

**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina: Redes de Computadores II**  
**AD1 – 2º semestre de 2018 – GABARITO**

**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
00111000 10001100 00000000 00000000 a	0
00111000 10001100 11111111 11111111 a	1
00111000 10001100 01100000 00000000 a	2
00111000 10001100 01101000 00000000 a	3
00111000 10001100 01101111 11111111 a	2
00111000 10001100 00000000 00000000 a	2
00111000 10001100 11110000 00000000 a	2
00111000 10001100 11111111 11111111 a	2
00111000 10001100 01101100 10000000 a	2
00111000 10001100 01101100 11111111 a	2
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
00111000 10001100	0
00111000 10001100 011	1
00111000 10001100 01101	2
00111000 10001100 0000	3
00111000 10001100 1111	2
00111000 10001100 01101100 1	2
-	1

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

**Resposta:**

Prefixo	Interface
56.140.0.0/16	0
56.140.96.0/19	1
56.140.104.0/21	2
56.140.0.0/20	3
56.140.240.0/20	2
56.140.108.128/25	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. 56.140.108.153 → <b>interface 2</b>   | vii. 56.140.105.179 → <b>interface 2</b>  |
| ii. 56.140.109.152 → <b>interface 2</b>  | viii. 56.140.111.157 → <b>interface 2</b> |
| iii. 56.141.204.182 → <b>interface 1</b> | ix. 56.140.242.154 → <b>interface 2</b>   |
| iv. 56.140.97.29 → <b>interface 1</b>    | x. 56.140.125.244 → <b>interface 1</b>    |
| v. 56.141.179.104 → <b>interface 1</b>   | xi. 56.140.98.171 → <b>interface 1</b>    |
| vi. 56.140.184.28 → <b>interface 0</b>   | xii. 56.140.152.173 → <b>interface 0</b>  |

- (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.<sup>1</sup>

**Resposta:**

A regra do prefixo 56.140.108.128/25 é redundante.

**Questão 2** ..... 20 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.<sup>2</sup>

<sup>1</sup>Dica: procure por faixas mais específicas.

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

i.  $R_1$ : 157.34.0.0/15  
com 5 subredes:  
 $R_{11}$ : 24000 estações  
**157.34.128.0/17**  
 $R_{12}$ : 30000 estações  
**157.34.0.0/17**  
 $R_{13}$ : 6000 estações  
**157.35.128.0/19**  
 $R_{14}$ : 14000 estações  
**157.35.64.0/18**  
 $R_{15}$ : 15000 estações  
**157.35.0.0/18**

ii.  $R_2$ : 163.152.0.0/14  
com 3 subredes:  
 $R_{21}$ : 10000 estações  
**163.152.128.0/18**  
 $R_{22}$ : 20000 estações  
**163.152.0.0/17**  
 $R_{23}$ : 10000 estações  
**163.152.192.0/18**

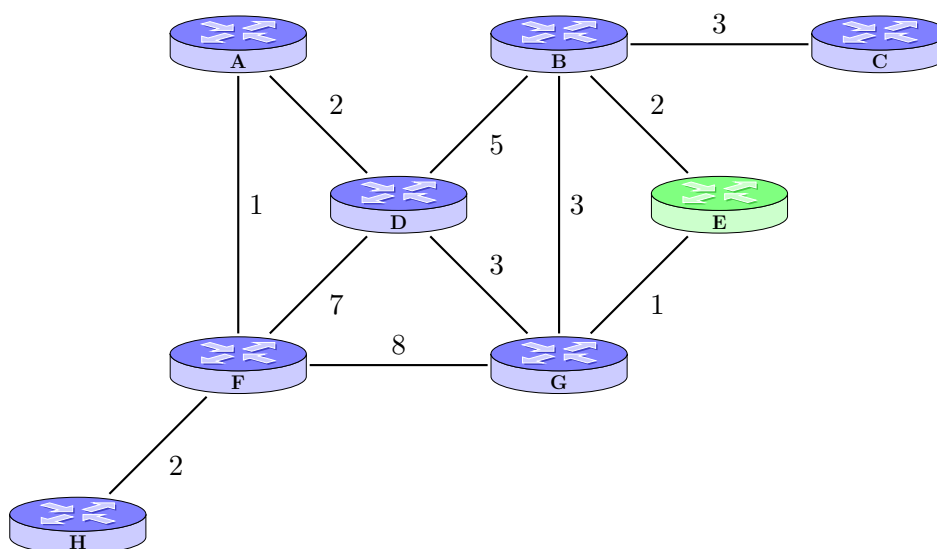
iii.  $R_3$ : 95.224.0.0/11  
com 5 subredes:  
 $R_{31}$ : 400000 estações  
 $R_{32}$ : 300000 estações  
 $R_{33}$ : 200000 estações  
 $R_{34}$ : 400000 estações  
 $R_{35}$ : 500000 estações

