000 00000000 a	1
11011100 01011111 11111111 11111111 11011100 01111100 00000000 00000000 a	3
11011100 01111100 11111111 11111111 caso contrário	2

- (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:

Em qualquer subrede, o prefixo é o conjunto de bits iniciais que determina se um endereço IP qualquer pertence à subrede em questão ou não; ou seja, é o conjunto de bits que é comum a todos os endereços da subrede, e que nenhum endereço fora da subrede pode possuir. Para resolver esta questão, basta identificar estes bits. Note que cada faixa de endereços apresentada na tabela segue este formato: um conjunto de bits que todos os endereços são forçados a ter, seguido de todas as combinações possíveis para os bits restantes. Logo, todas são subredes válidas. Reescrevendo a tabela com os prefixos identificados:

Prefixo	Interface
11011100 01	0
11011100 0100	1
11011100 01000011 1	2
11011100 01111	0
11011100 0101	1
11011100 01111100	3
-	2

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

Resposta:

A notação a.b.c.d/x é utilizada para representar subredes. “x” corresponde ao comprimento da máscara de subrede, isto é, a quantidade de bits utilizados para definir a subrede, enquanto “a.b.c.d” correspondente ao endereço de subrede. O valor de “x” pode ser obtido diretamente do prefixo, pois é igual ao comprimento do prefixo.

Para determinar os valores de “a.b.c.d”, é preciso identificar o primeiro endereço IP desta subrede em forma binária, e depois convertê-lo para a forma decimal. Este endereço é obtido completando o prefixo com 0’s ao final até completar o comprimento de um endereço IP (32 bits).

Exemplificaremos este cálculo para o primeiro prefixo:

11011100 01

O primeiro endereço desta subrede é:

11011100 01000000 00000000 00000000

Convertendo o endereço octeto a octeto para a forma decimal, temos:

11011100 = 220

01000000 = 64

00000000 = 0

00000000 = 0

Como o comprimento deste prefixo é de 10 bits, o endereço de subrede obtido é 220.64.0.0/10.

Para as outras faixas, deve ser realizado um procedimento semelhante, obtendo a seguinte tabela:

Prefixo	Interface
220.64.0.0/10	0
220.64.0.0/12	1
220.67.128.0/17	2
220.120.0.0/13	0
220.80.0.0/12	1
220.124.0.0/16	3
0.0.0.0/0	2

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

Resposta:

Duas regras devem ser observadas na hora de determinar a interface para a qual este pacote deve ser encaminhado. A primeira é que ele deve pertence à subrede delimitada por alguma das regras da tabela encaminhamento. Se houver mais de uma subrede possível, aplica-se a segunda regra: a regra do *prefixo mais longo*, que determina que prefixos mais longos (com mais bits, correspondendo a subredes menores) têm prioridade sobre prefixos mais curtos.

Considere o primeiro item, por exemplo. O endereço 220.88.238.159 pertence às subredes 220.64.0.0/10 e 220.80.0.0/12 (para ver isto, realize a expansão binária destes endereços), portanto só as regras da tabela de encaminhamento correspondentes a estas subredes podem ser utilizadas. Pela regra do prefixo mais longo, escolhe-se a regra correspondente à rede 220.80.0.0/12, logo o pacote será encaminhado pela interface 1.

O mesmo procedimento vale para os demais endereços.

- | | |
|---|---|
| i. 220.88.238.159 → interface 1 | vii. 220.124.43.224 → interface 3 |
| ii. 220.67.179.127 → interface 2 | viii. 220.81.216.219 → interface 1 |
| iii. 220.79.23.136 → interface 1 | ix. 220.138.216.229 → interface 2 |
| iv. 220.176.75.212 → interface 2 | x. 220.104.238.191 → interface 0 |
| v. 220.107.188.118 → interface 0 | xi. 220.67.141.168 → interface 2 |
| vi. 220.87.1.14 → interface 1 | xii. 220.126.161.225 → interface 0 |

- (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:

Uma regra é redundante se ela puder ser removida da tabela sem afetar o encaminhamento dos pacotes. Isto significa que, se ela for removida, todos os pacotes que seriam encaminhados por essa regra, e que agora vão ser incorporados por uma segunda regra, continuam sendo encaminhados pela mesma interface.

Neste exemplo, somente a regra do prefixo 220.120.0.0/13 é redundante. Isto porque todos os pacotes que são encaminhados segundo essa regra também satisfazem a regra da subrede 220.64.0.0/10 (e mais nenhuma outra). Mas as duas determinam que o pacote deve seguir pela interface 0, logo, se a regra 220.120.0.0/13 for removida da tabela, todos os pacotes afetados vão passar a ser encaminhados segundo a regra 220.64.0.0/10 e vão continuar sendo encaminhados pela mesma interface 0.

Questão 2 5 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.

¹Dica: procure por faixas mais específicas.

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

Resposta:

Há diversas formas de resolver este exercício, mas para todas elas alguns detalhes devem ser observados:

1. Todas as subredes que desejamos endereçar devem receber endereços de subrede com prefixos distintos; caso contrário, uma rede estará contida dentro da outra;
2. Todas devem estar contidas na subrede principal original, pois é a faixa de endereços que está disponível para distribuir entre as subredes;
3. Como cada subrede deve comportar um número mínimo de estações, cada uma tem também um comprimento máximo de prefixo permitido.

Vamos exemplificar a resolução com o primeiro item, em que nossa subrede principal R_1 tem endereço 110.0.0.0/8. Escolhemos a estratégia de endereçar primeiro as subredes maiores e depois as menores. Outras estratégias podem ser usadas, mas esta evita que desperdicemos endereços sem perceber.

Como nossa rede principal possui um prefixo de 8 bits, todas as nossas subredes devem possuir estes mesmos bits como parte do seu próprio prefixo. Na forma binária, então, todas as redes terão um endereço iniciado por 01101110.

A maior das nossas subredes desejadas é R_{13} , com 3100000 estações. Como $2^{21} < 3100000 < 2^{22}$ (pois $2^{21} = 2097152$ e $2^{22} = 4194304$), isto significa que precisaremos de, pelo menos, 22 bits de estação para endereçar todas estas estações. Restam 10 bits para serem utilizados como prefixo desta subrede. Os 8 primeiros bits já estão definidos, logo só precisamos escolher os bits 9 e 10: escolhemos os valores 00 para estes dois bits. A subrede R_{13} terá, portanto, prefixo 01101110 00 e endereço 110.0.0.0/10.

Note que poderíamos utilizar só 9 bits no prefixo desta subrede, mas isto pode nos atrapalhar no futuro para endereçar as outras estações, pois estaríamos “desperdiçando” endereços com esta subrede.

Seguimos o mesmo processo com as outras redes, tomando o cuidado de escolher um prefixo que não esteja contido nos prefixos das outras subredes:

R_{11} : 2700000 estações; ao menos 22 bits de estação, restam 10 bits para prefixo;

Prefixo escolhido: 01101110 01; endereço 110.64.0.0/10

R_{14} : 1200000 estações; ao menos 21 bits de estação, restam 11 bits para prefixo;

Prefixo escolhido: 01101110 100; endereço 110.128.0.0/11

R_{12} : 800000 estações; ao menos 20 bits de estação, restam 12 bits para prefixo;

Prefixo escolhido: 01101110 1010; endereço 110.160.0.0/12

Se, em alguma subrede, tivermos esgotado todos os prefixos disponíveis, isto significa que é impossível realizar a alocação desejada. É o que ocorre no segundo item, por exemplo.

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

i. R_1 : 110.0.0.0/8 com 4 subredes: R_{11} : 2700000 estações 110.64.0.0/10 R_{12} : 800000 estações 110.160.0.0/12 R_{13} : 3100000 estações 110.0.0.0/10 R_{14} : 1200000 estações 110.128.0.0/11	ii. R_2 : 220.154.192.0/19 com 5 subredes: R_{21} : 1700 estações R_{22} : 1100 estações R_{23} : 1700 estações R_{24} : 1300 estações R_{25} : 500 estações Alocação impossível	iii. R_3 : 32.0.0.0/7 com 4 subredes: R_{31} : 5000000 estações 32.0.0.0/9 R_{32} : 2000000 estações 32.192.0.0/10 R_{33} : 4000000 estações 32.128.0.0/10 R_{34} : 2000000 estações 33.0.0.0/11
---	--	---

Questão 3 10 pontos

Considere o problema de encaminhamento de pacotes em uma rede baseada em circuitos virtuais. Neste tipo de rede os roteadores possuem tabelas de roteamento que mapeiam a interface de entrada e o número de circuito em uma interface de saída e outro número de circuito virtual. Os pacotes de uma determinada conexão carregam o número do circuito virtual do enlace sendo atravessado, que é atualizado pelo roteador ao encaminhá-lo. O objetivo desta questão é entender como funcionam circuitos virtuais.

