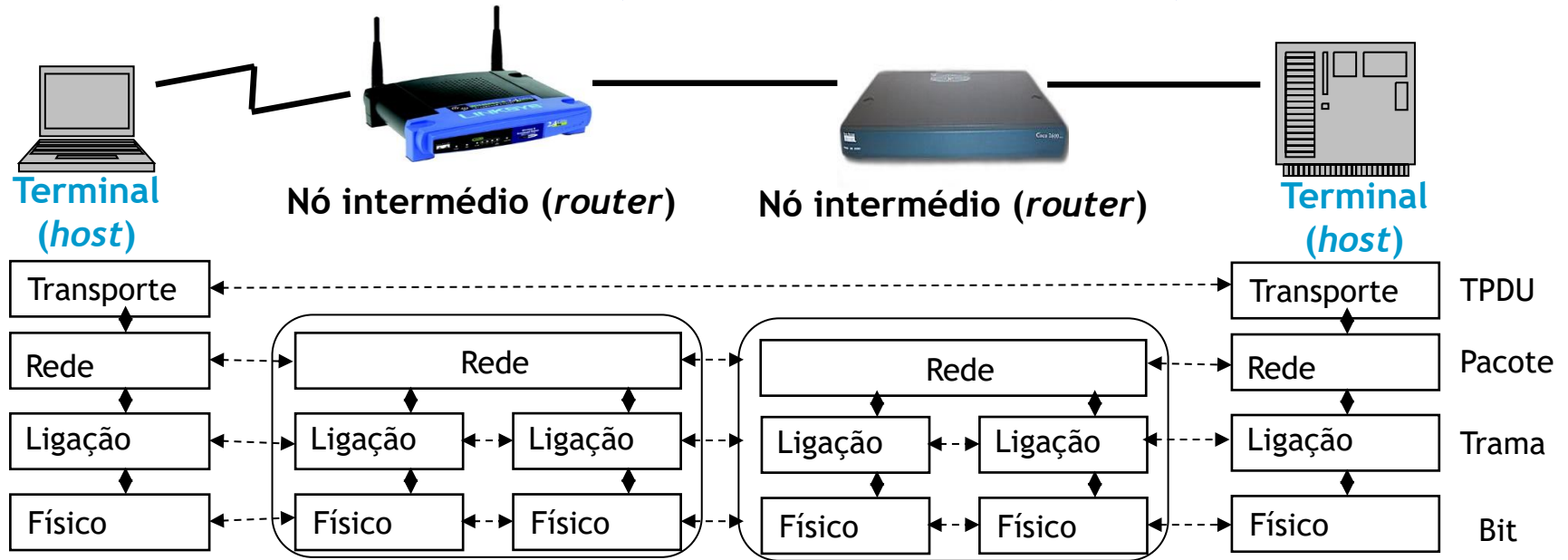


CP5 e CP6: Nível de Rede e Nível de Rede na Internet (refs.)

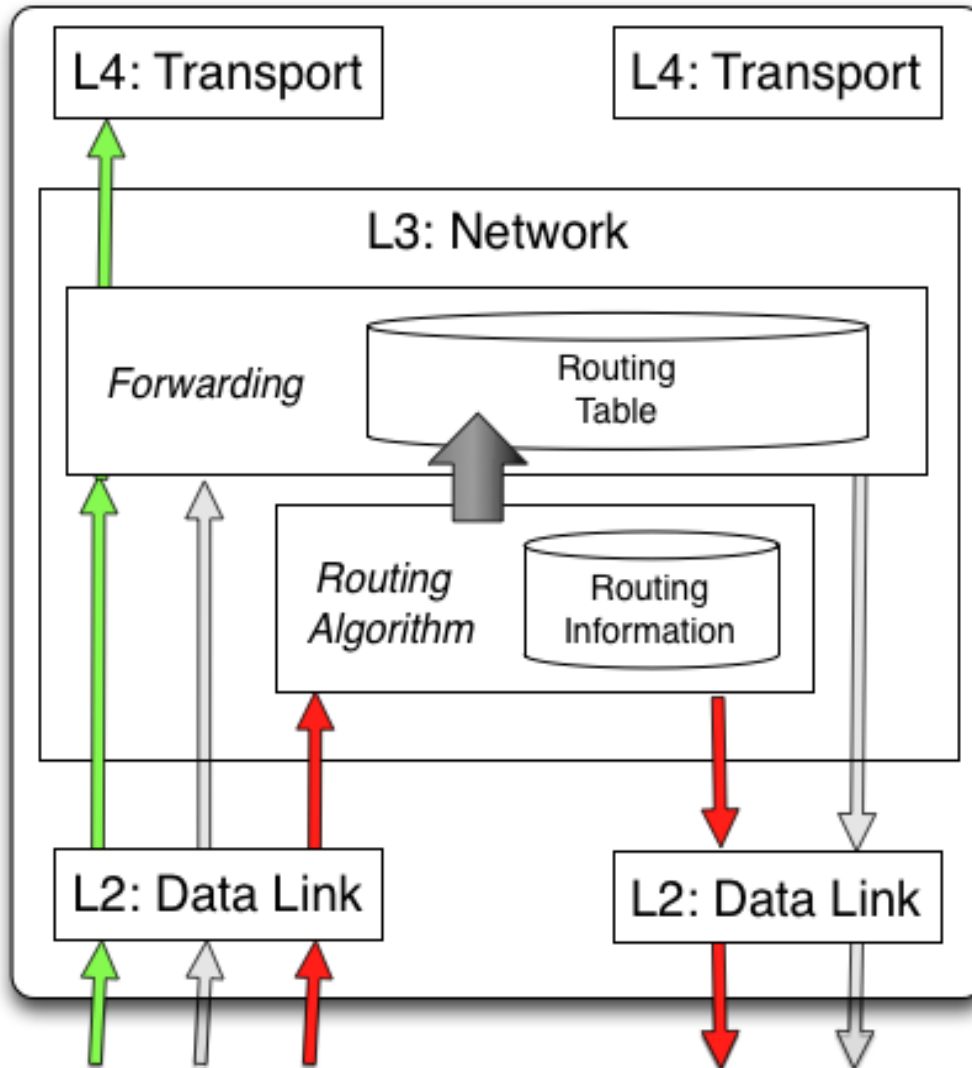
- Funções do Nível de Rede
(Peterson 2021, 3.1; Tanenbaum 2011, 5.1)
- Algoritmos de encaminhamento
(Peterson 2021, 3.4; Tanenbaum 2011, 5.2)
- Protocolos de encaminhamento
(Peterson 2021, 3.4; Tanenbaum 2011, 5.6)
- O protocolo IP
(Peterson 2021, 3.3; Tanenbaum 2011, 5.6)
- O protocolo ICMP, ARP e DHCP
(Peterson 2021, 3.3; Tanenbaum 2011, 5.6)

Nível de Rede (*Network Layer*)



- Utiliza os serviços do nível L2 (*Data Link*), presta serviços ao nível L4 (*Transport*).
Os serviços prestados ao nível transporte devem ser independentes da tecnologia da sub-rede.
O número, tipo e topologia das sub-redes que constituem o nível rede deve ser escondido ao nível transporte.
Todos os nós têm de ter pelo menos os níveis L1-L2-L3 implementados
- A unidade protocolar fundamental é o pacote (*packet*).
- O nível de rede preocupa-se com o percurso dos pacotes desde a origem através da rede até ao destino (*end-to-end*)
O percurso pode ser composto por vários nós intermédios - encaminhadores (*routers*).

Funcionalidades fundamentais do Nível de Rede



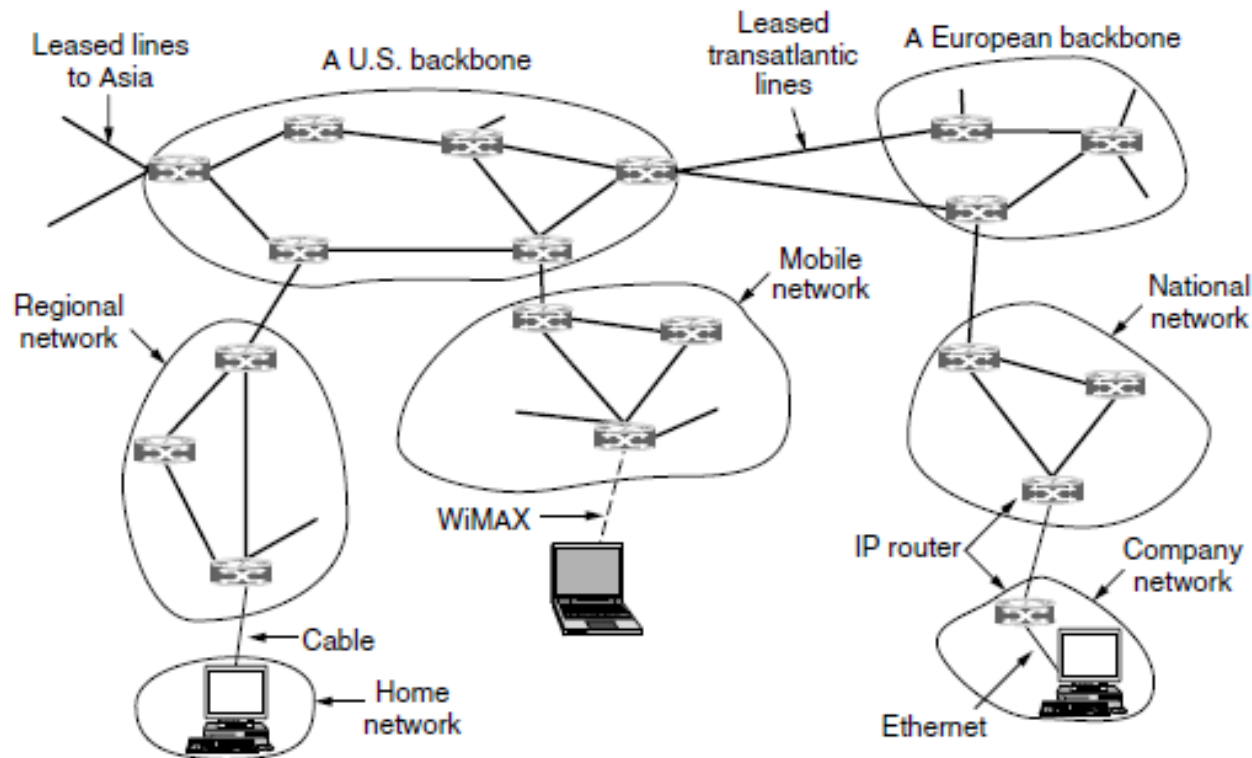
• **Expedição (*forwarding*):** envio dos pacotes para o próximo nó:

- Por inundação
- Por consulta de uma tabela de encaminhamento (*routing table*).
(Atenção à diferença entre os processos de expedição e de encaminhamento)

• **Encaminhamento (*routing*):** construção da tabela de encaminhamento

- Necessita de conhecer a topologia da sub-rede (os nós) e escolher o caminho apropriado através dos nós.
- Diferentes estratégias/algoritmos para recolher informação sobre a topologia da rede e para o cálculo do caminho através da rede.