**Alocação impossível**

**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.

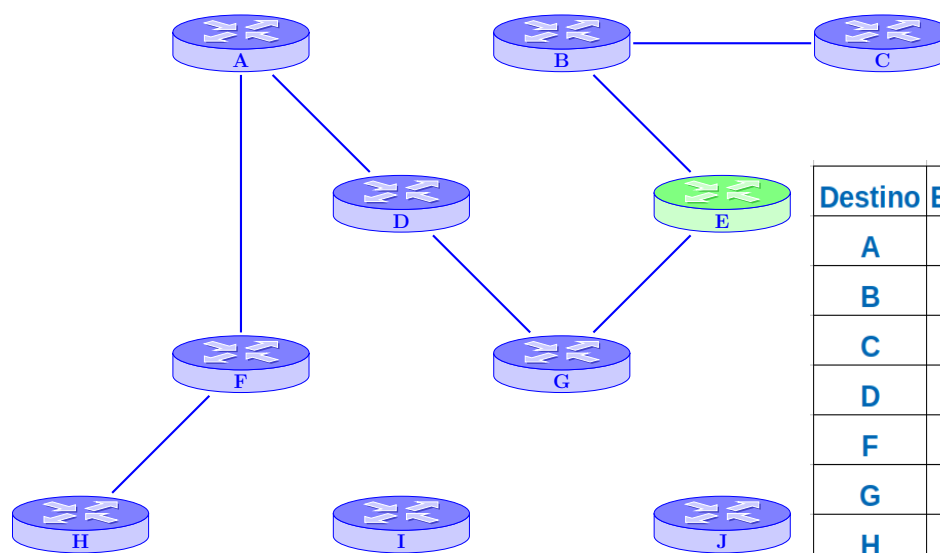


- (a) Utilizando o algoritmo de Dijkstra, calcule os caminhos mais curtos a partir do nó E, destacado em verde, para todos os outros nós da rede. Construa uma tabela igual à mostrada em aula que mostra o funcionamento do algoritmo de forma iterativa.

**Resposta:**

	N°	d <sub>A</sub> PA	d <sub>B</sub> PB	d <sub>C</sub> PC	d <sub>D</sub> PD	d <sub>F</sub> PF	d <sub>G</sub> PG	d <sub>H</sub> PH
0	E	∞ -	2 E	∞ -	∞ -	∞ -	1 E	∞ -
1	EG	∞ -	2 E	∞ -	4 G	9 G		∞ -
2	EGB	∞ -		5 B	4 G	9 G		∞ -
3	EGBD	6 D		5 B		9 G		∞ -
4	EGBDC	6 D				9 G		∞ -
5	EGBDCA					7 A		∞ -
6	EGBDCAF							9 F
7	EGBDCAFH							

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

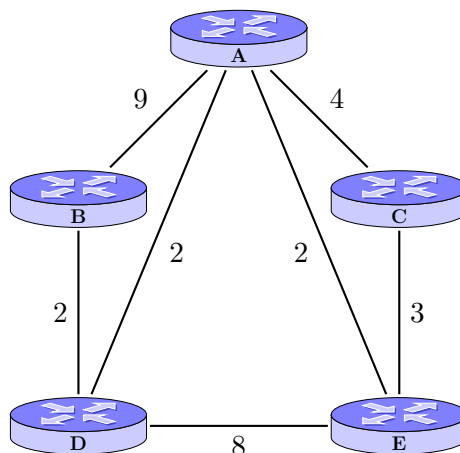
**Resposta:**

Destino	Enlace de Saída
A	(E, G)
B	(E, B)
C	(E, B)
D	(E, G)
F	(E, G)
G	(E, G)
H	(E, G)

**FALTA A TABELA DE ROTEAMENTO!!**

**Questão 4**..... 20 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	9	∞	∞	∞	via	A	9	∞	∞	∞	via	A	4	∞	∞	∞	via	A	2	∞	∞	∞	via	A	2	∞	∞	∞					
	C	∞	4	∞	∞		D	∞	∞	2	∞		E	∞	∞	∞	3		C	∞	2	∞	∞		C	∞	∞	3	∞	D	∞	∞	3	∞
	D	∞	∞	2	∞														D	∞	∞	∞	8		D	∞	∞	∞	8					
	E	∞	∞	∞	2														E	∞	∞	∞	8											