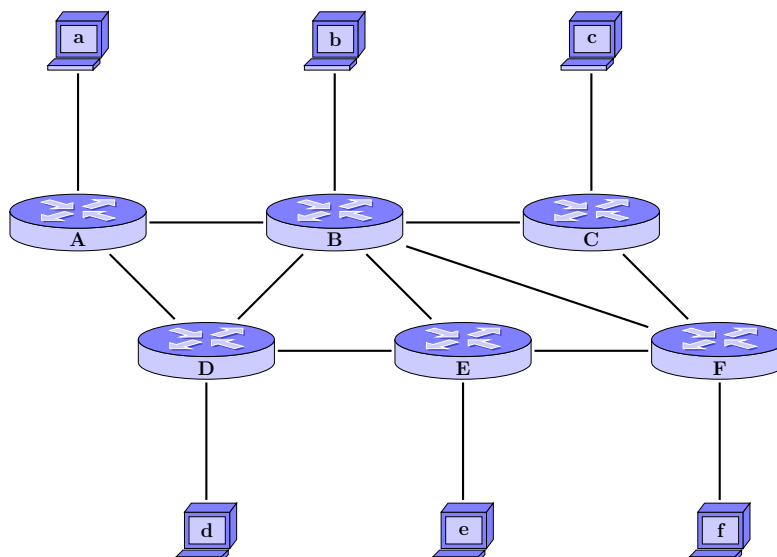


Tabela de A	
Entrada	Saída
a, 1	B, 8
D, 9	B, 6
a, 3	D, 6
D, 4	a, 6
B, 8	a, 8

Tabela de B	
Entrada	Saída
F, 3	E, 5
E, 7	A, 8
E, 4	b, 8
A, 6	F, 4
A, 8	F, 8
C, 4	D, 8
D, 8	b, 5

Tabela de C	
Entrada	Saída
c, 7	B, 4
c, 5	F, 1

Tabela de D	
Entrada	Saída
d, 4	A, 9
d, 5	A, 4
E, 5	d, 3
d, 7	B, 8
B, 8	E, 2
A, 6	d, 2

Tabela de E	
Entrada	Saída
F, 3	e, 2
e, 9	B, 4
e, 6	B, 7
e, 8	D, 5
F, 1	e, 7
B, 6	F, 9
D, 2	e, 9
B, 5	e, 1

Tabela de F	
Entrada	Saída
f, 3	B, 3
E, 9	f, 9
C, 1	E, 3
B, 8	f, 7
C, 3	E, 1
B, 4	f, 3

Considere a rede anterior, composta por estações e roteadores, e as tabelas de roteamento destes roteadores. Note que todo enlace de saída na tabela de roteamento corresponde a um enlace de entrada na tabela do roteador do outro lado do enlace. Por exemplo, a saída E,3 na tabela do roteador A corresponde à entrada A,3 na tabela do roteador E.

- (a) Em cada item a seguir, será apresentado um pacote, identificado por sua estação e circuito virtual de origem. Determine o caminho que estes pacotes irão percorrer pela rede. Em particular, determine a sequência de enlaces que cada pacote irá atravessar e, para cada enlace, o número do circuito virtual que o pacote irá conter ao atravessá-lo.

Resposta:

Em roteamento por circuito virtual, os circuitos não possuem números únicos que os identifiquem na rede. Em vez disso, em cada enlace que ele atravessar, os dois roteadores ligados por esse enlace devem concordar em um número de CV para identificar cada circuito naquele ponto da rede. No entanto, em enlaces diferentes, o circuito pode possuir números de CV diferentes.

Isto significa que, sempre que um roteador encaminhar um pacote por um circuito, ele deve reescrever o número de CV do pacote, substituindo o número “antigo” (do enlace de chegada) e colocando no lugar o número “novo” (do enlace de saída). Assim, os roteadores são capazes de encaminhar os pacotes até seu destino, mesmo sem saber quem é o destino.

Vamos exemplificar este processo com o primeiro item desta questão. A estação e envia um pacote para o seu gateway (o roteador E) através do circuito virtual de número 9. Ocorrem, então, os seguintes passos:

- O roteador E recebe o pacote e consulta a sua tabela de roteamento. Uma das entradas indica que pacotes recebidos da estação e, com número de CV igual a 9, devem ser encaminhados para o roteador B, com número de CV 4. O roteador E substitui o número de CV e encaminha o pacote para B.
- O roteador B recebe o pacote vindo de E pelo número de CV 4. Após consultar a tabela, ele encaminha o pacote para a estação b, com o número de circuito virtual 8.
- A estação b recebe o pacote vindo de B pelo número de CV 8. Sendo uma estação, que não realiza encaminhamento, b se reconhece como destino do pacote e recebe-o.

Desta forma, o circuito percorrido pelo pacote, com os respectivos números de circuito virtual denotados, é:

$$e \xrightarrow{9} E \xrightarrow{4} B \xrightarrow{8} b$$

- i. (e, 9) : $e \xrightarrow{9} E \xrightarrow{4} B \xrightarrow{8} b$
- ii. (d, 7) : $d \xrightarrow{7} D \xrightarrow{8} B \xrightarrow{5} b$
- iii. (d, 4) : $d \xrightarrow{4} D \xrightarrow{9} A \xrightarrow{6} B \xrightarrow{4} F \xrightarrow{3} f$
- iv. (d, 5) : $d \xrightarrow{5} D \xrightarrow{4} A \xrightarrow{6} a$
- v. (c, 5) : $c \xrightarrow{5} C \xrightarrow{1} F \xrightarrow{3} E \xrightarrow{2} e$
- vi. (c, 7) : $c \xrightarrow{7} C \xrightarrow{4} B \xrightarrow{8} D \xrightarrow{2} E \xrightarrow{9} e$

- (b) Considere agora que desejamos criar novos circuitos virtuais, indicados nos itens a seguir. Determine o número de CV que será utilizado em cada enlace destes novos circuitos. Considere que as numerações utilizadas nos dois sentidos de cada enlace são independentes, e que os circuitos criados em cada item continuam existindo nos itens seguintes. Note que há mais de uma resposta correta.

Resposta:

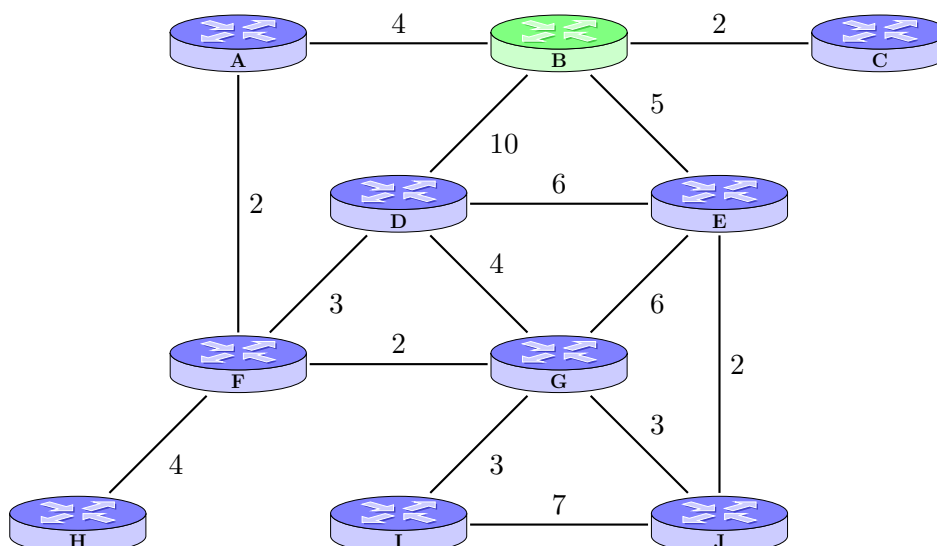
Como, nesta questão, só se pede para escolher números de CV para os circuitos a serem criados, basta escolher, para cada circuito e cada enlace, um número de CV, tomando o cuidado de não escolher um número que já esteja sendo usado por outro CV. Por exemplo: no primeiro item, o circuito não pode utilizar os números 6, 8 e 9 no enlace que liga a estação e ao roteador E, pois estes números já estão sendo utilizados por outros CVs, como podemos perceber pelas entradas existentes na tabela de roteamento de E. As soluções abaixo são respostas corretas, mas qualquer escolha de números de CV que respeite estas regras também estará correta.

- i. $e \xrightarrow{1} E \xrightarrow{1} D \xrightarrow{1} A \xrightarrow{1} a$
- ii. $a \xrightarrow{2} A \xrightarrow{1} D \xrightarrow{1} E \xrightarrow{1} F \xrightarrow{1} f$
- iii. $c \xrightarrow{1} C \xrightarrow{2} F \xrightarrow{2} E \xrightarrow{2} D \xrightarrow{2} A \xrightarrow{2} a$
- iv. $e \xrightarrow{2} E \xrightarrow{3} D \xrightarrow{3} A \xrightarrow{3} a$

Questão 4 15 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:

A solução desta questão consiste em desenvolver cada iteração do algoritmo de Dijkstra no formato de tabela até calcular todos os caminhos mais curtos desejados. Valem alguns lembretes gerais:

- A cada passo, *exatamente* um nó é adicionado ao conjunto N', que contém todos os nós para os quais o caminho mais curto é conhecido. O nó adicionado será *sempre* aquele que ainda não foi adicionado mas cujo caminho (temporário) calculado até o momento pelo algoritmo é menor do que o dos outros nós;
- A cada passo, para procurar novos caminhos, só é preciso verificar novos caminhos que passem pelo último nó a ser adicionado a N';
- A cada passo, sempre que não for descoberto um caminho melhor para algum nó, o caminho já conhecido e registrado na tabela no passo anterior deve ser *mantido*. Lembre-se que, em cada passo, a tabela registra o melhor caminho conhecido até o momento para todos os nós, e o algoritmo apenas vai descobrindo caminhos melhores.