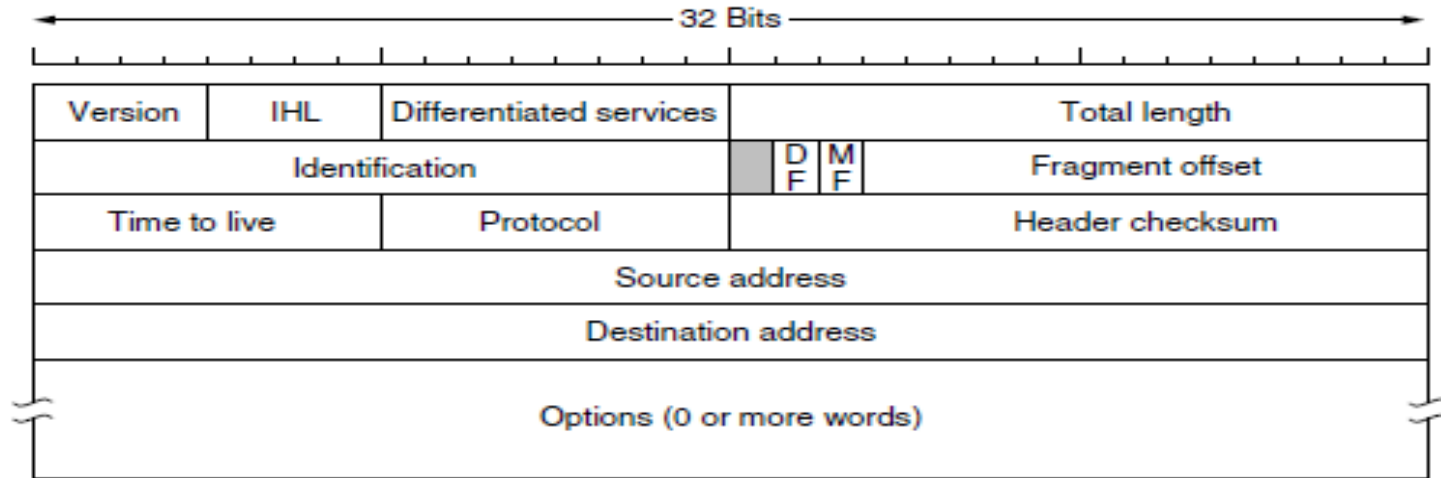
Internet: Colecção de Subredes



O Nível de Rede na Internet

- Protocolo IPv4
- Endereços IPv4
- Protocolos de controlo da Internet
 - ICMP
 - ARP
 - RARP, BOOTP, DHCP
- Protocolos de encaminhamento
 - *Routing Internet Protocol* (RIP)
 - *Open Shortest Path First* (OSPF)
 - *Border Gateway Protocol* (BGP)

O Protocolo IPv4



Versão: max. 16 (actualmente IPv4 e IPv6))

IHL: comp. do *header* (em palavras de 32 bits)

Differentiated services: os primeiros 6 bits indicam a classe de serviço, enquanto que os últimos 2 bits indicam informação sobre congestão na rede.

Comprimento: comprimento total do datagrama (em bytes)
(Máx: 65535, Máx típico 1500, Típico 576)

Identificação: Identifica o datagrama a que pertence

DF: não fragmentar **MF:** a zero no último fragmento

Frag.offset: frag são múltiplos de 8 bytes, excepto o último

Tempo de vida: max. 255 (dec. por “salto”)

Protocolo: TCP,UDP,... [RFC1700, RFC3232]

Checksum do cabeçalho: verificado e recalculado em cada salto

Opções: (não presentes no IPv6)

Security: qual a segurança do datagrama

Strict Source Routing: indica qual o caminho completo a seguir pelo datagrama

Loose Source Routing: o datagrama tem que passar pelos nós pela ordem indicada, (pode passar por outros)

Record Route: os nós registam o seu endereço

Timestamp: os nós registam o seu endereço e o tempo (4 bytes + 4 bytes)

Fragmentação de Datagramas IP

Exercício 4.3

- Transferência de um datagrama com 4000 bytes (ID = 777) através de uma ligação com MTU = 1500 bytes => Fragmentação do datagrama em 3 fragmentos

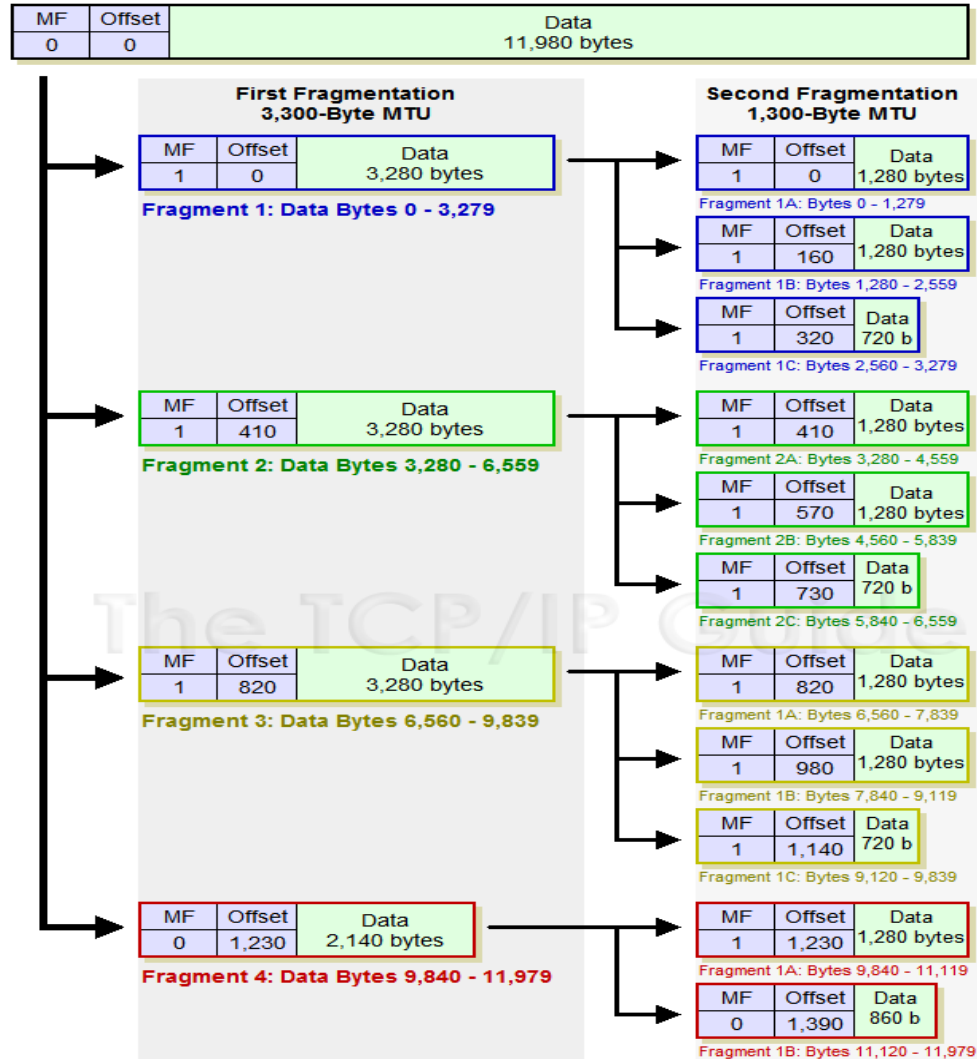
Fragmento	Bytes	ID	Offset	Flag MF
1	Header - 20; Data - 1480	777	0	1
2	Header - 20; Data - 1480	777	185*	1
3	Header - 20; Data - 1020 (1020=4000-20-1480-1480)	777	370**	0

* 185 = 1480/8

** 370 = 2960/8

MTU - Maximum Transfer Unit

Exemplo da fragmentação de um pacote IP com 12000 bytes (incluindo o cabeçalho IP de 20 bytes) enviada através de um link com um MTU = 3300 a que se segue um link com MTU = 1300



Exercício 4.7

Considere o diagrama de rede IP/Ethernet apresentado em que a máquina A pretende enviar 1 mensagem com dimensão total de 2350 bytes para a máquina B (DF=0).

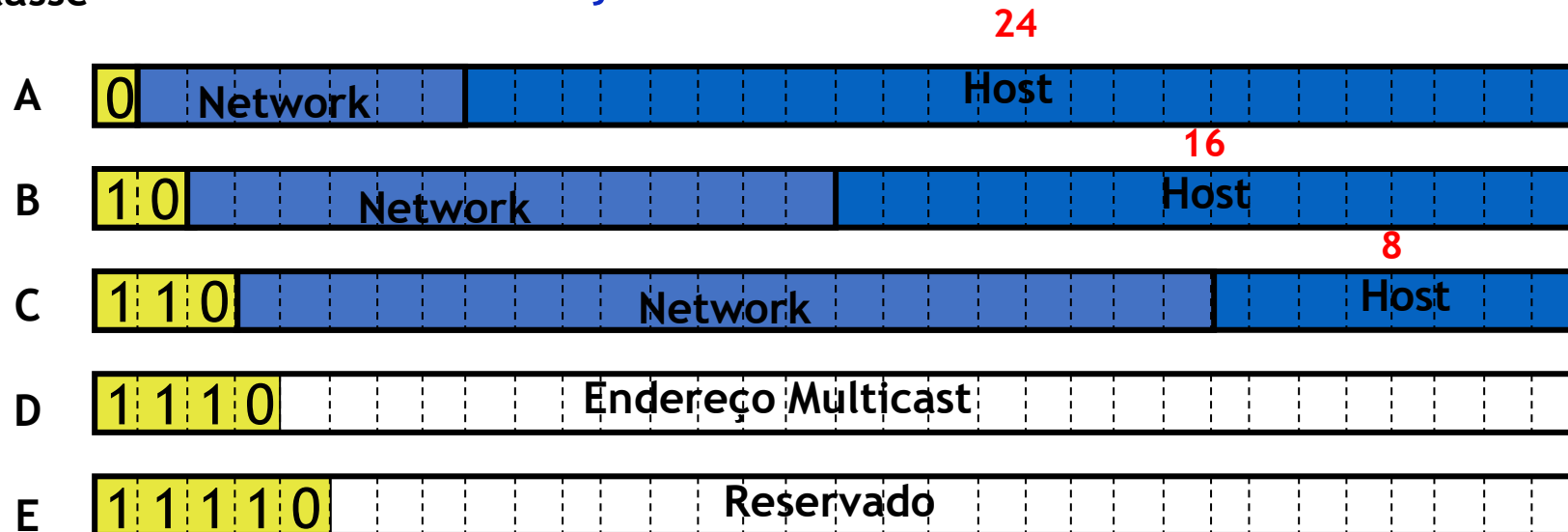
Tendo em conta os valores de MTU das redes Lan1 e Lan 2 indique todos os fragmentos que irão chegar a B considerando que os cabeçalhos IP têm sempre 20 bytes.



MTU=1024			MTU=512		
Payload	Total	offset	Payload	Total	offset
1000	1020	0	488	508	0
			488	508	61
			24	44	122
1000	1020	125	488	508	125
			488	508	186
			24	44	247
350	370	250	350	370	250

Endereços IPv4

Classe



Classe	Nº redes	Nº hosts	%
A:	128	16×10^6	50
B:	16834	65×10^3	25
C:	2×10^6	254	12.5

A:	1.0.0.0 a 127.255.255.255
B:	128.0.0.0 a 191.255.255.255
C:	192.0.0.0 a 223.255.255.255
D:	224.0.0.0 a 239.255.255.255
E:	240.0.0.0 a 255.255.255.255

Endereço IPv4: [0.0.0.0] a [255.255.255.255]

Uma máquina no ISCTE: [193.136.188.1]

Uma rede classe C no ISCTE: [193.136.188.0]

Este computador: [0.0.0.0]

Uma máquina nesta rede: [0..0|host]₍₂₎

Loopback: [127.x.x.x]

Difusão local: [255.255.255.255]

