

Curso de Tecnologia em Sistemas de Computação
Disciplina: Redes de Computadores II
AD1 – 1º semestre de 2018 – 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 146.49.225.203 **pertence** à rede 146.0.0.0/10
- ii. O endereço 35.2.41.124 **pertence** à rede 35.0.0.0/11
- iii. O endereço 89.213.7.139 **não pertence** à rede 89.213.64.0/19
- iv. O endereço 236.238.180.69 **pertence** à rede 236.128.0.0/9
- v. O endereço 110.140.139.42 **pertence** à rede 110.0.0.0/7
- vi. O endereço 104.97.11.154 **não pertence** à rede 107.0.0.0/9
- vii. O endereço 66.217.137.140 **não pertence** à rede 66.217.192.0/19
- viii. O endereço 180.5.46.215 **pertence** à rede 180.5.46.0/23
- ix. O endereço 155.67.23.237 **não pertence** à rede 155.80.0.0/13
- x. O endereço 143.132.233.203 **não pertence** à rede 143.132.0.0/18

(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. 134.0.0.0/9 **está contida em** 134.0.0.0/7
- ii. 214.68.0.0/14 **é disjunta de** 214.0.0.0/12
- iii. 204.128.0.0/9 **é disjunta de** 207.192.0.0/13
- iv. 69.154.170.96/27 **é disjunta de** 69.154.168.0/25
- v. 69.236.255.128/27 **está contida em** 69.236.252.0/22
- vi. 130.136.0.0/13 **é disjunta de** 130.171.0.0/17
- vii. 140.126.18.0/24 **é disjunta de** 140.126.128.0/19
- viii. 203.31.46.96/27 **está contida em** 203.31.46.0/25
- ix. 63.104.0.0/17 **é disjunta de** 63.32.0.0/12

x. 196.233.224.0/19 **está contida em** 196.233.128.0/17

Questão 2 20 pontos

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

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

Faixa de endereço destino	Interface
11001110 00000111 00111000 00000000 a	0
11001110 00000111 00111001 11111111 a	1
11001110 00000111 00111001 10011111 a	0
11001110 00000111 00111001 10001111 a	2
11001110 00000111 00111001 10001000 a	2
11001110 00000111 00111001 10001011 a	3
11001110 00000111 00111001 10001001 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
11001110 00000111 0011100	0
11001110 00000111 00111001 100	1
11001110 00000111 00111001 1000	0
11001110 00000111 00111001 10001	2
11001110 00000111 00111001 100010	2
11001110 00000111 00111001 1000100	3
-	1

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

Resposta:

Prefixo	Interface
206.7.56.0/23	0
206.7.57.128/27	1
206.7.57.128/28	0
206.7.57.136/29	2
206.7.57.136/30	2
206.7.57.136/31	3
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. 206.7.57.136 → interface 3 | vii. 206.7.57.143 → interface 2 |
| ii. 206.7.57.152 → interface 1 | viii. 206.7.56.210 → interface 0 |
| iii. 206.7.57.135 → interface 0 | ix. 206.7.57.137 → interface 3 |
| iv. 206.7.62.97 → interface 1 | x. 206.7.57.133 → interface 0 |
| v. 206.7.57.135 → interface 0 | xi. 206.7.56.8 → interface 0 |
| vi. 206.7.57.142 → interface 2 | xii. 206.7.57.141 → interface 2 |

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

Resposta:

A regra do prefixo 206.7.57.136/30 é redundante.

Questão 3 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.²

i. R_1 : 4.125.136.0/21 com 5 subredes: R_{11} : 500 estações R_{12} : 400 estações R_{13} : 100 estações R_{14} : 500 estações R_{15} : 400 estações Alocação impossível	ii. R_2 : 32.188.0.0/14 com 5 subredes: R_{21} : 20000 estações 32.189.0.0/18 R_{22} : 50000 estações 32.188.0.0/16 R_{23} : 20000 estações 32.189.0.0/17 R_{24} : 10000 estações 32.190.64.0/18 R_{25} : 20000 estações 32.189.128.0/17	iii. R_3 : 23.21.0.0/16 com 5 subredes: R_{31} : 4000 estações 23.21.96.0/20 R_{32} : 12000 estações 23.21.0.0/18 R_{33} : 2000 estações 23.21.128.0/20 R_{34} : 8000 estações 23.21.64.0/19 R_{35} : 3000 estações 23.21.112.0/20
---	--	--