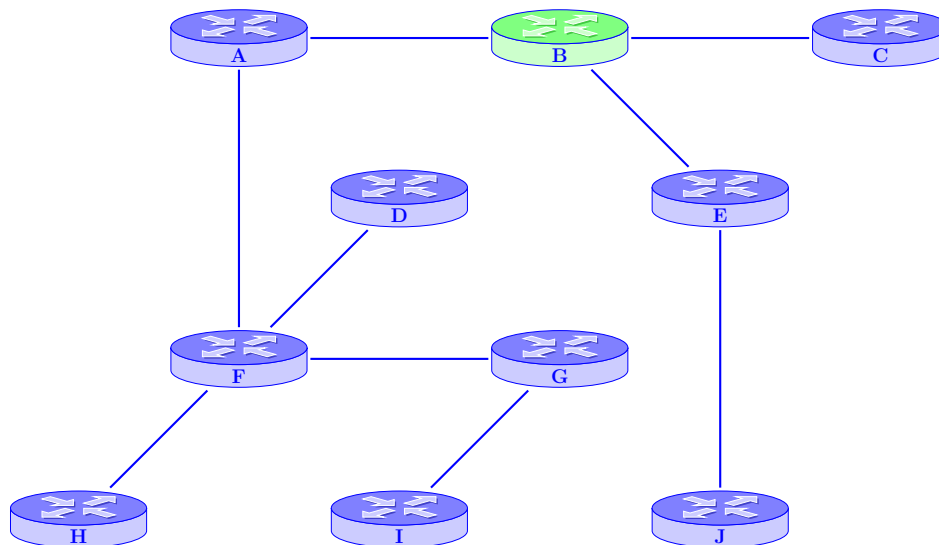
Desenvolvendo a execução iterativa do algoritmo:

	N'	$d_A p_A$	$d_C p_C$	$d_D p_D$	$d_E p_E$	$d_F p_F$	$d_G p_G$	$d_H p_H$	$d_I p_I$	$d_J p_J$
0	B	4 B	2 B	10 B	5 B	∞ -	∞ -	∞ -	∞ -	∞ -
1	BC	4 B		10 B	5 B	∞ -	∞ -	∞ -	∞ -	∞ -
2	BCA			10 B	5 B	6 A	∞ -	∞ -	∞ -	∞ -
3	BCAE			10 B		6 A	11 E	∞ -	∞ -	7 E
4	BCAEF			9 F			8 F	10 F	∞ -	7 E
5	BCAEFJ			9 F			8 F	10 F	14 J	
6	BCAEFJG			9 F				10 F	11 G	
7	BCAEFJGD							10 F	11 G	
8	BCAEFJGDH								11 G	
9	BCAEFJGDHI									

- (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:

Para construir a árvore de caminhos mínimos, basta olhar a tabela da execução iterativa, obtida no item anterior. Cada nó da rede (exceto pelo nó B), possui um antecessor, através do qual o melhor caminho de B para este nó tem que passar. Portanto, a árvore será formada por todos os enlaces entre os nós da rede e seus antecessores:

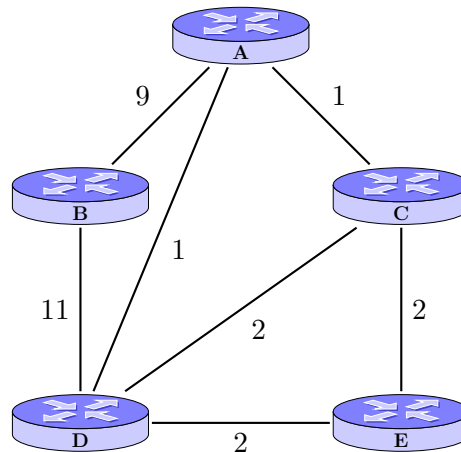


Já para montar a tabela de roteamento, devemos determinar, para cada nó da rede, qual é o *primeiro* enlace no caminho de B até este nó. Para isto, podemos utilizar duas estratégias: (i) utilizamos a tabela do item anterior e, para cada nó, consultamos quem é o seu antecessor, e quem é o antecessor deste, e assim sucessivamente até alcançarmos o nó B; ou (ii) utilizamos a árvore de caminhos mínimos para verificar qual é o caminho que liga cada nó ao nó B, e selecionamos o primeiro enlace deste caminho.

Destino	A	C	D	E	F	G	H	I	J
Enlace de saída	(B,A)	(B,C)	(B,A)	(B,E)	(B,A)	(B,A)	(B,A)	(B,A)	(B,E)

Questão 5 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:

No passo 0, todos os nós somente conhecem seus próprios enlaces. Para o nó A, por exemplo, isto significa que o único caminho que ele conhece até o nó B é via o próprio nó B (caminho direto), com distância igual ao próprio custo do enlace (no caso, 9). A entrada correspondente na tabela de distâncias de A (custo até B, via B) conterá, portanto, o valor 9. O mesmo vale para os outros vizinhos de A. Todas as outras entradas terão valor ∞ , já que os nós ainda não conhecem outros caminhos além dos enlaces diretos. Repetindo o raciocínio para os outros nós, temos as seguintes tabelas de distâncias:

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	9	∞	∞	∞	via	A	9	∞	∞	∞	via	A	1	∞	∞	∞	via	A	1	∞	∞	∞	via	C	∞	∞	2	∞
	C	∞	1	∞	∞		D	∞	∞	11	∞		D	∞	∞	2	∞		B	∞	11	∞	∞		D	∞	∞	∞	2
	D	∞	∞	1	∞								E	∞	∞	∞	2		C	∞	∞	2	∞						

Note que os nós calculam suas próprias tabelas de distância de maneira independente, sem trocar nenhuma informação.

Para calcular o seu vetor de distâncias, basta que cada nó selecione, para cada nó destino, o melhor caminho dentre as opções “via” (a distância para ele próprio é, obviamente, 0). Isto corresponde a selecionar o menor valor de custo para cada coluna da tabela de distâncias, obtendo assim os seguintes vetores.

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	9	1	1	∞	9	0	∞	11	∞	1	∞	0	2	2	1	11	2	0	2	∞	∞	2	2	0

Novamente, o cálculo feito até agora por cada nó é independente dos demais.

- (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:

Neste momento, ocorre a primeira troca de mensagens do algoritmo: todos os nós enviam seus vetores de distância a seus vizinhos. Vamos utilizar novamente os nós A e B como exemplo.

Considere que B enviou seu vetor de distâncias para A. Este vetor indica, por exemplo, que B conhece um caminho de custo 11 até D. Mesmo sem saber que caminho é este, A pode confiar nesta informação enviada por B e, a partir dela, deduzir que existe um caminho até D passando por B. O custo desse caminho é o custo do primeiro passo de A até B, que é igual a 9 (custo do enlace A-B, conhecido por A), mais o custo do restante do caminho de B até D, que é 11 segundo o vetor de distâncias. Logo, A conhece um caminho até D, via B, com custo 20, e registra este valor na entrada correspondente de sua tabela de distâncias.

Repetindo o processo para as outras posições do vetor de distâncias de B, A preenche a linha “via B” da sua tabela de distâncias. Para preencher as outras linhas, ele deve utilizar os vetores de distâncias dos outros vizinhos, e considerar o custo destes enlaces que o ligam a estes vizinhos. Todos os nós realizam este processo, obtendo assim as seguintes tabelas:

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	9	∞	20	∞	via A	9	10	10	∞	via A	1	10	2	∞	via A	1	10	2	∞	via C	3	∞	2	4
via C	∞	1	3	3	via D	12	13	11	13	via D	3	13	2	4	via B	20	11	∞	∞	via D	3	13	4	2
via D	12	3	1	3						via E	∞	∞	4	2	via C	3	∞	2	4					
															via E	∞	∞	4	2					

O cálculo dos novos vetores de distância é feito da mesma forma que no passo anterior:

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	9	1	1	3	9	0	10	10	13	1	10	0	2	2	1	10	2	0	2	3	13	2	2	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. Note que o algoritmo pode convergir mesmo que a tabela de distâncias seja modificada neste passo. O que importa é que o vetor de distâncias permaneça o mesmo.