Difusão numa rede distante: [net|1...1]₍₂₎

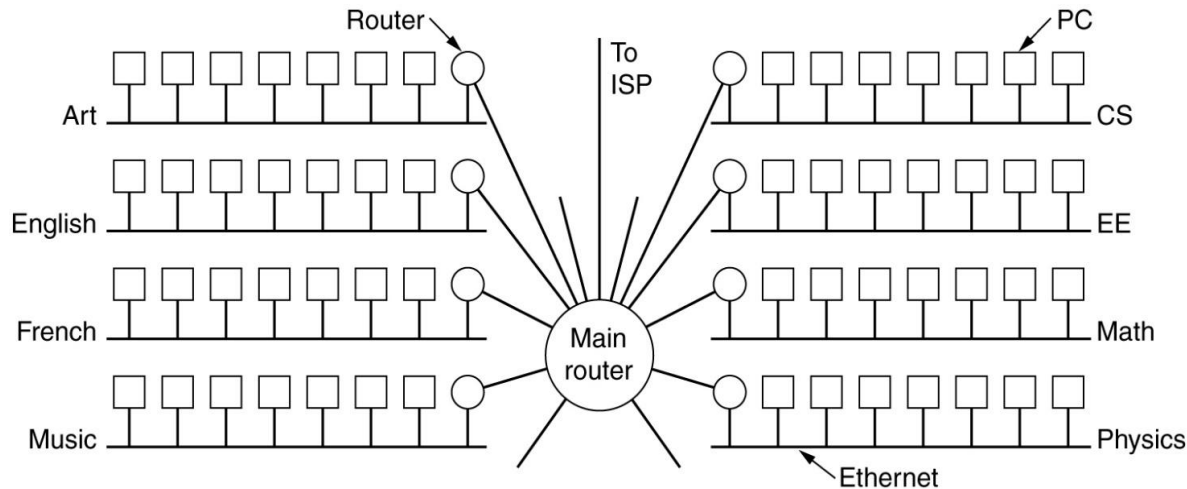
Redes Privadas:

Classe A (1) - [10.0.0.0 - 10.255.255.255]

Classe B (16) - [172.16.0.0 - 172.31.255.255]

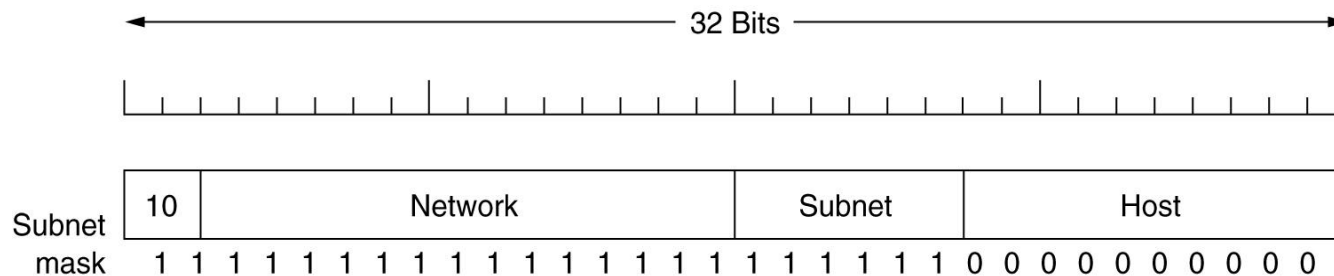
Classe C (256) - [192.168.0.0 a 192.168.255.255]

Subredes (*subnetting*)



Encaminhamento: i) Rede, ii) Subrede, iii) Estação.

Ex: Uma rede de classe B subdividida em 64 subredes.



Agrupamento de Endereços (*supernetting*)

- Agrupamento de vários blocos de uma determinada classe
 - Muito comum com blocos da classe C
- Atribuição de endereços
 - Comum atribuir endereços de forma contígua
 - Definição de uma máscara de rede
 - Primeiro endereço do bloco agregado (i.e., o endereço de rede) tem de ser divisível pelo tamanho do superbloco (\Rightarrow bits de *host* todos a zero).
 - Ex., Redes 193.168.2.0 e 193.168.3.0
 - Tamanho do superbloco: 512 endereços
 - Primeiro endereço do superbloco : 193.168.2.0 (divisível por 512)
 - Máscara de rede: 255.255.254.0

```
11111111 11111111 11111110 00000000
```

Máscara de rede
do superbloco

```
11111111 11111111 11111111 00000000
```

Máscara de rede
da Classe C

Classless Addressing

- Abandono das classes (i.e., blocos de endereços de tamanho variável)
- Condições:
 - O número de endereços num bloco é uma potência de 2
 - O primeiro endereço do bloco tem de ser divisível pelo tamanho do bloco
 - Para cada bloco deve ser definida a máscara respectiva
- Notação / (*Classless InterDomain Routing*):
 - w.x.y.z/n (n, tamanho do prefixo, número de 1s na máscara)
- *Subnetting*:
 - Continua a ser possível fazer *subnetting* dentro de cada rede
- Antes:
 1. Descobrir a Classe (4 bit)
 2. Descobrir a Rede (*Shift* 24, 16, ou 8 bits)
 3. Pesquisar na tabela (Classe A e B); *Hash Tables* (Classe C)

Exercício 4.4

Considere dois dispositivos A e B com os seguintes endereços IP em notação CIDR:

A – 221.188.73.3/23

B – 221.188.72.2/23

- a) Indique, justificando, se os dispositivos A e B pertencem ou não à mesma rede e os respectivos endereços de rede e de difusão na rede a que pertencem.
- b) Pretende-se dividir a rede onde se encontra o dispositivo A em 8 sub-redes distintas. Nestas condições indique justificadamente:
 - i. Qual a máscara de sub-rede a ser utilizada.
 - ii. Qual a gama de endereços que podem ser atribuídos a dispositivos na sub-rede de A.

Address: 221.188.73.3 11011101.10111100.0100100 1.00000011
 Netmask: 255.255.254.0 = 23 11111111.11111111.11111111 0.00000000

Network: 221.188.72.0/23 11011101.10111100.0100100 0.00000000
 HostMin: 221.188.72.1 11011101.10111100.0100100 0.00000001
 HostMax: 221.188.73.254 11011101.10111100.0100100 1.11111110
 Broadcast: 221.188.73.255 11011101.10111100.0100100 1.11111111
 Hosts/Net: 510 Class C

Address: 221.188.72.2 11011101.10111100.0100100 0.00000010
 Netmask: 255.255.254.0 = 23 11111111.11111111.11111111 0.00000000

Network: 221.188.72.0/23 11011101.10111100.0100100 0.00000000
 HostMin: 221.188.72.1 11011101.10111100.0100100 0.00000001
 HostMax: 221.188.73.254 11011101.10111100.0100100 1.11111110
 Broadcast: 221.188.73.255 11011101.10111100.0100100 1.11111111
 Hosts/Net: 510 Class C

Net A = Net B = 221.188.72.0/23 => Host A e Host B pertencem à mesma rede.
 Broadcast: 221.188.73.255

b) 8 sub-redes => 3 bits

SubNetmask: 255.255.255.192 = 26 11111111.11111111.11111111 1.11 000000
 Subnetwork: 221.188.73.0/26 11011101.10111100.0100100 1.00 000000 (Subnet 4)
 HostMin: 221.188.73.1 11011101.10111100.0100100 1.00 000001
 HostMax: 221.188.73.62 11011101.10111100.0100100 1.00 111110
 Broadcast: 221.188.73.63 11011101.10111100.0100100 1.00 111111
 Hosts/Net: 62 Class C

Nota: A pertence à Sub-Net 4 (100) e B pertence à Sub-Net 0 (000)

Usar a aplicação *ipcalc*

Exercício 4.8

Considere um cenário em que a unidade mínima que uma organização pode adquirir em termos de blocos de endereços IP é um endereço de rede de classe C.

A organização A detém o endereço de rede ao qual pertence a estação com o endereço 193.168.1.151.

- Indique qual o número máximo de estações que pode possuir e qual o endereço de rede e de difusão nessa rede.
- Admitindo que pretende expandir a rede para suportar 600 estações, qual a gama de endereços que deve adquirir?

Nas condições da alínea anterior, indique qual a nova máscara de rede, o endereço de rede, o endereço de difusão e a gama de endereços que podem ser atribuídos a *hosts* (estações) na nova rede.

a)

Address:	193.168.1.151	11000001.10101000.00000001. 10010111
Netmask:	255.255.255.0 = 24	11111111.11111111.11111111. 00000000
Wildcard:	0.0.0.255	00000000.00000000.00000000. 11111111
=>		
Network:	193.168.1.0/24	11000001.10101000.00000001. 00000000
HostMin:	193.168.1.1	11000001.10101000.00000001. 00000001
HostMax:	193.168.1.254	11000001.10101000.00000001. 11111110
Broadcast:	193.168.1.255	11000001.10101000.00000001. 11111111
Hosts/Net:	254	Class C

Neste caso, é necessário agrupar os seguintes endereços:

Network:	193.168.0.0/24	11000001.10101000.00000000. 00.00000000	
Network:	193.168.1.0/24	11000001.10101000.00000000. 01.00000000	<- Endereço actual
Network:	193.168.2.0/24	11000001.10101000.00000000. 10.00000000	
Network:	193.168.3.0/24	11000001.10101000.00000000. 11.00000000	

Nova Rede:

Address:	193.168.1.151	11000001.10101000.00000000 01.10010111
Netmask:	255.255.252.0 = 22	11111111.11111111.111111 00.00000000
Wildcard:	0.0.3.255	00000000.00000000.000000 11.11111111
=>		
Network:	193.168.0.0/22	11000001.10101000.00000000 00.00000000
HostMin:	193.168.0.1	11000001.10101000.00000000 00.00000001
HostMax:	193.168.3.254	11000001.10101000.00000000 11.11111110
Broadcast:	193.168.3.255	11000001.10101000.00000000 11.11111111
Hosts/Net:	1022	Class C

b)

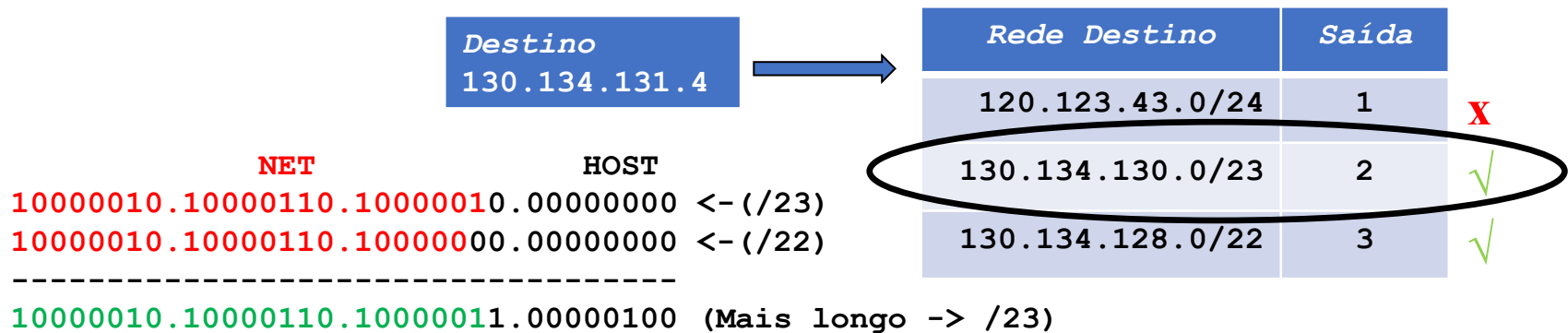
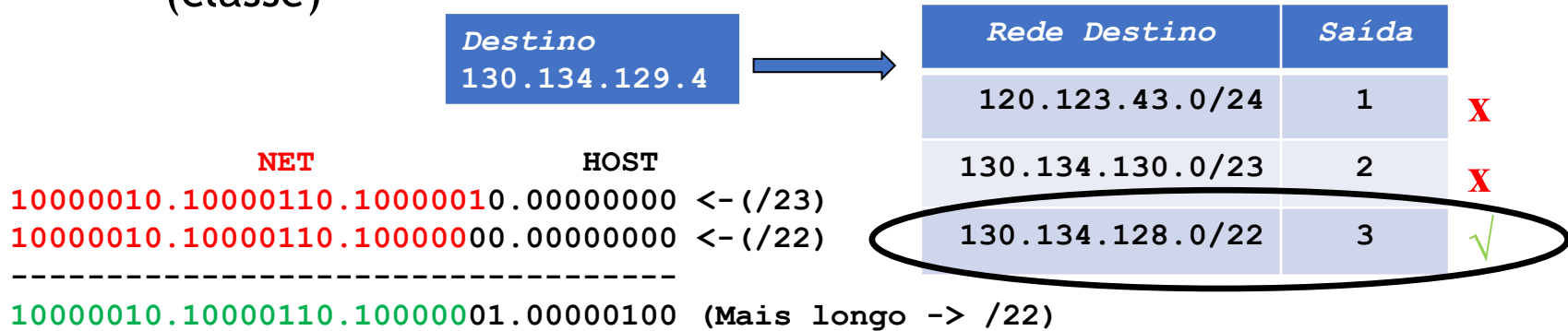
Nº de bits de *host* mínimo = 10 ($2^9 = 512 < 600 < 2^{10} = 1024$) => rede /22