Questão 4 20 pontos

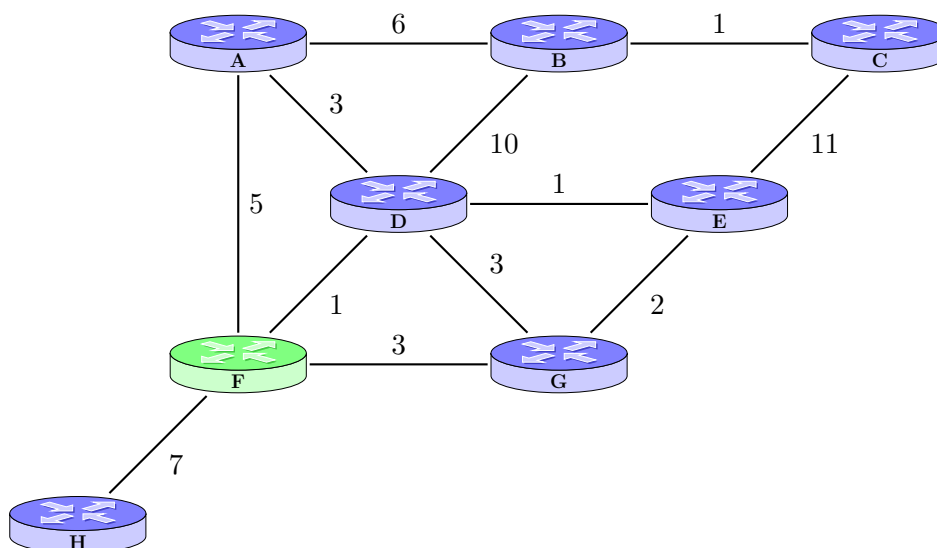
O algoritmo de Dijkstra — que leva o nome do professor holandês Edsger W. Dijkstra, que

¹Dica: procure por faixas mais específicas.

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

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.



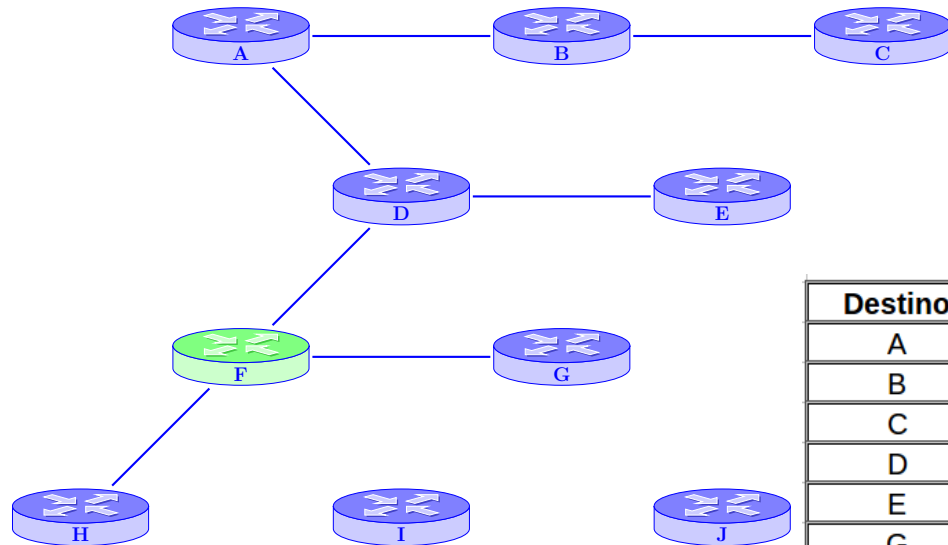
- (a) Utilizando o algoritmo de Dijkstra, calcule os caminhos mais curtos a partir do nó F, destacado em verde, para todos os outros nós da rede. Construa uma tabela igual à mostrada em aula que mostra o funcionamento do algoritmo de forma iterativa.

Resposta:

	N'	$d_A p_A$	$d_B p_B$	$d_C p_C$	$d_D p_D$	$d_E p_E$	$d_G p_G$	$d_H p_H$
0	F	5 F	∞ -	∞ -	1 F	∞ -	3 F	7 F
1	FD	4 D	11 D	∞ -		2 D	3 F	7 F
2	FDE	4 D	11 D	13 E			3 F	7 F
3	FDEG	4 D	11 D	13 E				7 F
4	FDEGA		10 A	13 E				7 F
5	FDEGAH		10 A	13 E				
6	FDEGAHB			11 B				
7	FDEGAHBC							

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

Resposta:

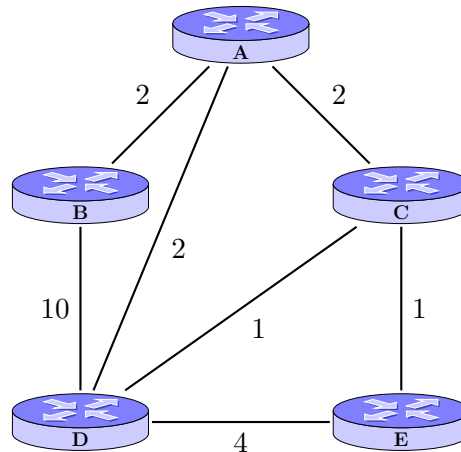


FALTA A TABELA DE ROTEAMENTO!!