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	9 19 19 22	via A	9 10 10 12	via A	1 10 2 4	via A	1 10 2 4	via C	3 12 2 4
via C	11 1 3 3	via D	12 13 11 13	via D	3 12 2 4	via B	20 11 21 24	via D	3 12 4 2
via D	11 3 1 3			via E	5 15 4 2	via C	3 12 2 4		
						via E	5 15 4 2		

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 9 1 1 3	9 0 10 10 12	1 10 0 2 2	1 10 2 0 2	3 12 2 2 0

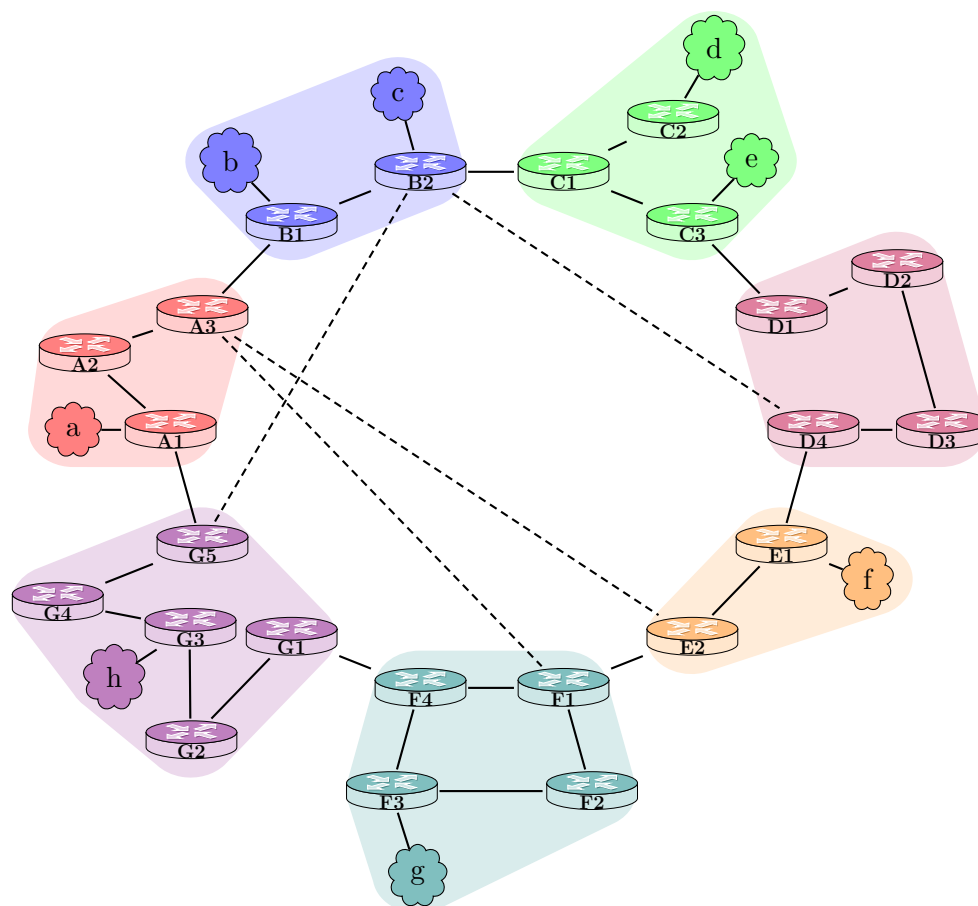
Passo 3:

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	9 19 19 21	via A	9 10 10 12	via A	1 10 2 4	via A	1 10 2 4	via C	3 12 2 4
via C	11 1 3 3	via D	12 13 11 13	via D	3 12 2 4	via B	20 11 21 23	via D	3 12 4 2
via D	11 3 1 3			via E	5 14 4 2	via C	3 12 2 4		
						via E	5 14 4 2		

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 9 1 1 3	9 0 10 10 12	1 10 0 2 2	1 10 2 0 2	3 12 2 2 0

Questão 6 10 pontos

Ao contrário de roteamento dentro de um sistema autônomo (intra-AS), que é orientado a desempenho (encontrar caminhos mínimos pelos quais os pacotes irão transitar), o roteamento entre sistemas autônomos (inter-AS) na Internet é orientado a política de uso. Os sistemas autônomos negociam enlaces entre eles e políticas de utilização para estes enlaces. O objetivo desta questão é compreender o funcionamento do protocolo de roteamento inter-AS utilizado na Internet, o BGP.



Considere a rede a seguir, onde as cores distintas identificam diferentes sistemas autônomos (AS). Nesta rede, há dois tipos de enlaces entre ASs: os enlaces denotados por linhas contínuas, que formam um ciclo entre todos os ASs, compõem o *backbone* da rede, e podem ser utilizados por todo tipo de tráfego. Além disso, alguns ASs negociam *peering links*, “atalhos” diretos entre eles, denotados por linhas tracejadas. Cada um destes enlaces somente pode ser utilizado para comunicações entre estações destes ASs, não sendo permitido tráfego de terceiros. As nuvens, identificadas por letras minúsculas, representam subredes.

- (a) Determine se, entre cada um dos seguintes pares de roteadores, existe comunicação iBGP ou eBGP (ou indique se não houver comunicação BGP entre eles).

Resposta:

As comunicações iBGP e eBGP têm propósitos distintos. As comunicações iBGP servem para que os roteadores de um mesmo sistema autônomo troquem informações entre si, e ocorrem entre todos os roteadores deste sistema autônomo. Já as comunicações eBGP servem para troca de informações BGP entre roteadores de sistemas autônomos distintos, e ocorrem apenas entre roteadores de ASs diferentes que estejam diretamente ligados por um enlace inter-AS.

- | | | |
|----------------------------|-----------------------------|----------------------------|
| i. D1 e G5 : — | v. D1 e A2 : — | ix. A1 e A3 : iBGP |
| ii. A3 e A1 : iBGP | vi. A1 e A2 : iBGP | x. F1 e C2 : — |
| iii. D4 e E1 : eBGP | vii. E2 e A3 : eBGP | xi. D4 e B2 : eBGP |
| iv. A3 e D3 : — | viii. D3 e D4 : iBGP | xii. G1 e G4 : iBGP |

- (b) Considere, agora, que diversos pacotes serão enviados entre estas redes. Em cada item abaixo, são apresentados as subredes de origem e destino de um destes pacotes. Determine quais ASs estes pacotes irão atravessar e, a partir desta informação, determine os roteadores que ele irá atravessar até o seu destino.

Resposta:

Ao contrário dos exercícios anteriores, em que o caminho desejado entre dois roteadores é sempre o de menor custo, no protocolo BGP outras prioridades têm vez. Em primeiro lugar, antes de determinar especificamente quais roteadores irão formar o caminho entre duas estações, é preciso determinar quais ASs este caminho irá atravessar, e isto depende da política utilizada pelos ASs para permitir o uso dos enlaces inter-AS.

Neste exercício, isto significa que um enlace do tipo *peering link* só pode ser utilizado para roteamento se as subredes comunicantes estão cada uma em um dos ASs conectados por este enlace. Se isto não for verdade, o caminho a ser percorrido pelos pacotes terá que seguir através do backbone circular. Neste caso, a prioridade será minimizar o número de ASs atravessados (e não o número de roteadores, por exemplo).

Uma vez determinada a sequência de ASs que irá compor o caminho, também estão determinados os roteadores de borda que farão a ligação entre os ASs. Falta determinar os trechos do caminho que passam por dentro de cada AS — basta escolher os caminhos que passam pelo menor número de roteadores.

- i. $g \rightarrow e$
ASs: F - E - D - C
Caminho: F3 - F2 - F1 - E2 - E1 - D4 - D3 - D2 - D1 - C3
- ii. $d \rightarrow g$
ASs: C - D - E - F
Caminho: C2 - C1 - C3 - D1 - D2 - D3 - D4 - E1 - E2 - F1 - F4 - F3
- iii. $h \rightarrow g$
ASs: G - F
Caminho: G3 - G2 - G1 - F4 - F3
- iv. $a \rightarrow f$
ASs: A - E
Caminho: A1 - A2 - A3 - E2 - E1
- v. $h \rightarrow c$
ASs: G - B
Caminho: G3 - G4 - G5 - B2
- vi. $f \rightarrow d$
ASs: E - D - C
Caminho: E1 - D4 - D3 - D2 - D1 - C3 - C1 - C2
- vii. $b \rightarrow h$
ASs: B - G
Caminho: B1 - B2 - G5 - G4 - G3
- viii. $a \rightarrow b$
ASs: A - B
Caminho: A1 - A2 - A3 - B1

ix. $b \rightarrow e$ **ASs: B - C****Caminho: B1 - B2 - C1 - C3**x. $c \rightarrow d$ **ASs: B - C****Caminho: B2 - C1 - C2**

- (c) Em cada item a seguir, os ASs apresentados devem tomar a decisão de anunciar ou não uma rota até uma determinada subrede para um AS vizinho. Com base nas políticas de utilização descritas anteriormente, determine se os ASs realizarão tais anúncios. Considere que a rota em questão é a rota que o AS anunciante utiliza para enviar seus pacotes para a subrede destino.

Resposta:

Vamos novamente lembrar a política de utilização dos enlaces inter-AS: um *peering link* somente pode ser utilizado para comunicação direta entre dois ASs, enquanto enlaces do backbone podem ser utilizados para comunicação entre quaisquer ASs. Os anúncios de rota são a forma de implementar esta política.

Quando um AS X anuncia a um AS Y, vizinho a ele, a rota que ele conhece até uma subrede z, ele está permitindo que o AS Y utilize o enlace que liga os dois ASs para rotear pacotes até a subrede z. Isto pode ser desejado ou não, dependendo da política que descrevemos anteriormente. Vamos ver dois exemplos:

Exemplo 1: o AS G tem a opção de anunciar sua rota até a subrede f para o AS A. A rota que G conhece para enviar pacotes para a subrede f passa através do backbone da rede, e o enlace que liga G a A é parte deste backbone. Se A enviar pacotes para f através do AS G, isto não estará em desacordo com as políticas de utilização, logo G irá fazer o anúncio desta rota.