CrITÉRIOS para se efectuar o agrupamento de endereços:

- O agrupamento tem que ser efectuado em grupos de potências de 2 (i.e., 2^0 , 2^1 , 2^2 , ...)
- A gama de endereços do grupo tem que ser contínua
- O primeiro endereço do grupo tem que possuir todos os bits de *host* a '0'

Classless InterDomain Routing (CIDR)

- Mudanças nas tabelas de encaminhamento e emparelhamento
É usado *longest prefix match*, i.e., não existe máscara por omissão (classe)



Classless InterDomain Routing (CIDR)

- Entrada por omissão

Todos os pacotes
emparelham com
esta entrada

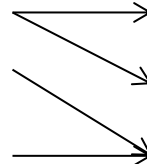


Rede Destino	Saída
120.123.43.0/24	1
130.134.130.0/23	2
130.134.128.0/22	3
0.0.0.0/0	4



- Redução/aumento no tamanho das tabelas de encaminhamento

Rede Destino	Saída
120.123.0.0	1
130.134.129.0	2
130.134.128.0	2



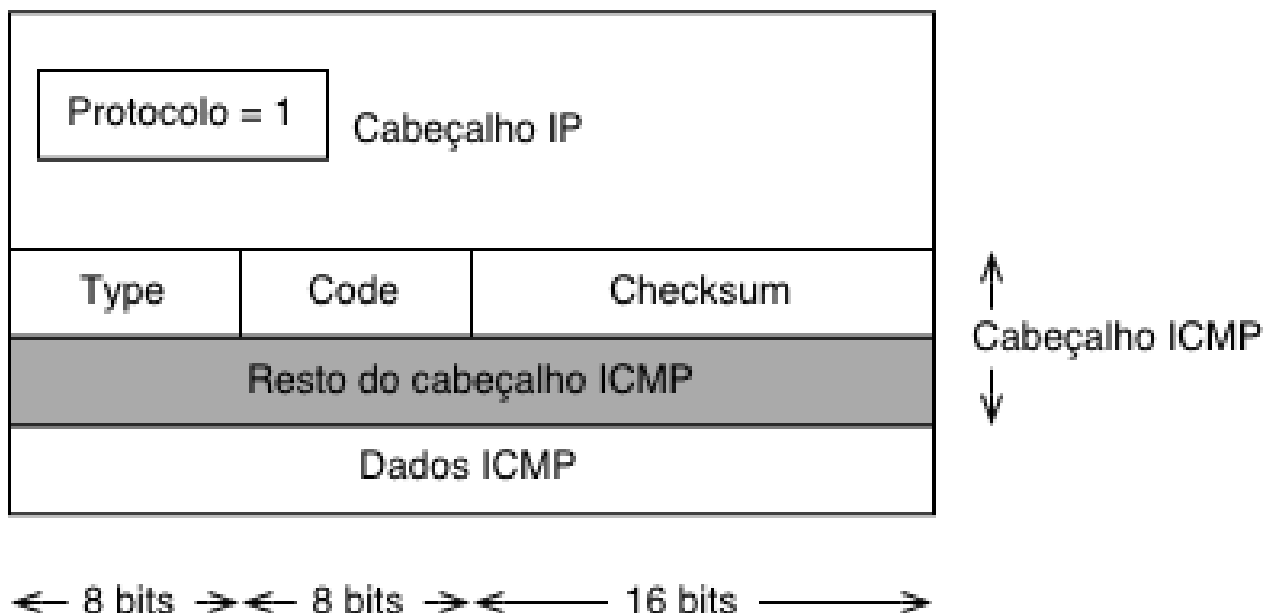
Rede Destino	Saída
120.123.0.0/17	1
120.123.128.0/17	3
130.134.128.0/23	2

Sub-divisão
de tráfego

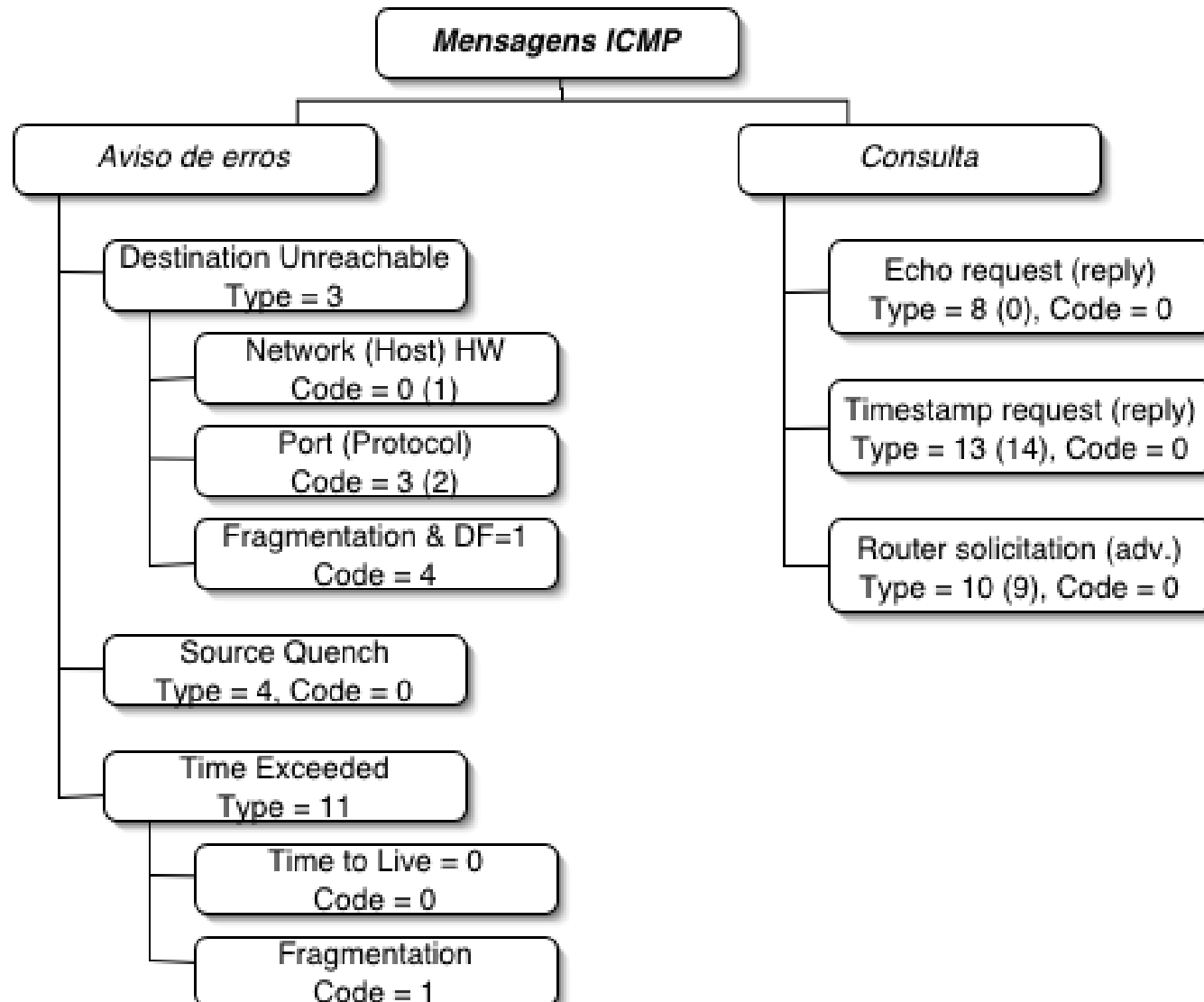
Agrupamento

Internet Control Message Protocol - ICMP

- Protocolo de Controlo da Internet
- Teste da Internet e aviso de anomalias
- Formato de um pacote ICMP
 - No caso de mensagens ICMP de erro os dados são a cópia do cabeçalho IP e dos primeiros bytes do fragmento que causou/sofreu o erro

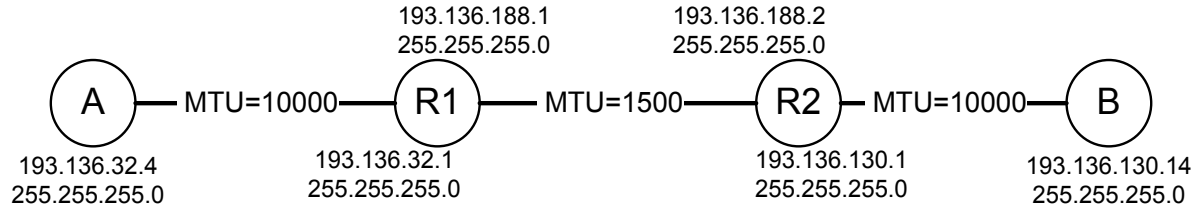


Internet Control Message Protocol - ICMP



Exercício 4.5

Considere a seguinte rede na qual se apresentam duas estações e dois *routers* com os respectivos endereços IP e máscaras de rede, bem como o valor em bytes da unidade máxima de transferência (*Maximum Transfer Unit – MTU*) em cada ligação.



(i) {P1}
DF = 0
TTL ≥ 3
L = 1000 < 1500 (é enviado um pacote)

Para o seguinte conjunto de quatro pacotes, indique, justificadamente, para cada um dos pacotes enviados pelo nó A quantos pacotes chegam ao nó B.

Ver. = 4	IHL = 5	TOS= --			Comprimento Total = 1000 bytes		
Identificação = 0					DF=0	MF=0	Offset = 0
TTL = 4		Protocol = --		Checksum = --			
Origem = 193.136.32.4							
Destino = 193.136.130.14							
Dados (980 bytes)							i)

Ver. = 4	IHL = 5	TOS= --			Comprimento Total = 2520 bytes		
Identificação = 1					DF=0	MF=0	Offset = 0
TTL = 4		Protocol = --		Checksum = --			
Origem = 193.136.32.4							
Destino = 193.136.130.14							
Dados (2500 bytes)							ii)

(ii) {P1a, P1b}
DF = 0
TTL ≥ 3
L = 2520 > 1500 (pacote é fragmentado em dois:
La = 1500; Lb = 1040)

Ver. = 4	IHL = 5	TOS= --			Comprimento Total = 2520 bytes		
Identificação = 2					DF=1	MF=0	Offset = 0
TTL = 4		Protocol = --		Checksum = --			
Origem = 193.136.32.4							
Destino = 193.136.130.14							
Dados (2500 bytes)							
iii)							

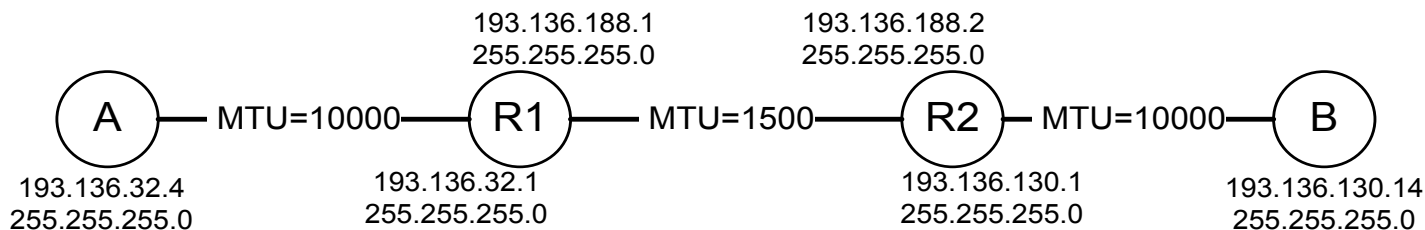
Ver. = 4	IHL = 5	TOS= --			Comprimento Total = 2520 bytes		
Identificação = 3					DF=0	MF=0	Offset = 0
TTL = 1		Protocol = --		Checksum = --			
Origem = 193.136.32.4							
Destino = 193.136.130.14							
Dados (2500 bytes)							
iv)							

(iii) ∅
L > 1500 & DF = 1 (pacote não pode ser fragmentado => é descartado)

(iv) ∅
TTL = 1 < 3 (pacote é descartado antes de chegar ao destino)

Exercício 4.6

Considere a seguinte rede na qual se apresentam duas estações e dois *routers* com os respectivos endereços IP e máscaras de rede, bem como o valor em bytes da unidade máxima de transferência (*Maximum Transfer Unit – MTU*) em cada ligação. Considera-se ainda que os *routers* *R1* e *R2* utilizam o protocolo ICMP para o controlo da rede.