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	4	2	2	9	0	∞	2	∞	4	∞	0	∞	3	2	2	∞	0	8	2	∞	3	8	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	9	∞	11	∞	via	A	9	13	11	11	via	A	4	13	6	6	via	A	2	11	6	4	via	A	2	11	6	4
	C	∞	4	∞	7		D	4	∞	2	10		E	5	∞	11	3		B	11	2	∞	∞		C	7	∞	3	∞
	D	4	∞	2	10														E	10	∞	11	8		D	10	10	∞	8
	E	∞	5	10	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	4	4	2	2	4	0	13	2	10	4	13	0	6	3	2	2	6	0	4	2	10	3	4	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	9	22	11	19	via	A	9	13	11	11	via	A	4	8	6	6	via	A	2	6	6	4	via	A	2	6	6	4
	C	17	4	10	7	D	4	8	2	6	E	5	13	7	3	via	B	6	2	15	12	via	C	7	16	3	9		
	D	4	8	2	6											E	10	18	11	8		D	10	10	14	8			
	E	12	5	6	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	4	4	2	2	4	0	8	2	6	4	8	0	6	3	2	2	6	0	4	2	6	3	4	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				C	A B D E				D	A B C E				E	A B C D							
via	B	9	17	11	15	via	A	9	13	11	11	via	A	4	8	6	6	via	A	2	6	6	4	via	A	2	6	6	4
	C	12	4	10	7		D	4	8	2	6		E	5	9	7	3		B	6	2	10	8		C	7	11	3	9
	D	4	8	2	6														E	10	14	11	8		D	10	10	14	8
	E	8	5	6	2																								

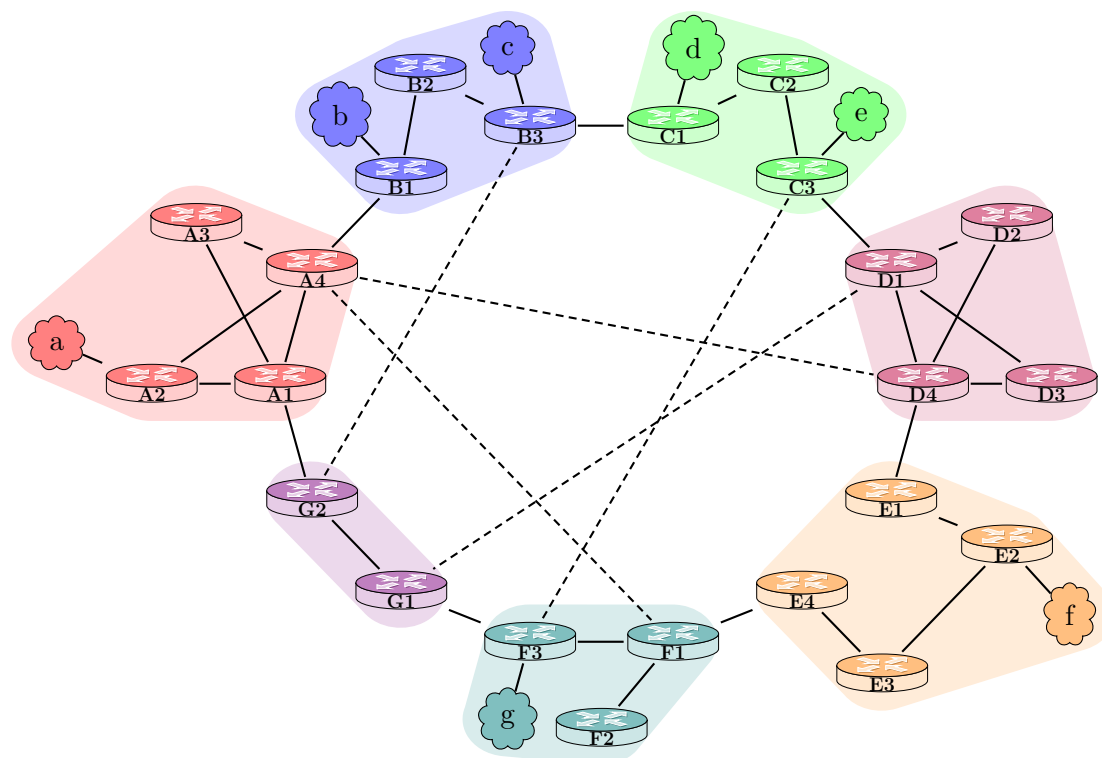
  

vetor de A					vetor de B					vetor de C					vetor de D					vetor de E				
A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
0	4	4	2	2	4	0	8	2	6	4	8	0	6	3	2	2	6	0	4	2	6	3	4	0