Exemplo 2: o AS D tem a opção de anunciar sua rota até a subrede c para o AS C. A rota que D conhece para enviar pacotes para a subrede c passa através de um *peering link*, e o enlace que liga D a C é parte do backbone da rede. Se C enviar pacotes para c através do AS D, ele irá utilizar o *peering link* indevidamente, de acordo com as políticas de utilização, logo D não irá fazer o anúncio desta rota.

- i. AS G **anuncia** rota até f para AS A
- ii. AS G **anuncia** rota até h para AS A
- iii. AS D **não anuncia** rota até c para AS C
- iv. AS G **não anuncia** rota até c para AS F
- v. AS B **anuncia** rota até c para AS G
- vi. AS G **anuncia** rota até h para AS F
- vii. AS C **anuncia** rota até a para AS D
- viii. AS F **anuncia** rota até d para AS G

Questão 7 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 93.248.179.173 e que gerencia as conexões da rede privada, que ocupa a faixa 10.0.0.0/8. 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)	10.0.0.1, 28418	214.56.127.43, 3414	17655
(2)	10.0.0.1, 3099	146.120.38.247, 27245	1029
(3)	10.0.0.3, 2708	230.132.30.76, 15972	21138
(4)	10.0.0.3, 17735	161.23.35.16, 24917	24315
(5)	10.0.0.2, 12651	65.254.55.245, 11420	1026
(6)	10.0.0.4, 20211	251.76.63.142, 19162	31524
(7)	10.0.0.2, 6921	238.130.56.4, 25965	25344
(8)	10.0.0.3, 17157	191.213.39.232, 29643	1025
(9)	10.0.0.5, 8992	234.121.195.92, 3295	28584
(10)	10.0.0.1, 1828	138.170.123.187, 20642	11165

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.

Resposta:

Para que o NAT seja capaz de encaminhar um pacote vindo da rede pública para a estação correta, é preciso que a tabela de tradução indique quem são a estação e a porta corretas para este pacote. Isto significa que deve existir uma entrada na tabela correspondente à origem do pacote, e que ele deve ser recebido pelo NAT em uma porta aberta que tenha sido reservada para pacotes com essa origem. Se esta entrada não existir, o NAT não tem como saber para quem encaminhar o pacote, e irá descartá-lo.

Se a entrada existir, tudo que o NAT precisa fazer é realizar a tradução de endereço: o pacote, que ainda tinha como endereço destino “o NAT, na porta x”, passará a ter como destino “uma das estações, na porta y”. Qual estação e qual porta são obtidos consultando a entrada selecionada da tabela de tradução.

Exemplo 1: pacote com origem: 238.130.56.4, porta 25965; e destino: 93.248.179.173, porta 25344; note que a entrada (7) da tabela indica que uma estação estabeleceu comunicação com o endereço 238.130.56.4 na porta 25965, e que esta comunicação utilizou a porta 25344 no NAT. Logo, os pacotes que fazem o caminho contrário têm como destino verdadeiro a estação indicada nesta entrada da tabela (10.0.0.2), na porta indicada na tabela (6921).

Após a tradução, o cabeçalho do pacote passará a ter como origem: 238.130.56.4, porta 25965; e destino: 10.0.0.2, porta 6921.

Exemplo 2: pacote com origem: 214.56.127.43, porta 3414; e destino: 93.248.179.173, porta 24315; note que nenhuma entrada da tabela indica uma comunicação com este endereço e porta na rede pública, que esteja associada à porta 24315 do NAT, portanto, o NAT deve descartar este pacote.

- i. Origem: 238.130.56.4, 25965; Destino: 93.248.179.173, 25344
→ **Origem: 238.130.56.4, 25965; Destino: 10.0.0.2, 6921**
 - ii. Origem: 65.254.55.245, 11420; Destino: 93.248.179.173, 1026
→ **Origem: 65.254.55.245, 11420; Destino: 10.0.0.2, 12651**
 - iii. Origem: 234.121.195.92, 3295; Destino: 93.248.179.173, 28584
→ **Origem: 234.121.195.92, 3295; Destino: 10.0.0.5, 8992**
 - iv. Origem: 214.56.127.43, 3414; Destino: 93.248.179.173, 24315
descartado
 - v. Origem: 191.213.39.232, 29643; Destino: 93.248.179.173, 11165
descartado
 - vi. Origem: 251.76.63.142, 19162; Destino: 93.248.179.173, 17655
descartado
 - vii. Origem: 251.76.63.142, 19162; Destino: 93.248.179.173, 31524
→ **Origem: 251.76.63.142, 19162; Destino: 10.0.0.4, 20211**
 - viii. Origem: 238.130.56.4, 25965; Destino: 93.248.179.173, 1029
descartado
 - ix. Origem: 161.23.35.16, 24917; Destino: 93.248.179.173, 24315
→ **Origem: 161.23.35.16, 24917; Destino: 10.0.0.3, 17735**
 - x. Origem: 214.56.127.43, 3414; Destino: 93.248.179.173, 1026
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

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

Resposta:

Novamente, teremos que exercer o papel de NAT nesta questão. Para cada pacote, se já houver uma entrada correspondente a essa comunicação (isto é, a estes pares de endereço e porta, origem e destino) na tabela de tradução, isto significa que o NAT já reservou uma porta pública para esta comunicação. Se não existir uma entrada na tabela de tradução, então o NAT deverá criar uma, e deve reservar uma porta para esta comunicação. Esta porta pode ser qualquer porta livre (isto é, que não esteja reservada para outra comunicação nesta tabela) e, claro, seja maior do que 1024 como pede o enunciado.

Note que nenhum pacote pode ser bloqueado nesta direção pois, caso contrário, nenhuma estação na rede privada conseguiria abrir comunicação com nenhuma estação na rede pública.

Exemplo 1: pacote com origem: 10.0.0.5, porta 8992; e destino: 234.121.195.92, porta 3295. Note que a entrada (9) corresponde a estes endereços e portas, logo o pacote deverá ser encaminhado à rede pública e ser traduzido segundo esta entrada.

Exemplo 2: pacote com origem: 10.0.0.1, porta 28418; e destino: 214.56.127.43, porta 16833. Não há nenhuma entrada correspondente a estes endereços e portas, logo o NAT deverá criar uma nova entrada (11) para este fim na tabela de tradução, e reservar uma porta para esta comunicação (por exemplo, porta 1024), registrando esta porta na nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(11)	10.0.0.1, 28418	214.56.127.43, 16833	1024

- i. Origem: 10.0.0.5, 8992; Destino: 234.121.195.92, 3295

Encaminhado segundo a entrada (9)

- ii. Origem: 10.0.0.1, 28418; Destino: 214.56.127.43, 3414

Encaminhado segundo a entrada (1)

- iii. Origem: 10.0.0.1, 1828; Destino: 138.170.123.187, 20642

Encaminhado segundo a entrada (10)

- iv. Origem: 10.0.0.1, 28418; Destino: 214.56.127.43, 16833

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(11)	10.0.0.1, 28418	214.56.127.43, 16833	1024

- v. Origem: 10.0.0.2, 12651; Destino: 70.130.131.164, 19715

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(12)	10.0.0.2, 12651	70.130.131.164, 19715	1027

- vi. Origem: 10.0.0.1, 1828; Destino: 138.170.123.187, 20642

Encaminhado segundo a entrada (10)

- vii. Origem: 10.0.0.3, 27032; Destino: 161.23.35.16, 24917

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(13)	10.0.0.3, 27032	161.23.35.16, 24917	1028

- viii. Origem: 10.0.0.3, 27032; Destino: 161.23.35.16, 32649

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(14)	10.0.0.3, 27032	161.23.35.16, 32649	1030

- ix. Origem: 10.0.0.2, 12651; Destino: 70.130.131.164, 9391

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(15)	10.0.0.2, 12651	70.130.131.164, 9391	1031

- x. Origem: 10.0.0.2, 6921; Destino: 238.130.56.4, 25965

Encaminhado segundo a entrada (7)

- (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.

Resposta:

Após o item anterior, para todos os pacotes o NAT já sabe como realizar a tradução de endereço, pois todos os pacotes já possuem uma entrada associada na tabela de tradução. Tudo o que ele precisa fazer agora é “se passar” pela estação origem, substituindo o endereço e porta de origem do pacote pelo seu próprio endereço e pela porta que está reservada na entrada apropriada.

Nos exemplos anteriores: o pacote do exemplo 1 passará a ter origem: 93.248.179.173, porta 28584; e o pacote do exemplo 2 passará a ter origem: 93.248.179.173, porta 1024. Note que os destinos, sendo estações na rede pública, permanecem inalterados.