- a) Represente os cabeçalhos IPv4 dos pacotes **enviados** pelos *routers* *R1* e *R2* quando o nó A envia um pacote IPv4 cujo cabeçalho é representado na figura seguinte:

Ver. = 4	IHL = 5	TOS= --			Comprimento Total = 2520 bytes			
Identificação = 0						DF=0	MF=0	Offset = 0
TTL = 4		Protocol = --			Checksum = --			
Origem = 193.136.32.4								
Destino = 193.136.130.14								
Dados (2500 bytes)								

Repita a alínea anterior considerando agora que o valor do campo *Time to Live – TTL* é **2**

Exercício 4.6 (resolução)

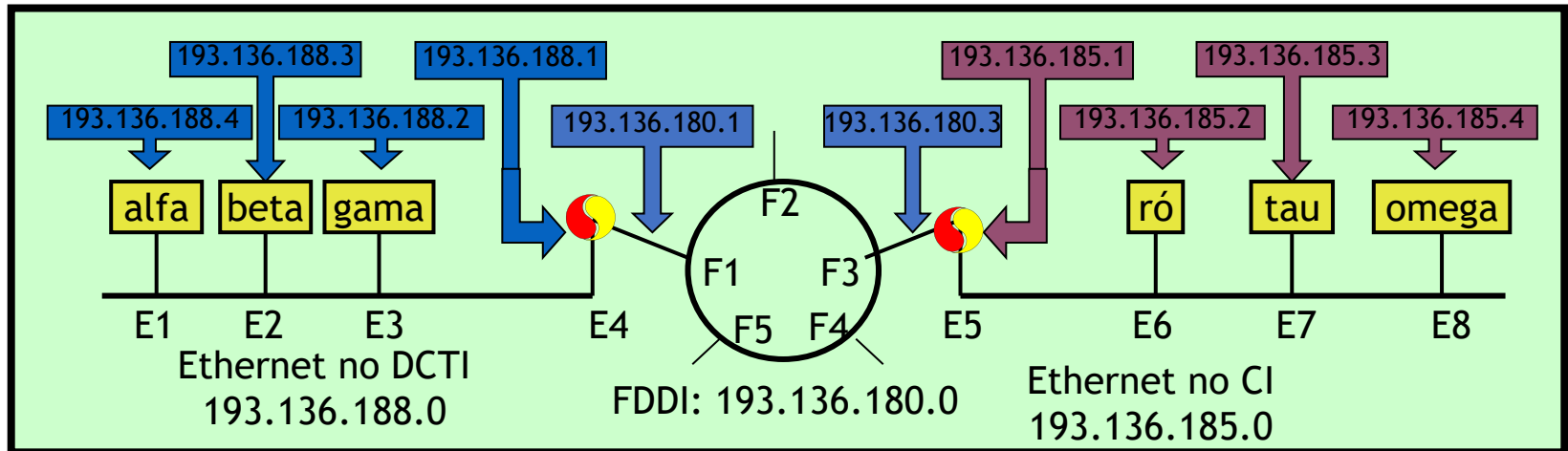
1a R1 -> P1a + P1b
R2 -> P2a + P2b

	R1		R2	
	P1a	P1b	P2a	P2b
L	1500	1040	1500	1040
MF	1	0	1	0
Offset	0	1480	0	1480
TTL	3	3	2	2
End. Orig.	193.136.32.4	193.136.32.4	193.136.32.4	193.136.32.4
End. Dest.	193.136.130.14	193.136.130.14	193.136.130.14	193.136.130.14

1b Pacote descartado em R2 pq TTL = 0
 Mensagem ICMP R2 -> A (Type:11 - Time Exceeded; Code: 0 - TTL=0)
 End. Orig ICMP (R2); End. Dest ICMP (A)
 L = 56 bytes 20 (IP header) +
 8 (ICMP header) +
 28 (Received IP packet header + 8 bytes packet data)

Address Resolution Protocol - ARP

- O endereço IP não pode ser usado para enviar tramas pois o “*Data Link Layer*” não os compreende. (e.g., Ethernet 48-bit)
- É necessário mapear o endereço IP num endereço DL (e.g., *Ethernet*, FDDI, etc.)



alfa \Rightarrow beta

(alfa constrói um pacote com o endereço de destino = 193.136.188.3)

Optimizações:

- *Cache ARP results for a period*
- *Broadcast Ethernet address at boot*

- 1) Existir um ficheiro com o mapeamento entre end. IP e end. *Ethernet* - **impensável e não funciona!**
- 2) ARP:
 - i) alfa difunde: “Quem tem o endereço 193.136.188.3 ?”
 - ii) beta responde com E2 e alfa fica a saber que 193.136.188.3 \Leftrightarrow E2 (193.136.188.4 \Leftrightarrow E1)
 - iii) alfa constrói uma trama *Ethernet* para E2 com o pacote IP (193.136.188.3)

Address Resolution Protocol - ARP (2)

alfa \Rightarrow omega (alfa constrói um pacote com o endereço de destino = 193.136.185.4)

Como E8 não pertence à sub-rede:

- Opção 1** - alfa reconhece que o endereço 193.136.185.4 é remoto e envia para o endereço Ethernet que trata todo o tráfego remoto (*gateway*) - **caso típico**
- Opção 2** - E4 o “router” do DCTI responde (193.136.185.4 \Leftrightarrow E4) e o tráfego para E8 vai para o “router” - **proxy ARP**

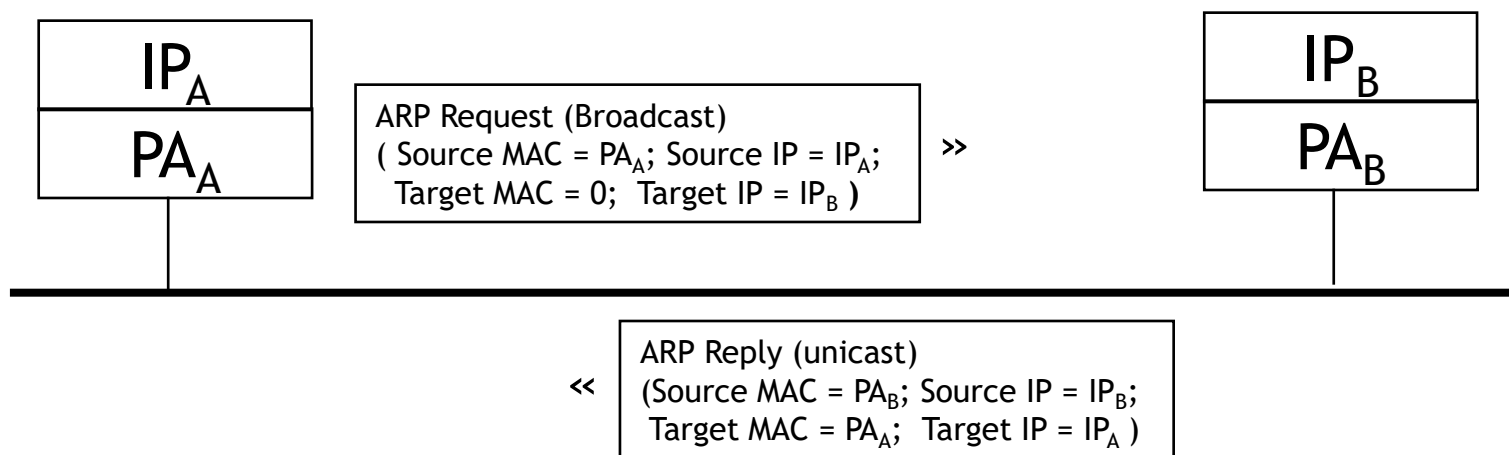
Em ambos os casos:

- alfa constrói uma trama *Ethernet* para E4 com o pacote IP (193.136.185.4)
- O “router” retira o pacote IP da trama *Ethernet* e consulta a sua tabela para 193.136.185.4
- Fica a saber que para a rede 193.136.185.0 tem que ir pelo “router” 193.136.180.3
- Se não conhecer o endereço FDDI de 193.136.180.3 usa ARP para conhecer o endereço F3
- No “router” do CI o pacote IP é retirado da trama FDDI e o software IP verifica que este tem que ser entregue ao 193.136.185.4
- Caso não conheça o endereço *Ethernet* de 193.136.185.4 usa o ARP e fica a conhecer E8.
- Constrói uma trama *Ethernet* para E8

Protocolo ARP

•Exemplo

- Máquina A envia por *broadcast* o ARP Request
- Máquina B envia por *unicast* o ARP Reply
- IP_A e IP_B, pertencem à mesma rede IP
- PA_A e PA_B estão na mesma rede física



PA – physical address

RARP, BOOTP e DHCP

- Obtenção de um endereço IP sendo conhecido apenas o endereço *Ethernet*
- **Exemplo:** “boot” de uma “workstation” sem disco em que a máquina recebe uma imagem do sistema operativo a partir de um servidor remoto
«O meu endereço ethernet é ... alguém conhece o meu endereço IP ?»

- **RARP (*Reverse Address Resolution Protocol*):**

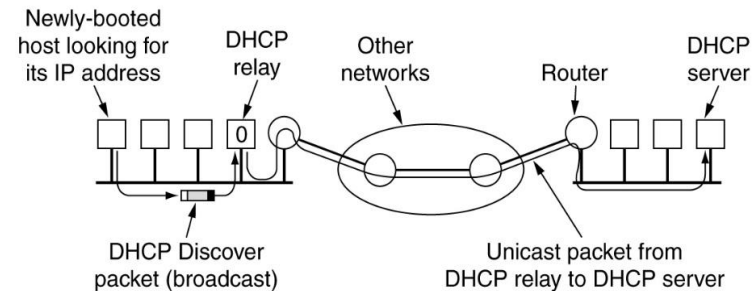
- Utiliza a difusão limitada para aceder ao servidor RARP (não passa pelos *routers*) => É necessário um servidor RARP em cada rede

- **BOOTP (*Bootstrap Protocol*):**

- Utiliza mensagens UDP que são encaminhadas pelos *routers*
- Requer configuração manual das tabelas que mapeiam os endereços IP em endereços *Ethernet*