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

Questão 5 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	2	∞	∞	∞	via A	2	∞	∞	∞	via A	2	∞	∞	∞	A	2	∞	∞	∞	via C	∞	∞	1	∞
via C	∞	2	∞	∞	via D	∞	∞	10	∞	via D	∞	∞	1	∞	B	∞	10	∞	∞	via D	∞	∞	∞	4
via D	∞	∞	2	∞						via E	∞	∞	∞	1	C	∞	∞	1	∞					
															E	∞	∞	∞	4					

vetor de A					vetor de B					vetor de C					vetor de D					vetor de E				
A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
0	2	2	2	∞	2	0	∞	10	∞	2	∞	0	1	1	2	10	1	0	4	∞	∞	1	4	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	2	∞	12	∞	via A	2	4	4	∞	via A	2	4	4	∞	A	2	4	4	∞	via C	3	∞	1	2
via C	∞	2	3	3	via D	12	11	10	14	via D	3	11	1	5	B	12	10	∞	∞	via D	6	14	5	4
via D	12	3	2	6						via E	∞	∞	5	1	C	3	∞	1	2					
															E	∞	∞	5	4					

vetor de A					vetor de B					vetor de C					vetor de D					vetor de E				
A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
0	2	2	2	3	2	0	4	4	14	2	4	0	1	1	2	4	1	0	2	3	14	1	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.

Passo 2:

A	custo até	B	custo até	C	custo até	D	custo até	E	custo até
	B C D E		A C D E		A B D E		A B C E		A B C D
via B	2 6 6 16	via A	2 4 4 5	via A	2 4 4 5	via A	2 4 4 5	via C	3 5 1 2
C	6 2 3 3	D	12 11 10 12	D	3 5 1 3	B	12 10 14 24	D	6 8 5 4
D	6 3 2 4			E	4 15 3 1	C	3 5 1 2		
						E	7 18 5 4		

vetor de A	vetor de B	vetor de C	vetor de D	vetor de E
A B C D E	A B C D E	A B C D E	A B C D E	A B C D E
0 2 2 2 3	2 0 4 4 5	2 4 0 1 1	2 4 1 0 2	3 5 1 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	2 6 6 7	via A	2 4 4 5	via A	2 4 4 5	via A	2 4 4 5	via C	3 5 1 2
C	6 2 3 3	D	12 11 10 12	D	3 5 1 3	B	12 10 14 15	D	6 8 5 4
D	6 3 2 4			E	4 6 3 1	C	3 5 1 2		
						E	7 9 5 4		

vetor de A	vetor de B	vetor de C	vetor de D	vetor de E
A B C D E	A B C D E	A B C D E	A B C D E	A B C D E
0 2 2 2 3	2 0 4 4 5	2 4 0 1 1	2 4 1 0 2	3 5 1 2 0

Questão 6 10 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 16.78.249.206 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, 22026	87.121.79.103, 19712	28220
(2)	172.16.0.1, 21848	70.78.116.17, 9042	28246
(3)	172.16.0.2, 11813	193.42.189.45, 2985	1028
(4)	172.16.0.1, 7157	196.198.141.176, 10555	20508
(5)	172.16.0.2, 26787	226.243.48.226, 26704	19395
(6)	172.16.0.1, 9604	164.23.72.116, 12072	12696
(7)	172.16.0.1, 18652	55.83.92.134, 2127	15450
(8)	172.16.0.1, 23578	8.215.140.49, 16291	29019
(9)	172.16.0.3, 18855	78.77.121.159, 16815	17935
(10)	172.16.0.2, 3375	228.76.22.61, 19855	1024