- i. Origem: 10.0.0.5, 8992; Destino: 234.121.195.92, 3295
 → **Origem: 93.248.179.173, 28584; Destino: 234.121.195.92, 3295**
- ii. Origem: 10.0.0.1, 28418; Destino: 214.56.127.43, 3414
 → **Origem: 93.248.179.173, 17655; Destino: 214.56.127.43, 3414**
- iii. Origem: 10.0.0.1, 1828; Destino: 138.170.123.187, 20642
 → **Origem: 93.248.179.173, 11165; Destino: 138.170.123.187, 20642**
- iv. Origem: 10.0.0.1, 28418; Destino: 214.56.127.43, 16833
 → **Origem: 93.248.179.173, 1024; Destino: 214.56.127.43, 16833**
- v. Origem: 10.0.0.2, 12651; Destino: 70.130.131.164, 19715
 → **Origem: 93.248.179.173, 1027; Destino: 70.130.131.164, 19715**
- vi. Origem: 10.0.0.1, 1828; Destino: 138.170.123.187, 20642
 → **Origem: 93.248.179.173, 11165; Destino: 138.170.123.187, 20642**
- vii. Origem: 10.0.0.3, 27032; Destino: 161.23.35.16, 24917
 → **Origem: 93.248.179.173, 1028; Destino: 161.23.35.16, 24917**
- viii. Origem: 10.0.0.3, 27032; Destino: 161.23.35.16, 32649
 → **Origem: 93.248.179.173, 1030; Destino: 161.23.35.16, 32649**
- ix. Origem: 10.0.0.2, 12651; Destino: 70.130.131.164, 9391
 → **Origem: 93.248.179.173, 1031; Destino: 70.130.131.164, 9391**
- x. Origem: 10.0.0.2, 6921; Destino: 238.130.56.4, 25965
 → **Origem: 93.248.179.173, 25344; Destino: 238.130.56.4, 25965**

Questão 8 10 pontos

A técnica de bit de paridade é frequentemente usada para detectar erros em transmissões

de sequências de bits. O objetivo desta questão é entender como calcular o bit de paridade e como a técnica é empregada para detecção de erros.

- (a) Considere o mecanismo de paridade par. Calcule o bit de paridade para cada um dos pacotes de 10 bits abaixo.

Resposta:

O bit de paridade de um pacote é igual ao XOR de todos os bits neste pacote, e indica se a quantidade total de 1's é par ou ímpar (paridade 0 para quantidade par de 1's, e paridade 1 para quantidade ímpar de 1's).

- | | |
|--------------------------------------|---------------------------------------|
| i. 0011100000 — paridade 1 | xi. 1010110010 — paridade 1 |
| ii. 0110001100 — paridade 0 | xii. 0100010100 — paridade 1 |
| iii. 0001101100 — paridade 0 | xiii. 1111000101 — paridade 0 |
| iv. 1001100110 — paridade 1 | xiv. 0001111101 — paridade 0 |
| v. 0001100100 — paridade 1 | xv. 1110011010 — paridade 0 |
| vi. 1110001110 — paridade 0 | xvi. 1011101110 — paridade 1 |
| vii. 1010000010 — paridade 1 | xvii. 1000011101 — paridade 1 |
| viii. 1011101111 — paridade 0 | xviii. 1000000110 — paridade 1 |
| ix. 0100110001 — paridade 0 | xix. 0100111100 — paridade 1 |
| x. 1100001011 — paridade 1 | xx. 0000000010 — paridade 1 |

- (b) Suponha que cada pacote acima foi transmitido juntamente com seu bit de paridade por um canal de comunicação ruidoso. Este canal pode introduzir erros invertendo os bits que são transmitidos, como um canal de rádio. Considere que a estação do outro lado do canal recebeu os bits indicados nos itens abaixo (onde o último bit da sequência é o bit de paridade recebido). Repare que o bit de paridade também está sujeito aos erros introduzidos pelo canal!

Para cada pacote, determine a paridade da sequência recebida, e determine também se o pacote será aceito ou rejeitado em função das paridades⁴. Baseado neste resultado, decida, para cada pacote, se a técnica de detecção de erro funcionou adequadamente, ou seja, se os erros de bit apresentados pelo pacote foram detectados.

⁴Lembre-se que o receptor dos dados não possui acesso ao pacote original, logo ele não pode comparar o pacote recebido com o original para decidir se aceita ou não.

Resposta:

Como o receptor não tem acesso ao pacote original, a única forma de verificar a integridade dos bits enviados é verificando a paridade do pacote. Isto pode ser feito calculando a paridade do pacote sem o último bit e verificar se o resultado bate com o bit de paridade. Outra forma de fazer essa checagem é perceber que o pacote enviado (incluindo o bit de paridade) *sempre* terá um número par de 1's. Basta, então, checar se o XOR de todos os bits é igual a 0.

Se a paridade estiver incorreta, com certeza pelo menos um bit sofreu erro de transmissão; o receptor deve então rejeitar o pacote. Se a paridade estiver correta, ainda assim podem ter ocorrido erros (por exemplo, se dois bits forem invertidos, a paridade do pacote não muda), mas o receptor não tem como detectar os erros e aceita o pacote.

Agora, enquanto o receptor do pacote não tem como saber sempre se a detecção de erros funcionou adequadamente, nós temos como saber, pois sabemos qual era o pacote original. Basta comparar o conteúdo dos dois pacotes (incluindo o bit de paridade do pacote original) e verificar se eles são iguais ou não. Vamos utilizar o primeiro item como exemplo.

O pacote recebido tem a sequência de bits 00111010011. O receptor, sem olhar para o pacote original verifica que, excluindo o último bit, o pacote possui paridade 1, que é igual ao valor do último bit. Portanto, a paridade do pacote recebido está correta, e o receptor aceita o pacote.

No entanto, comparando com o pacote original (que é igual a 00111000001, se incluirmos o bit de paridade), notamos que os bits 7 e 10 foram invertidos. Logo, temos um erro que não foi detectado, ou seja, a detecção de erros não funcionou corretamente.

- | | |
|--|---|
| i. Pacote: 0011100000
Recebido: 00111010011
Paridade: 0 — Pacote aceito
Resultado: Erro não-detectado | vi. Pacote: 1110001110
Recebido: 11100001100
Paridade: 1 — Pacote rejeitado
Resultado: Erro detectado |
| ii. Pacote: 0110001100
Recebido: 01100011001
Paridade: 1 — Pacote rejeitado
Resultado: Erro detectado | vii. Pacote: 1010000010
Recebido: 10100000100
Paridade: 1 — Pacote rejeitado
Resultado: Erro detectado |
| iii. Pacote: 0001101100
Recebido: 01011011010
Paridade: 0 — Pacote aceito
Resultado: Erro não-detectado | viii. Pacote: 1011101111
Recebido: 10111110110
Paridade: 0 — Pacote aceito
Resultado: Erro não-detectado |
| iv. Pacote: 1001100110
Recebido: 10001000101
Paridade: 0 — Pacote aceito
Resultado: Erro não-detectado | ix. Pacote: 0100110001
Recebido: 01001100010
Paridade: 0 — Pacote aceito
Resultado: Sem erro |
| v. Pacote: 0001100100
Recebido: 00010001001
Paridade: 1 — Pacote rejeitado
Resultado: Erro detectado | x. Pacote: 1100001011
Recebido: 11000010111
Paridade: 0 — Pacote aceito
Resultado: Sem erro |

- | | |
|---|---|
| xi. Pacote: 1010110010
Recebido: 10101100101
Paridade: 0 — Pacote aceito
Resultado: Sem erro | xvi. Pacote: 1011101110
Recebido: 10111011001
Paridade: 1 — Pacote rejeitado
Resultado: Erro detectado |
| xii. Pacote: 0100010100
Recebido: 01000101001
Paridade: 0 — Pacote aceito
Resultado: Sem erro | xvii. Pacote: 1000011101
Recebido: 10001111111
Paridade: 0 — Pacote aceito
Resultado: Erro não-detectado |
| xiii. Pacote: 1111000101
Recebido: 11110001010
Paridade: 0 — Pacote aceito
Resultado: Sem erro | xviii. Pacote: 1000000110
Recebido: 10000000101
Paridade: 1 — Pacote rejeitado
Resultado: Erro detectado |
| xiv. Pacote: 0001111101
Recebido: 00011101010
Paridade: 1 — Pacote rejeitado
Resultado: Erro detectado | xix. Pacote: 0100111100
Recebido: 01001111001
Paridade: 0 — Pacote aceito
Resultado: Sem erro |
| xv. Pacote: 1110011010
Recebido: 11000110100
Paridade: 1 — Pacote rejeitado
Resultado: Erro detectado | xx. Pacote: 0000000010
Recebido: 00000000101
Paridade: 0 — Pacote aceito
Resultado: Sem erro |

QUESTÕES EXTRAS

Questão 9 5 pontos

O protocolo TDMA é um protocolo de acesso a um meio de transmissão compartilhado. Ele funciona alocando fatias de tempo (*slots*) às estações de forma sucessiva. Somente a estação a quem o slot pertencer pode transmitir em um dado momento e, se ela não possuir dados, o meio fica ocioso. Nesta questão você deve compreender como funciona este mecanismo.

Considere um meio compartilhado por 5 estações por TDMA, em slots de 70 ms. Suponha que as estações transmitam dados a uma taxa de 40 Mbps, em quadros de 35000 bytes.

- (a) Qual é a quantidade máxima de dados que uma estação pode transmitir em um slot?