- **DHCP (*Dynamic Host Configuration Protocol*):**

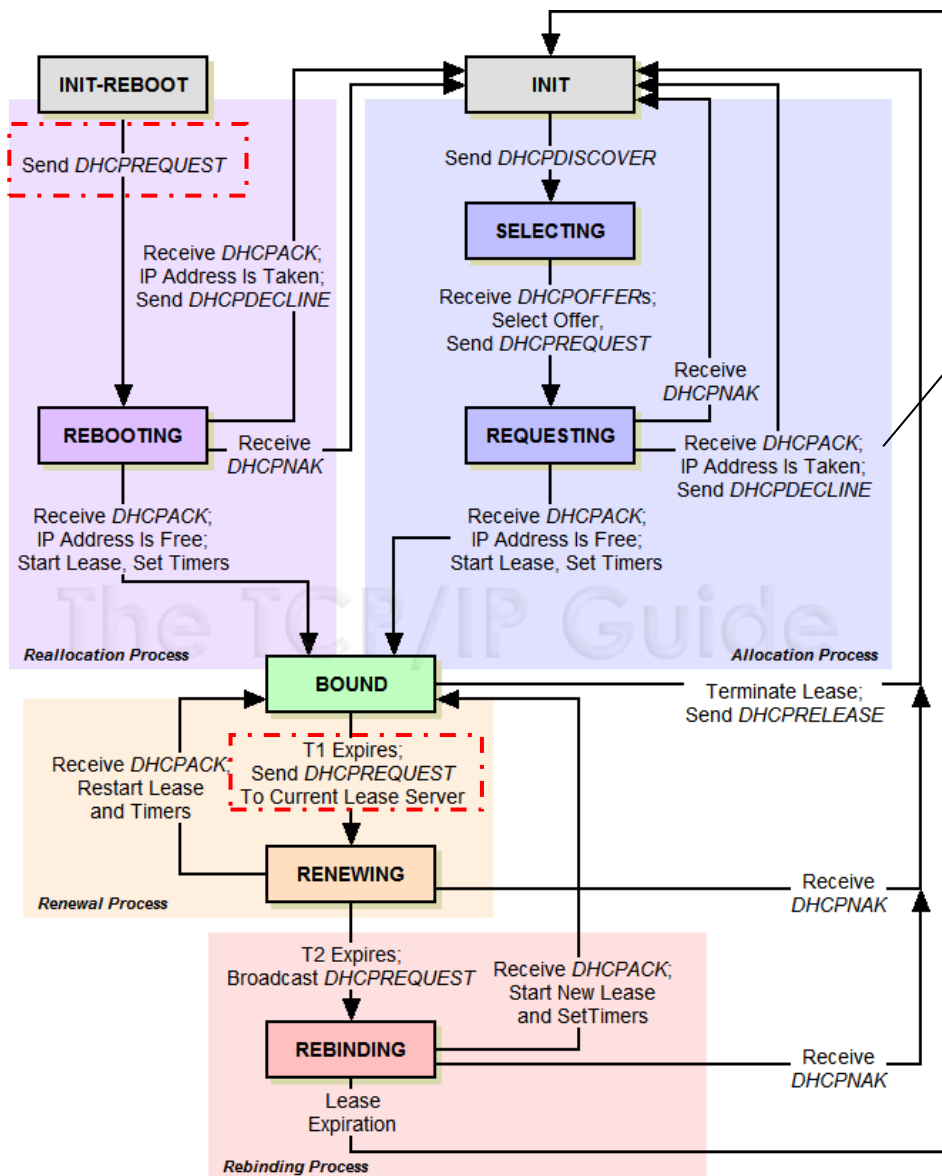
- Evolução do protocolo BOOTP
- Permite configuração automática e manual



DHCP (2)

- Protocolo cliente/servidor de nível aplicação
 - Configuração do cliente
 - e.g., endereço IP, máscara rede, IP *router*, servidor DNS
 - Pode ser utilizado para outros parâmetros que não relacionados com o IP
 - Exemplos: TIME OFFSET, Localização geográfica, Printer server
- Dois tipos de entradas na base de dados do servidor
 - Entradas estáticas (pares cliente/parâmetros fixos)
 - Entradas dinâmicas: conjunto de endereços disponíveis
- Arquitectura do DHCP
 - Pode existir mais de um servidor de DHCP por rede
 - Um servidor DHCP pode servir várias redes (DHCP *relay*)

DHCP: Diagrama de estados/Mensagens

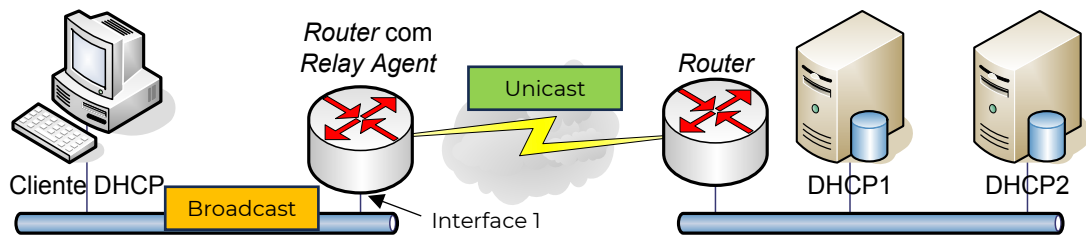


Esta verificação é feita através do envio de trama(s) ARP com o(s) IP(s) oferecido(s)

OpCode	Mensagem	
1	DHCPDISCOVER (C -> S)	Broadcast
2	DHCPOFFER (S -> C)	
3	DHCPREQUEST (C -> S)	
4	DHCPDECLINE (C -> S)	
5	DHCPACK (S -> C)	Unicast
6	DHCPNAK (S -> C)	
7	DHCPRELEASE (C -> S)	

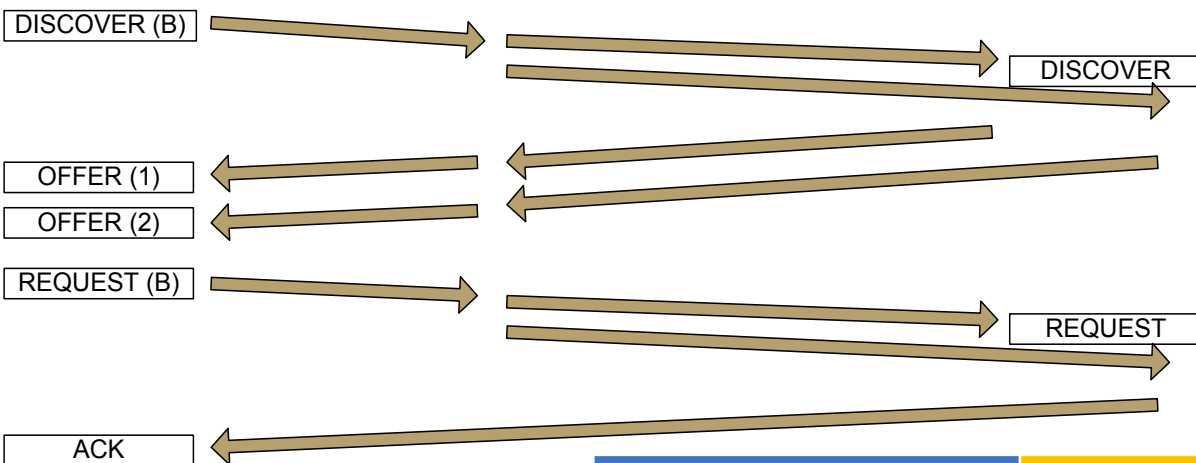
T1 - 50% Lease Time
T2 - 87.5% Lease Time

Obtenção de configuração via Relay Agent



Router com Relay Agent:

- As mensagens DHCP enviadas pelo cliente em Broadcast (e.g. DHCP Discover) são re-enviadas pelo router com Relay Agent em mensagens Unicast dirigidas ao(s) servidor(es) DHCP
- O endereço IP do Relay Agent é o endereço IP do interface do router que está na rede do cliente DHCP
- Como sabe o servidor DHCP em que rede está o cliente ?



Exemplo DHCP Discover ➡

	Broadcast	Unicast
Pacote IP (end. origem)	0.0.0.0	IP interface 1 router
Pacote IP (end. destino)	255.255.255.255	IP server DHCP
Mensagem DHCP (Relay Agent)	0.0.0.0	IP interface 1 router

Exercício 4.11

Uma empresa possui o endereço de rede classe C 194.120.10.0 e necessita de criar 4 sub-redes para:

- o departamento A com 110 máquinas,
- o departamento B com 32 máquinas,
- o departamento C com 20 máquinas e
- o departamento D com 8 máquinas.

Usando o menor número de endereços possível indique, para cada departamento, os endereços de rede, máscara e o número máximo de endereços disponíveis (incluindo rede e difusão) por departamento?

Departamento	Nº máquinas	Endereço de Rede	Máscara	Nº Max. de Endereços
A	110	194.120.10.0	/25	128
B	32	194.120.10.128	/26	64
C	20	194.120.10.192	/27	32
D	8	194.120.10.224	/28	16

Ficam a sobrar 16 endereços por atribuir.

Exercício 4.11 (resolução)

```
Address: 194.120.10.0
Netmask: 255.255.255.0 = 24
Wildcard: 0.0.0.255
=>
Network: 194.120.10.0/24
HostMin: 194.120.10.1
HostMax: 194.120.10.254
Broadcast: 194.120.10.255
Hosts/Net: 254
```

```
11000010.01111000.00001010. 00000000
11111111.11111111.11111111. 00000000
00000000.00000000.00000000. 11111111

11000010.01111000.00001010. 00000000
11000010.01111000.00001010. 00000001
11000010.01111000.00001010. 11111110
11000010.01111000.00001010. 11111111
```

Class C

1. Requested size: 120 hosts

```
Netmask: 255.255.255.128 = 25
Network: 194.120.10.0/25
HostMin: 194.120.10.1
HostMax: 194.120.10.126
Broadcast: 194.120.10.127
Hosts/Net: 126
```

```
11111111.11111111.11111111.1 0000000
11000010.01111000.00001010.0 0000000
11000010.01111000.00001010.0 0000001
11000010.01111000.00001010.0 1111110
11000010.01111000.00001010.0 1111111
```

Class C

2. Requested size: 32 hosts

```
Netmask: 255.255.255.192 = 26
Network: 194.120.10.128/26
HostMin: 194.120.10.129
HostMax: 194.120.10.190
Broadcast: 194.120.10.191
Hosts/Net: 62
```

```
11111111.11111111.11111111.11 000000
11000010.01111000.00001010.10 000000
11000010.01111000.00001010.10 000001
11000010.01111000.00001010.10 111110
11000010.01111000.00001010.10 111111
```

Class C

Needed size: 240 addresses.
Used network: 194.120.10.0/24
Unused: 194.120.10.240/28

3. Requested size: 20 hosts

```
Netmask: 255.255.255.224 = 27
Network: 194.120.10.192/27
HostMin: 194.120.10.193
HostMax: 194.120.10.222
Broadcast: 194.120.10.223
Hosts/Net: 30
```

```
11111111.11111111.11111111.111 00000
11000010.01111000.00001010.110 00000
11000010.01111000.00001010.110 00001
11000010.01111000.00001010.110 11110
11000010.01111000.00001010.110 11111
```

Class C

4. Requested size: 8 hosts

```
Netmask: 255.255.255.240 = 28
Network: 194.120.10.224/28
HostMin: 194.120.10.225
HostMax: 194.120.10.238
Broadcast: 194.120.10.239
Hosts/Net: 14
```

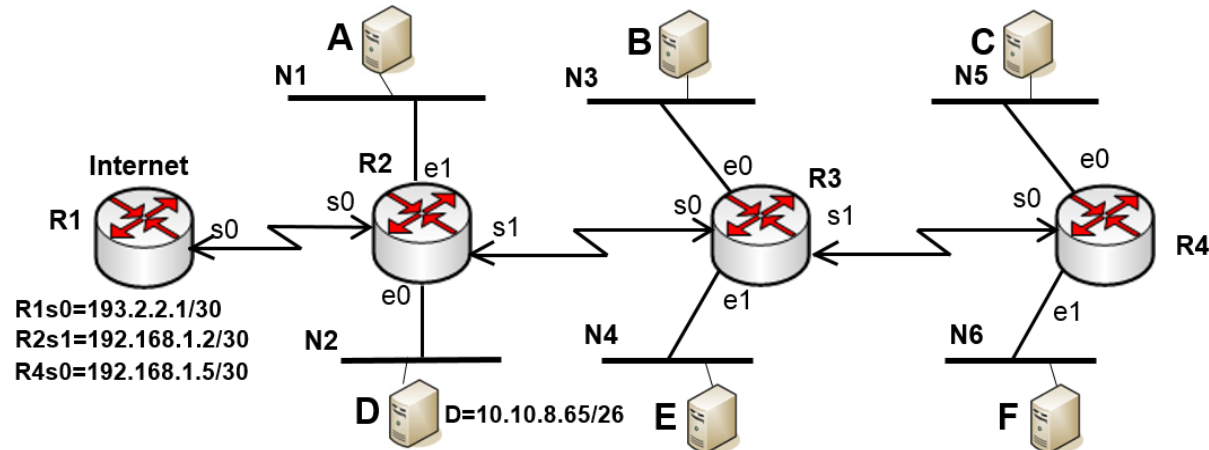
```
11111111.11111111.11111111.1111 0000
11000010.01111000.00001010.1110 0000
11000010.01111000.00001010.1110 0001
11000010.01111000.00001010.1110 1110
11000010.01111000.00001010.1110 1111
```