**Questão 5**..... 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).

- |                            |                             |                           |
|----------------------------|-----------------------------|---------------------------|
| i. B2 e C2 : —             | v. C2 e C3 : <b>iBGP</b>    | ix. A4 e F1 : <b>eBGP</b> |
| ii. E4 e F1 : <b>eBGP</b>  | vi. F2 e F1 : <b>iBGP</b>   | x. B3 e F2 : —            |
| iii. E3 e E2 : <b>iBGP</b> | vii. G1 e G2 : <b>iBGP</b>  | xi. C3 e B2 : —           |
| iv. G2 e B3 : <b>eBGP</b>  | viii. D3 e D2 : <b>iBGP</b> | xii. B3 e A4 : —          |

(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.

- i.  $g \rightarrow f$   
**ASs: F - E**  
**Caminho: F3 - F1 - E4 - E3 - E2**
- ii.  $e \rightarrow d$   
**ASs: C**  
**Caminho: C3 - C2 - C1**

- iii.  $d \rightarrow f$   
**ASs: C - D - E**  
**Caminho: C1 - C2 - C3 - D1 - D4 - E1 - E2**
  - iv.  $f \rightarrow e$   
**ASs: E - D - C**  
**Caminho: E2 - E1 - D4 - D1 - C3**
  - v.  $a \rightarrow g$   
**ASs: A - F**  
**Caminho: A2 - A4 - F1 - F3**
  - vi.  $f \rightarrow c$   
**ASs: E - D - C - B**  
**Caminho: E2 - E1 - D4 - D1 - C3 - C2 - C1 - B3**
  - vii.  $e \rightarrow b$   
**ASs: C - B**  
**Caminho: C3 - C2 - C1 - B3 - B2 - B1**
  - viii.  $d \rightarrow c$   
**ASs: C - B**  
**Caminho: C1 - B3**
  - ix.  $a \rightarrow c$   
**ASs: A - B**  
**Caminho: A2 - A4 - B1 - B2 - B3**
  - x.  $g \rightarrow d$   
**ASs: F - C**  
**Caminho: F3 - C3 - C2 - C1**
- (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.
- i. AS C **anuncia** rota até a para AS D
  - ii. AS E **anuncia** rota até f para AS D
  - iii. AS A **não anuncia** rota até f para AS D
  - iv. AS A **anuncia** rota até a para AS F
  - v. AS B **anuncia** rota até b para AS C
  - vi. AS F **anuncia** rota até g para AS A
  - vii. AS G **não anuncia** rota até b para AS A
  - viii. AS A **anuncia** rota até a para AS B

---

**Questão 6** ..... 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 47.59.17.30 e que gerencia as conexões da rede privada, que ocupa a faixa 172.16.0.0/12. 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)	172.16.0.1, 12503	118.202.29.204, 18181	32033
(2)	172.16.0.1, 24072	110.56.2.182, 26503	1028
(3)	172.16.0.1, 5203	226.202.247.183, 20441	1026
(4)	172.16.0.2, 3005	99.9.47.111, 23558	24729
(5)	172.16.0.3, 18673	234.66.47.43, 12313	1027
(6)	172.16.0.4, 12122	111.158.139.83, 22445	27525
(7)	172.16.0.4, 19105	249.35.9.93, 17123	26301
(8)	172.16.0.3, 31794	160.66.226.17, 20444	1025
(9)	172.16.0.2, 2209	201.185.119.134, 26725	18687
(10)	172.16.0.1, 14534	25.220.233.184, 31691	20834