Resposta:

Como um slot dura 70 ms e a estação pode transmitir a uma taxa de 40 Mbps (Mbits por segundo), basta multiplicar os dois valores (tomando cuidado com as conversões de unidades). Assim, em um único slot, cada estação pode transmitir até 2.8 Mbits, ou 350000 bytes de dados.

- (b) Qual é o maior número de quadros que uma estação pode transmitir em um slot?

Resposta:

Se um quadro tem 35000 bytes, então em um slot cada estação transmite no máximo $350000/35000 = 10$ quadros por slot.

- (c) Considere agora que as estações estejam prontas para transmitir seus dados conforme a tabela a seguir. Repare que cada estação irá transmitir uma quantidade diferente de dados, e que estes dados estarão disponíveis em instantes de tempo diferentes.

	Dados	Pronta para transmitir em
Estação 1	5.04 Mbits	721.0 ms
Estação 2	1.12 Mbits	455.0 ms
Estação 3	6.44 Mbits	62.0 ms
Estação 4	7.84 Mbits	6.0 ms
Estação 5	0.84 Mbits	359.0 ms

Suponha que o primeiro slot pertence à estação 1 e segue em ordem crescente. Para cada estação, determine:

- Quantos quadros são necessários para transmitir seus dados;
- O instante de tempo em que seu primeiro quadro começa a ser transmitido;
- O instante de tempo em que a estação termina de transmitir seu último quadro;
- O retardo inicial da transmissão;⁵
- A vazão média (*throughput*) obtida pela estação.⁶

Resposta:

Vamos exemplificar a resolução deste exercício com a estação 1. Note que as estações transmitem de forma independente, cada uma nos slots que lhe pertencem. A estação 1 possui 5.04 Mbits, ou 630000 bytes, para transmitir no total. Logo, ela precisará enviar $630000/35000 = 18$ quadros. Isto resolve o item (i).

O primeiro slot da estação 1 começa $t = 0$ ms e termina em $t = 70$ ms. Os próximos 4 slots pertencerão às outras estações, em um ciclo que dura $5 \cdot 70.0 = 350.0$ ms. Depois disso, o processo se repete, logo ela irá obter o slot a cada 350.0 ms (nos instantes $t = 350.0$ ms, $t = 700.0$ ms, e assim por diante). Como a estação 1 recebe os dados em $t = 721.0$ ms, então ela possui o slot no momento em que recebe os dados e pode começar a transmitir imediatamente. Portanto, a transmissão começa em $t = 721.0$ ms. Isto significa que o retardo inicial é igual a $721.0 - 721.0 = 0.0$ ms. Resolvemos, com isto, os itens (ii) e (iv). Para resolver o item (iii), vamos detalhar a transmissão da estação 1. Ela inicia em $t = 721.0$ ms e é interrompida ao final do slot, em $t = 770.0$ ms. Neste período de 49.0 ms, foi possível transmitir 7 quadros (cada quadro leva 7.0 ms, já que em um slot de 70 ms podem ser transmitidos 10 quadros). Restam 11 a serem transmitidos.

Estes 11 quadros precisarão de 2 slots para serem transmitidos. No primeiro slot seguinte (que começa em $t = 1050.0$ ms), serão transmitidos 10 quadros. O quadro restante será transmitido no próximo slot, que começa em $t = 1400.0$ ms. A transmissão deste quadro irá durar 7.0 ms, e se encerra em $t = 1407.0$ ms. Isto completa o item (iii).

Falta calcularmos o item (v). Para isto, note que, entre a chegada dos dados e o final da transmissão, o tempo decorrido foi igual a $1407.0 - 721.0 = 686.0$ ms. Foram transmitidos, no total, 5.04 Mbits de dados pela estação 1, logo a vazão média foi de $5.04 \text{ Mbits} / 0.686 \text{ s} \approx 7.3 \text{ Mbps}$.

⁵Tempo decorrido entre a estação ter dados para transmitir e a transmissão efetivamente iniciar.

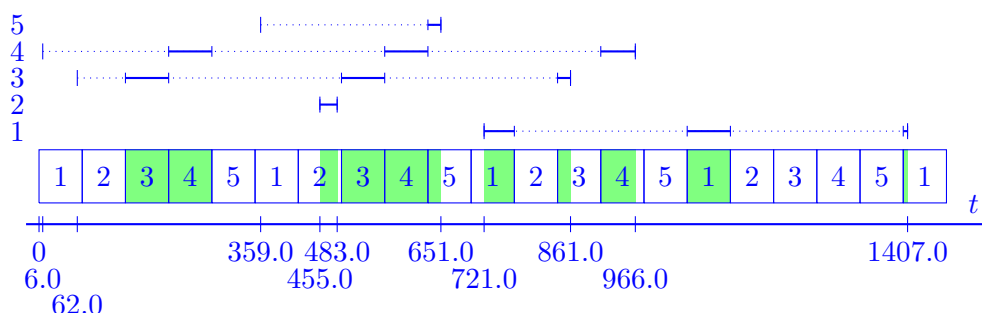
⁶Razão entre a quantidade de dados transmitidos e o tempo necessário para transmitir estes dados, medido entre a disponibilidade dos dados e o final da transmissão.

Resposta:

Realizando os cálculos para as demais estações, temos:

	(i) Quadros	(ii) — (iii) Período de transmissão	(iv) Retardo inicial	(v) Vazão média
Estação 1	18 quadros	721.0 ms — 1407.0 ms	0.0 ms	7.3 Mbps
Estação 2	4 quadros	455.0 ms — 483.0 ms	0.0 ms	40.0 Mbps
Estação 3	23 quadros	140.0 ms — 861.0 ms	78.0 ms	8.1 Mbps
Estação 4	28 quadros	210.0 ms — 966.0 ms	204.0 ms	8.2 Mbps
Estação 5	3 quadros	630.0 ms — 651.0 ms	271.0 ms	2.9 Mbps

A figura a seguir ilustra o cenário de transmissões observado:



- (d) No cenário anterior, determine a vazão (*throughput*) média e a utilização média (fração de tempo em uso) do meio de transmissão. Para ambos, considere o tempo desde a primeira disponibilidade dos dados (entre todas as estações) até o final de todas as transmissões.

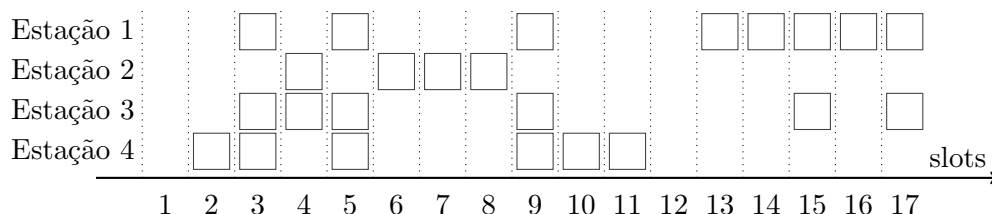
Resposta:

Durante o período de tempo considerado ($1407.0 - 6.0 = 1401.0\text{ms}$), a quantidade total de dados transmitidos por todas as estações foi de 21.28 Mbits. Portanto foi obtida uma vazão média de $21.28 \text{ Mbits} / 1.407 \text{ s} = 15.19 \text{ Mbps}$, que corresponde a uma utilização do meio de $15.19 / 40.0 = 38.0\%$ neste período.

Questão 10..... 5 pontos

Em contraste com o protocolo TDMA, o protocolo Slotted ALOHA (S-ALOHA) é um protocolo de acesso a meio de transmissão compartilhado com uma abordagem distribuída, porém sincronizada e que ainda permite a ocorrência de colisões. Nesta questão você deve compreender como funciona este protocolo.

Considere o seguinte perfil de transmissões realizadas por estações executando o protocolo Slotted ALOHA.



- (a) Para cada slot de tempo mostrado na figura acima, determine se ocorreu uma transmissão com sucesso (S), uma colisão (C), ou se o slot permaneceu ocioso (O).

Resposta:

Só é possível ocorrer uma transmissão com sucesso se, em um dado slot, exatamente uma estação transmitir um quadro. Se mais de uma estação transmitir, ocorrerá uma colisão, e se nenhuma estação transmitir, o quadro ficará ocioso.

slot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
estado	O	S	C	C	C	S	S	S	C	S	S	O	S	S	C	S	C

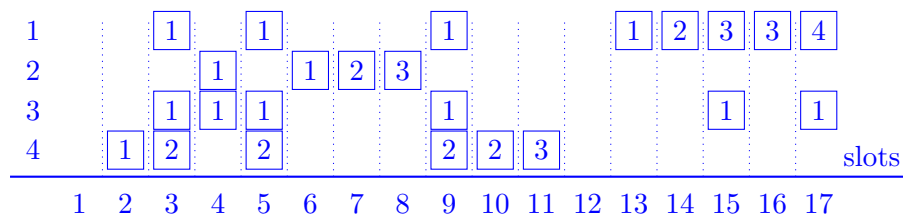
- (b) Considere que cada estação deseja transmitir o seguinte número de quadros a partir do instante de tempo zero:

Estação	Quadros para transmitir
Estação 1	4 quadros
Estação 2	3 quadros
Estação 3	4 quadros
Estação 4	3 quadros

Utilizando o perfil de transmissões ilustrado na figura acima, determine qual quadro está sendo transmitido por cada estação em cada uma das transmissões realizadas. Identifique cada quadro com o seu número de ordem na sequência transmitida por aquela estação.