Class C

Exercício 4.12

Considere a seguinte rede da figura. A gama de endereços da rede 10.10.8.0/23 encontra-se distribuída pelas sub-redes N1 a N6. Os endereços das ligações série deverão estar de acordo com os valores indicados assim como com o endereço IP do PC “D”. Os PCs A a F estão configurados nas redes a que estão ligados incluindo o *default gateway* (router, na figura, de ligação à Internet).

a) Preencha a tabela dos endereços IP das interfaces série indicadas dos *routers*, nas redes definidas:

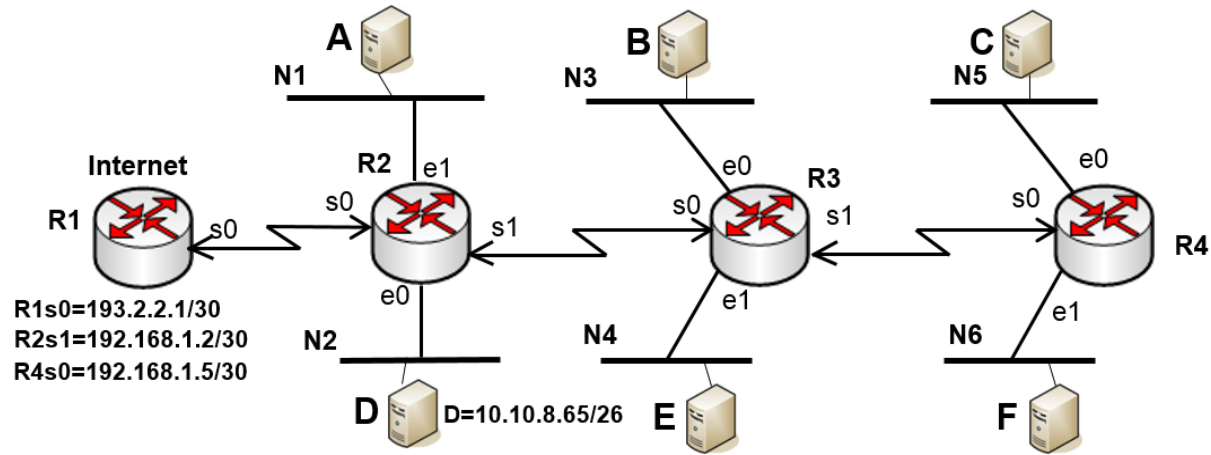


a)

Interface	IP address	CIDR	(Network)
R1s0	193.2.2.1	/30	193.2.2.0
R2s0			
R2s1	192.168.1.2	/30	192.168.1.0
R3s0			
R3s1			
R4s0	192.168.1.5	/30	192.168.1.4

Exercício 4.12- Considere a seguinte rede da figura. A gama de endereços da rede 10.10.8.0/23 encontra-se distribuída pelas sub-redes N1 a N6. Os endereços das ligações série deverão estar de acordo com os valores indicados assim como com o endereço IP do PC “D”. Os PCs A a F estão configurados nas redes a que estão ligados incluindo o *default gateway* (router, na figura, de ligação à Internet).

a) Preencha a tabela dos endereços IP das interfaces série indicadas dos *routers*, nas redes definidas:

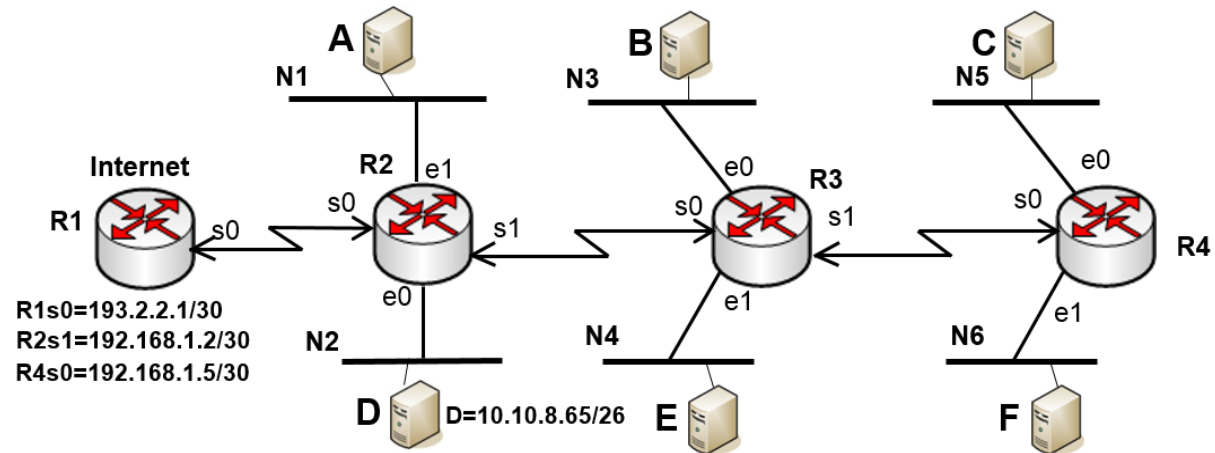


a)

Interface	IP address	CIDR	(Network)
R1s0	193.2.2.1	/30	193.2.2.0
R2s0	193.2.2.2	/30	193.2.2.0
R2s1	192.168.1.2	/30	192.168.1.0
R3s0	192.168.1.1	/30	192.168.1.0
R3s1	192.168.1.6	/30	192.168.1.4
R4s0	192.168.1.5	/30	192.168.1.4

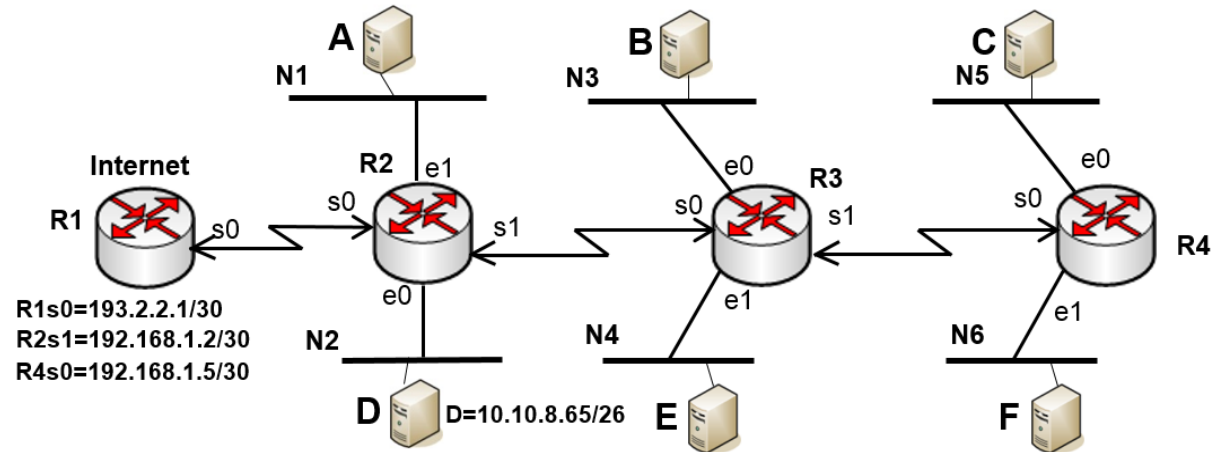
Exercício 4.12- Considere a seguinte rede da figura. A gama de endereços da rede 10.10.8.0/23 encontra-se distribuída pelas sub-redes N1 a N6. Os endereços das ligações série deverão estar de acordo com os valores indicados assim como com o endereço IP do PC “D”. Os PCs A a F estão configurados nas redes a que estão ligados incluindo o *default gateway* (router, na figura, de ligação à Internet).

b) Preencha a tabela dos endereços de rede redistribuindo o endereço de rede global 10.10.8.0/23 pelas redes N1 a N6, sem desperdiçar endereços, que permita minimizar as tabelas de encaminhamento dos routers. As redes deverão satisfazer os endereços IP indicados na figura.



b)

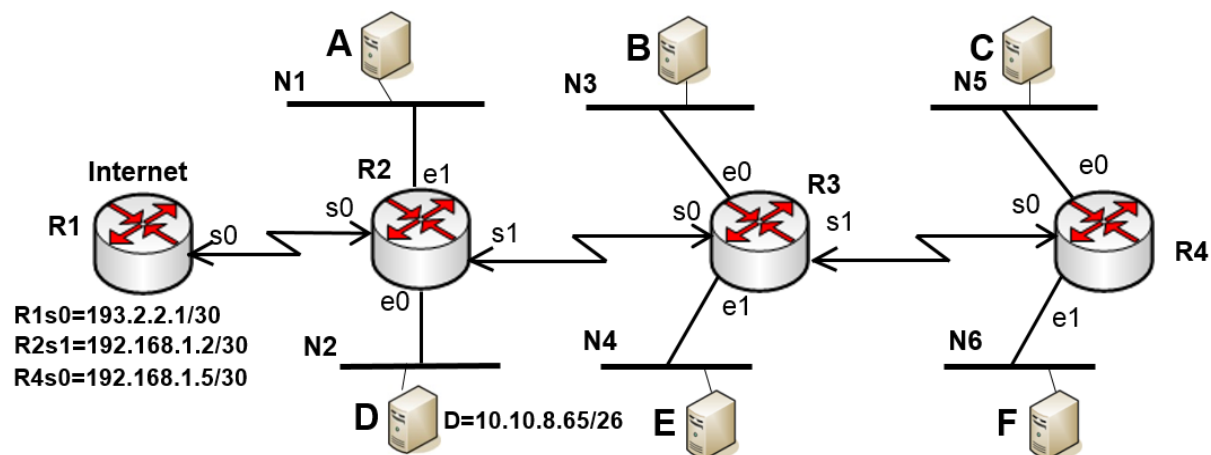
Exercício 4.12- Considere a seguinte rede da figura. A gama de endereços da rede 10.10.8.0/23 encontra-se distribuída pelas sub-redes N1 a N6. Os endereços das ligações série deverão estar de acordo os valores indicados assim com o endereço IP do PC “D”. Os PCs A a F estão configurados nas redes a que estão ligados incluindo o *default gateway* (router, na figura, de ligação à Internet).



b)

Network	Network address	CIDR	Broadcast address	Max. interfaces
Rede N2	10.10.8.64	/26	10.10.8.127	62

Exercício 4.12- Considere a seguinte rede da figura. A gama de endereços da rede 10.10.8.0/23 encontra-se distribuída pelas sub-redes N1 a N6. Os endereços das ligações série deverão estar de acordo com os valores indicados assim como com o endereço IP do PC “D”. Os PCs A a F estão configurados nas redes a que estão ligados incluindo o *default gateway* (router, na figura, de ligação à Internet).