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: 110.56.2.182, 26503; Destino: 47.59.17.30, 1028  
 → **Origem: 110.56.2.182, 26503; Destino: 172.16.0.1, 24072**
- ii. Origem: 160.66.226.17, 20444; Destino: 47.59.17.30, 1025  
 → **Origem: 160.66.226.17, 20444; Destino: 172.16.0.3, 31794**
- iii. Origem: 118.202.29.204, 18181; Destino: 47.59.17.30, 24729  
**descartado**
- iv. Origem: 25.220.233.184, 31691; Destino: 47.59.17.30, 20834  
 → **Origem: 25.220.233.184, 31691; Destino: 172.16.0.1, 14534**
- v. Origem: 21.144.81.196, 26474; Destino: 47.59.17.30, 1028  
**descartado**
- vi. Origem: 186.28.50.159, 24334; Destino: 47.59.17.30, 1028  
**descartado**
- vii. Origem: 167.107.32.255, 2819; Destino: 47.59.17.30, 27525  
**descartado**
- viii. Origem: 111.158.139.83, 22445; Destino: 47.59.17.30, 1026  
**descartado**
- ix. Origem: 61.158.169.160, 21725; Destino: 47.59.17.30, 1028  
**descartado**
- x. Origem: 25.220.233.184, 31691; Destino: 47.59.17.30, 1027  
**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.<sup>3</sup>

- i. Origem: 172.16.0.1, 5203; Destino: 226.202.247.183, 20441

**Encaminhado segundo a entrada (3)**

- ii. Origem: 172.16.0.1, 12503; Destino: 118.202.29.204, 18181

**Encaminhado segundo a entrada (1)**

- iii. Origem: 172.16.0.1, 5203; Destino: 226.202.247.183, 20441

**Encaminhado segundo a entrada (3)**

- iv. Origem: 172.16.0.4, 19105; Destino: 249.35.9.93, 17123

**Encaminhado segundo a entrada (7)**

- v. Origem: 172.16.0.3, 31794; Destino: 160.66.226.17, 20444

**Encaminhado segundo a entrada (8)**

- vi. Origem: 172.16.0.4, 19105; Destino: 249.35.9.93, 17123

**Encaminhado segundo a entrada (7)**

- vii. Origem: 172.16.0.1, 5203; Destino: 51.235.53.124, 10210

**Nova entrada:**

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(11)	172.16.0.1, 5203	51.235.53.124, 10210	1024

- viii. Origem: 172.16.0.1, 12503; Destino: 144.126.169.67, 8818

**Nova entrada:**

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(12)	172.16.0.1, 12503	144.126.169.67, 8818	1029

- ix. Origem: 172.16.0.4, 19105; Destino: 249.35.9.93, 23942

**Nova entrada:**

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(13)	172.16.0.4, 19105	249.35.9.93, 23942	1030

- x. Origem: 172.16.0.1, 5203; Destino: 226.202.247.183, 6754

**Nova entrada:**

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(14)	172.16.0.1, 5203	226.202.247.183, 6754	1031

- (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.

- i. Origem: 172.16.0.1, 5203; Destino: 226.202.247.183, 20441

→ **Origem: 47.59.17.30, 1026; Destino: 226.202.247.183, 20441**

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

- ii. Origem: 172.16.0.1, 12503; Destino: 118.202.29.204, 18181  
→ **Origem: 47.59.17.30, 32033; Destino: 118.202.29.204, 18181**
- iii. Origem: 172.16.0.1, 5203; Destino: 226.202.247.183, 20441  
→ **Origem: 47.59.17.30, 1026; Destino: 226.202.247.183, 20441**
- iv. Origem: 172.16.0.4, 19105; Destino: 249.35.9.93, 17123  
→ **Origem: 47.59.17.30, 26301; Destino: 249.35.9.93, 17123**
- v. Origem: 172.16.0.3, 31794; Destino: 160.66.226.17, 20444  
→ **Origem: 47.59.17.30, 1025; Destino: 160.66.226.17, 20444**
- vi. Origem: 172.16.0.4, 19105; Destino: 249.35.9.93, 17123  
→ **Origem: 47.59.17.30, 26301; Destino: 249.35.9.93, 17123**
- vii. Origem: 172.16.0.1, 5203; Destino: 51.235.53.124, 10210  
→ **Origem: 47.59.17.30, 1024; Destino: 51.235.53.124, 10210**
- viii. Origem: 172.16.0.1, 12503; Destino: 144.126.169.67, 8818  
→ **Origem: 47.59.17.30, 1029; Destino: 144.126.169.67, 8818**
- ix. Origem: 172.16.0.4, 19105; Destino: 249.35.9.93, 23942  
→ **Origem: 47.59.17.30, 1030; Destino: 249.35.9.93, 23942**
- x. Origem: 172.16.0.1, 5203; Destino: 226.202.247.183, 6754  
→ **Origem: 47.59.17.30, 1031; Destino: 226.202.247.183, 6754**