Resposta:

Todas as estações começam com o quadro de número 1. Sempre que tentarem transmitir e houver uma colisão, eles tentarão o mesmo quadro novamente. Quando a transmissão tiver sucesso, elas passarão para o quadro de número 2, e continuam assim até acabarem os quadros.



- (c) Determine o instante de tempo em que cada uma das estações consegue realizar sua primeira transmissão de quadro com sucesso.

Resposta:

Observando o diagrama do item anterior:

Estação	Primeira transmissão
Estação 1	Slot 13
Estação 2	Slot 6
Estação 3	Nenhum sucesso
Estação 4	Slot 2

- (d) Determine quantos quadros restam a ser transmitidos por cada uma das estações ao final do tempo mostrado acima.

Resposta:

Novamente observando o diagrama:

Estação	Quadros restantes
Estação 1	1 quadros
Estação 2	0 quadros
Estação 3	4 quadros
Estação 4	0 quadros

- (e) Determine a utilização e a eficiência de uso do canal — lembrando que a utilização é dada pela fração de tempo que o canal foi utilizado, e a eficiência é dada pela fração de tempo que o canal foi utilizado com sucesso.

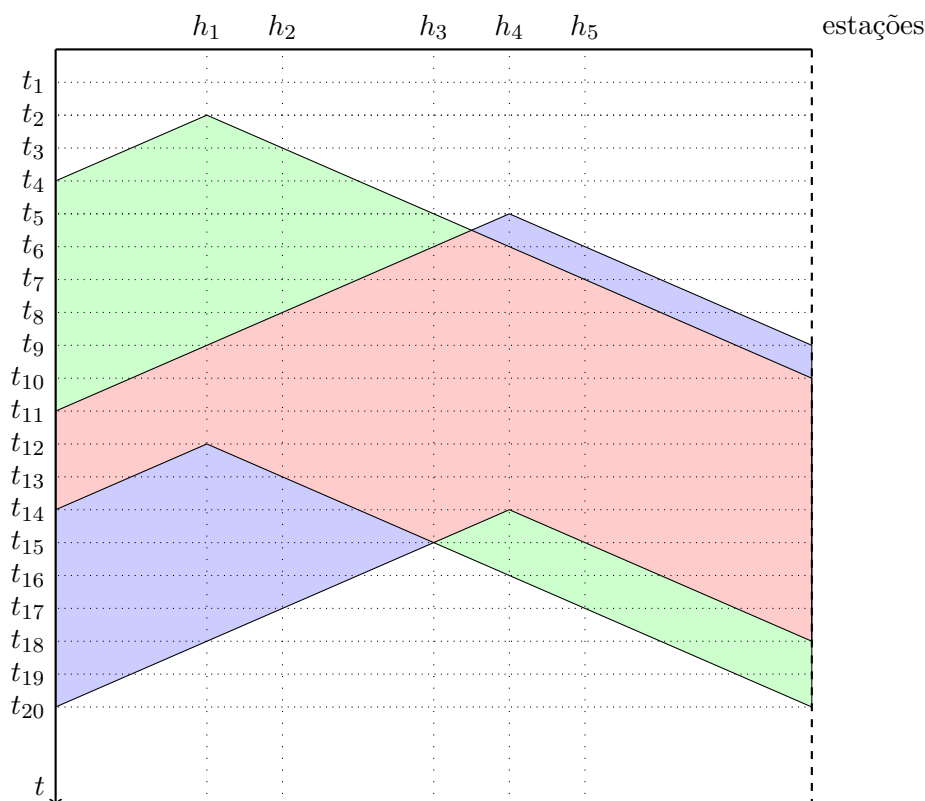
Resposta:

O canal ficou ocioso em 2 slots dos 17, logo a utilização foi de $15/17 = 88.2\%$. Mas só houve sucesso em 9 slots, levando a uma eficiência bem inferior: $9/17 = 52.9\%$.

Questão 11 5 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:

Analisando o diagrama, percebemos que duas estações transmitiram: a estação h_1 iniciou sua transmissão primeiro, utilizando o meio entre os instantes de tempo t_2 e t_{12} . Depois disto, a estação h_4 transmite dados entre t_5 e t_{14} .

- (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:

Novamente analisando o diagrama:

Estação	Transmissão de h_1	Transmissão de h_4
h_2	$t_3 - t_{13}$	$t_8 - t_{17}$
h_3	$t_5 - t_{15}$	$t_6 - t_{15}$
h_5	$t_7 - t_{17}$	$t_6 - t_{15}$

- (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:

Cada estação vê o meio ocupado durante o período em que ela percebe ao menos uma das duas transmissões. Por outro lado, a colisão é detectada no primeiro momento em que a estação percebe, ao mesmo tempo, as duas transmissões. Temos, então:

Estação	Vê meio ocupado entre:	Detecta colisão em:
h_1	$t_2 - t_{18}$	t_9
h_2	$t_3 - t_{17}$	t_8
h_3	$t_5 - t_{15}$	t_6
h_4	$t_5 - t_{16}$	t_6
h_5	$t_6 - t_{17}$	t_7

- (e) Considere os instantes de tempo t_3 , t_6 e t_{16} . Quais estações podem iniciar uma transmissão nestes instantes de tempo?

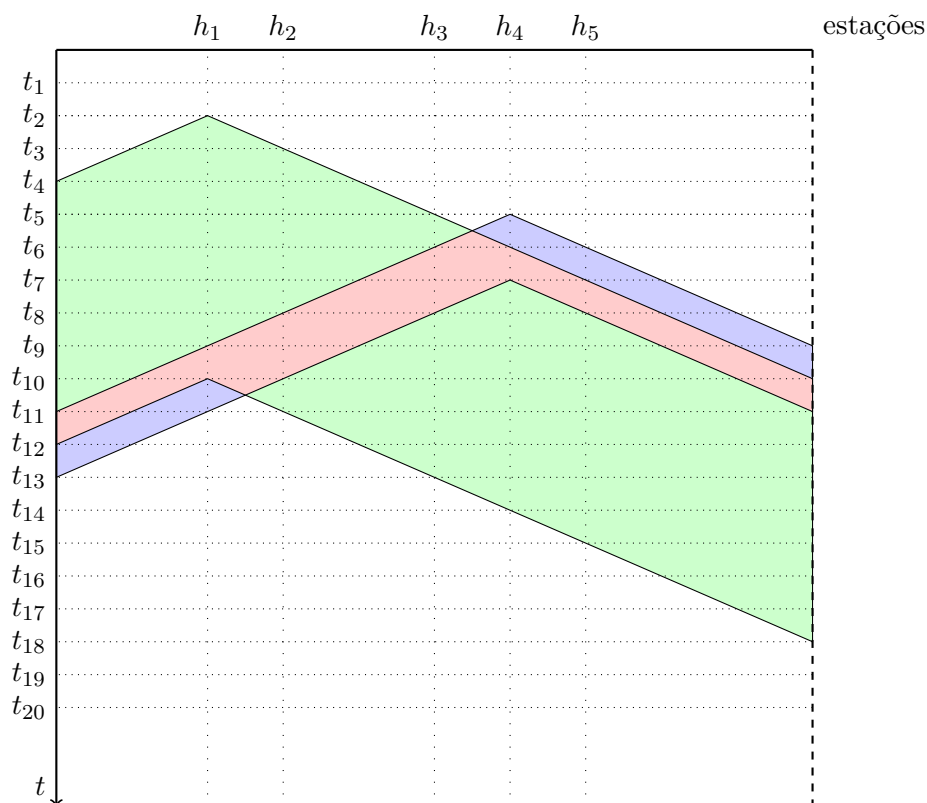
Resposta:

Nestes instantes de tempo, somente as estações que não percebem nenhuma transmissão podem transmitir, pois elas julgam que o meio está livre nestes instantes.

	Estações:
t_3	h_3, h_4, h_5
t_6	—
t_{16}	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_4 transmitiram. Desta vez, a estação h_1 transmitiu entre os instantes de tempo t_2 e t_{10} , e a estação h_4 transmitiu entre os instantes de tempo t_5 e t_7 . As demais estações recebem estas transmissões conforme a tabela a seguir:

Estação	Transmissão de h_1	Transmissão de h_4
h_2	$t_3 - t_{11}$	$t_8 - t_{10}$
h_3	$t_5 - t_{13}$	$t_6 - t_8$
h_5	$t_7 - t_{15}$	$t_6 - t_8$

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	$t_2 - t_{11}$	t_9
h_2	$t_3 - t_{11}$	t_8
h_3	$t_5 - t_{13}$	t_6
h_4	$t_5 - t_{14}$	t_6
h_5	$t_6 - t_{15}$	t_7

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

	Estações:
t_3	h_3, h_4, h_5
t_6	—
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	$t_{11} - t_{18}$
h_2	$t_{11} - t_{17}$
h_3	$t_{13} - t_{15}$
h_4	$t_{14} - t_{16}$
h_5	$t_{15} - t_{17}$