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: 70.78.116.17, 9042; Destino: 16.78.249.206, 28246
 → **Origem: 70.78.116.17, 9042; Destino: 172.16.0.1, 21848**
- ii. Origem: 8.215.140.49, 16291; Destino: 16.78.249.206, 29019
 → **Origem: 8.215.140.49, 16291; Destino: 172.16.0.1, 23578**
- iii. Origem: 25.26.183.57, 27002; Destino: 16.78.249.206, 29019
descartado
- iv. Origem: 193.42.189.45, 2985; Destino: 16.78.249.206, 1028
 → **Origem: 193.42.189.45, 2985; Destino: 172.16.0.2, 11813**
- v. Origem: 219.160.238.201, 10608; Destino: 16.78.249.206, 12696
descartado
- vi. Origem: 78.77.121.159, 16815; Destino: 16.78.249.206, 28220
descartado
- vii. Origem: 164.23.72.116, 12072; Destino: 16.78.249.206, 12696
 → **Origem: 164.23.72.116, 12072; Destino: 172.16.0.1, 9604**
- viii. Origem: 87.121.79.103, 19712; Destino: 16.78.249.206, 28220
 → **Origem: 87.121.79.103, 19712; Destino: 172.16.0.1, 22026**
- ix. Origem: 188.236.32.121, 10780; Destino: 16.78.249.206, 1028
descartado
- x. Origem: 78.77.121.159, 16815; Destino: 16.78.249.206, 17935
 → **Origem: 78.77.121.159, 16815; Destino: 172.16.0.3, 18855**

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

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

- i. Origem: 172.16.0.2, 26787; Destino: 147.202.186.184, 28005

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(11)	172.16.0.2, 26787	147.202.186.184, 28005	1025

- ii. Origem: 172.16.0.3, 18855; Destino: 78.77.121.159, 16815

Encaminhado segundo a entrada (9)

- iii. Origem: 172.16.0.1, 9624; Destino: 8.215.140.49, 16291

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(12)	172.16.0.1, 9624	8.215.140.49, 16291	1026

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

- iv. Origem: 172.16.0.1, 9585; Destino: 196.198.141.176, 10555

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(13)	172.16.0.1, 9585	196.198.141.176, 10555	1027

- v. Origem: 172.16.0.1, 18652; Destino: 55.83.92.134, 9822

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(14)	172.16.0.1, 18652	55.83.92.134, 9822	1029

- vi. Origem: 172.16.0.2, 26787; Destino: 226.243.48.226, 26704

Encaminhado segundo a entrada (5)

- vii. Origem: 172.16.0.2, 26787; Destino: 226.243.48.226, 10730

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(15)	172.16.0.2, 26787	226.243.48.226, 10730	1030

- viii. Origem: 172.16.0.1, 18652; Destino: 55.83.92.134, 2127

Encaminhado segundo a entrada (7)

- ix. Origem: 172.16.0.1, 22026; Destino: 87.121.79.103, 27155

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(16)	172.16.0.1, 22026	87.121.79.103, 27155	1031

- x. Origem: 172.16.0.1, 18652; Destino: 55.83.92.134, 12947

Nova entrada:

	(IP, porta) local	(IP, porta) destino	Porta pública no NAT
(17)	172.16.0.1, 18652	55.83.92.134, 12947	1032

- (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: 172.16.0.2, 26787; Destino: 147.202.186.184, 28005
 → **Origem: 16.78.249.206, 1025; Destino: 147.202.186.184, 28005**
- ii. Origem: 172.16.0.3, 18855; Destino: 78.77.121.159, 16815
 → **Origem: 16.78.249.206, 17935; Destino: 78.77.121.159, 16815**
- iii. Origem: 172.16.0.1, 9624; Destino: 8.215.140.49, 16291
 → **Origem: 16.78.249.206, 1026; Destino: 8.215.140.49, 16291**
- iv. Origem: 172.16.0.1, 9585; Destino: 196.198.141.176, 10555
 → **Origem: 16.78.249.206, 1027; Destino: 196.198.141.176, 10555**
- v. Origem: 172.16.0.1, 18652; Destino: 55.83.92.134, 9822
 → **Origem: 16.78.249.206, 1029; Destino: 55.83.92.134, 9822**
- vi. Origem: 172.16.0.2, 26787; Destino: 226.243.48.226, 26704
 → **Origem: 16.78.249.206, 19395; Destino: 226.243.48.226, 26704**
- vii. Origem: 172.16.0.2, 26787; Destino: 226.243.48.226, 10730
 → **Origem: 16.78.249.206, 1030; Destino: 226.243.48.226, 10730**
- viii. Origem: 172.16.0.1, 18652; Destino: 55.83.92.134, 2127
 → **Origem: 16.78.249.206, 15450; Destino: 55.83.92.134, 2127**

- ix. Origem: 172.16.0.1, 22026; Destino: 87.121.79.103, 27155
 → **Origem: 16.78.249.206, 1031; Destino: 87.121.79.103, 27155**
- x. Origem: 172.16.0.1, 18652; Destino: 55.83.92.134, 12947
 → **Origem: 16.78.249.206, 1032; Destino: 55.83.92.134, 12947**