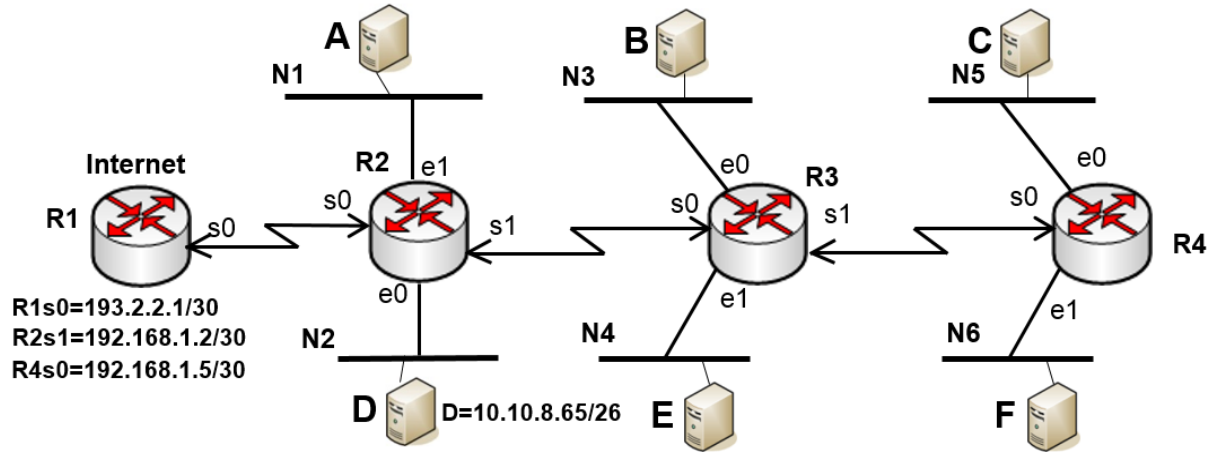


b) Preencha a tabela dos endereços de rede redistribuindo o endereço de rede global 10.10.8.0/23 pelas redes N1 a N6, sem desperdiçar endereços, que permita minimizar as tabelas de encaminhamento dos routers. As redes deverão satisfazer os endereços IP indicados na figura.

b)

Network	Network address	CIDR	Broadcast address	Max. interfaces
Rede N1				
Rede N2	10.10.8.64	/26	10.10.8.127	62
Rede N3				
Rede N4				
Rede N5				
Rede N6				
Série R1-R2	193.2.2.0	/30	193.2.2.3	2
Série R2-R3	192.168.1.0	/30	192.168.1.3	2
Série R3-R4	192.168.1.4	/30	192.168.1.7	2

Exercício 4.12- Considere a seguinte rede da figura. A gama de endereços da rede 10.10.8.0/23 encontra-se distribuída pelas sub-redes N1 a N6. Os endereços das ligações série deverão estar de acordo com os valores indicados assim como com o endereço IP do PC “D”. Os PCs A a F estão configurados nas redes a que estão ligados incluindo o *default gateway* (router, na figura, de ligação à Internet).

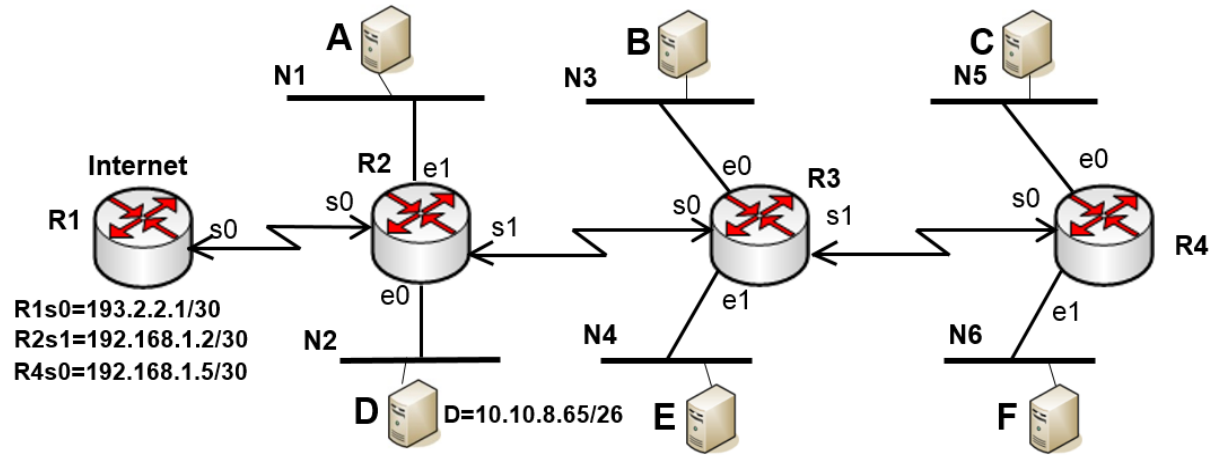


b) Preencha a tabela dos endereços de rede redistribuindo o endereço de rede global 10.10.8.0/23 pelas redes N1 a N6, sem desperdiçar endereços, que permita minimizar as tabelas de encaminhamento dos routers. As redes deverão satisfazer os endereços IP indicados na figura.

Network	Network address	CIDR	Broadcast address	Max. interfaces
Rede N1	10.10.8.0	/26	10.10.8.63	62
Rede N2	10.10.8.64	/26	10.10.8.127	62
Rede N3	10.10.8.128	/26	10.10.8.191	62
Rede N4	10.10.8.192	/26	10.10.8.255	62
Rede N5	10.10.9.0	/25	10.10.9.127	126
Rede N6	10.10.9.128	/25	10.10.9.255	126
Série R1-R2	193.2.2.0	/30	193.2.2.3	2
Série R2-R3	192.168.1.0	/30	192.168.1.3	2
Série R3-R4	192.168.1.4	/30	192.168.1.7	2

Exercício 4.12- Considere a seguinte rede da figura. A gama de endereços da rede 10.10.8.0/23 encontra-se distribuída pelas sub-redes N1 a N6. Os endereços das ligações série deverão estar de acordo com os valores indicados assim como com o endereço IP do PC “D”. Os PCs A a F estão configurados nas redes a que estão ligados incluindo o *default gateway* (router, na figura, de ligação à Internet).

c) Atribua endereços IP e respectivas máscaras e indique o *default gateway* das máquinas das redes indicadas, para estarem de acordo com os endereços atribuídos na alínea b) *Atribua os endereços mais baixos aos PCs e os mais elevados aos routers.*



c)	Interface	IP address	CIDR	(Network)
	R1s0	193.2.2.1	/30	193.2.2.0
	R2s0	193.2.2.2	/30	193.2.2.0
	R2s1	192.168.1.2	/30	192.168.1.0
	R3s0	192.168.1.1	/30	192.168.1.0
	R3s1	192.168.1.6	/30	192.168.1.4
	R4s0	192.168.1.5	/30	192.168.1.4

	Interface	IP address	CIDR	(Network)
	R2e0	10.10.8.126	/26	10.10.8.64
	R2e1	10.10.8.62	/26	10.10.8.0
	R3e0	10.10.8.190	/26	10.10.8.128
	R3e1	10.10.8.254	/26	10.10.8.192
	R4e0	10.10.9.126	/25	10.10.9.0
	R4e1	10.10.9.254	/25	10.10.9.128

Exercício 4.12- Considere a seguinte rede da figura. A gama de endereços da rede 10.10.8.0/23 encontra-se distribuída pelas sub-redes N1 a N6. Os endereços das ligações série deverão estar de acordo com os valores indicados assim como com o endereço IP do PC “D”. Os PCs A a F estão configurados nas redes a que estão ligados incluindo o *default gateway* (router, na figura, de ligação à Internet).

GLOBAL

Settings

Algorithm Settings

ROUTING

Static

RIP

SWITCHING

VLAN Database

INTERFACE

GigabitEthernet0/0

GigabitEthernet0/1

GigabitEthernet0/2

Serial0/0/0

Serial0/0/1

R2

Physical Config CLI Attributes

GigabitEthernet0/0

Port Status ☒ On

Bandwidth ☒ 1000 Mbps ☐ 100 Mbps ☐ 10 Mbps ☒ Auto

Duplex ☐ Half Duplex ☒ Full Duplex ☒ Auto

MAC Address 00D0.BC28.50EA

IP Configuration

IP Address 10.10.8.126

Subnet Mask 255.255.255.192

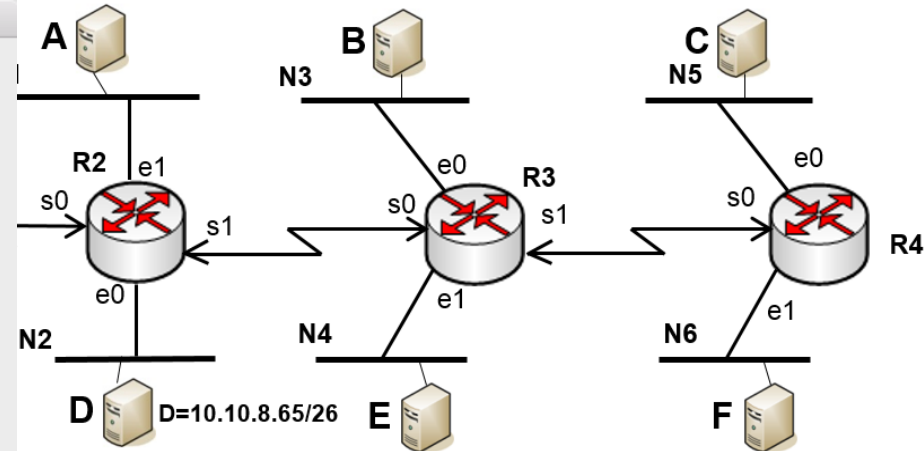
Tx Ring Limit 10

Equivalent IOS Commands

```

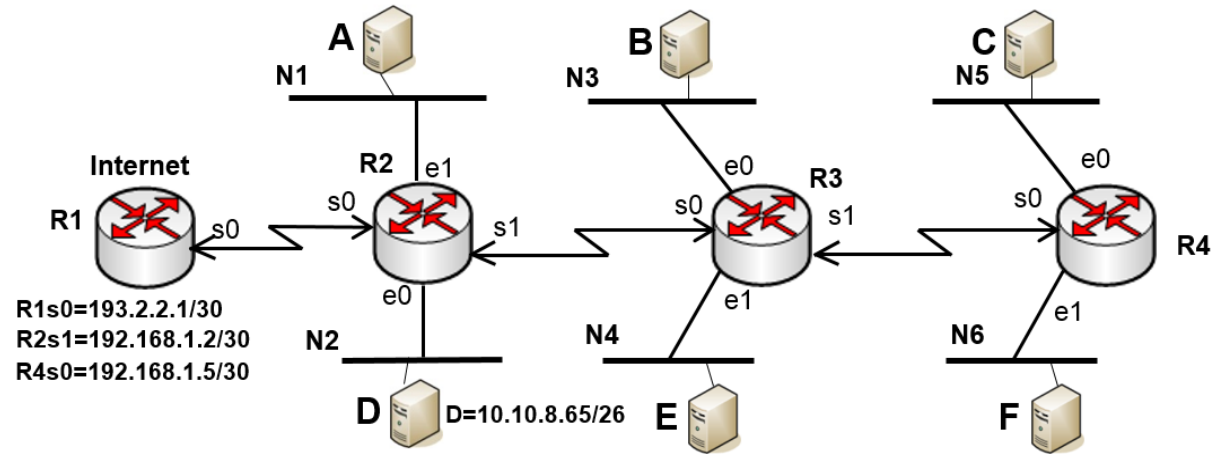
Router(config-if)#ip address 10.10.8.190 255.255.255.192
Router(config-if)#ip address 10.10.8.190 255.255.255.192
Router(config-if)#ip address 10.10.8.126 255.255.255.192
Router(config-if)#

```



Interface	IP address	CIDR	Mask
R2e0	10.10.8.126	/26	255.255.255.192

Exercício 4.12- Considere a seguinte rede da figura. A gama de endereços da rede 10.10.8.0/23 encontra-se distribuída pelas sub-redes N1 a N6. Os endereços das ligações série deverão estar de acordo com os valores indicados assim como com o endereço IP do PC “D”. Os PCs A a F estão configurados nas redes a que estão ligados incluindo o *default gateway* (router, na figura, de ligação à Internet).

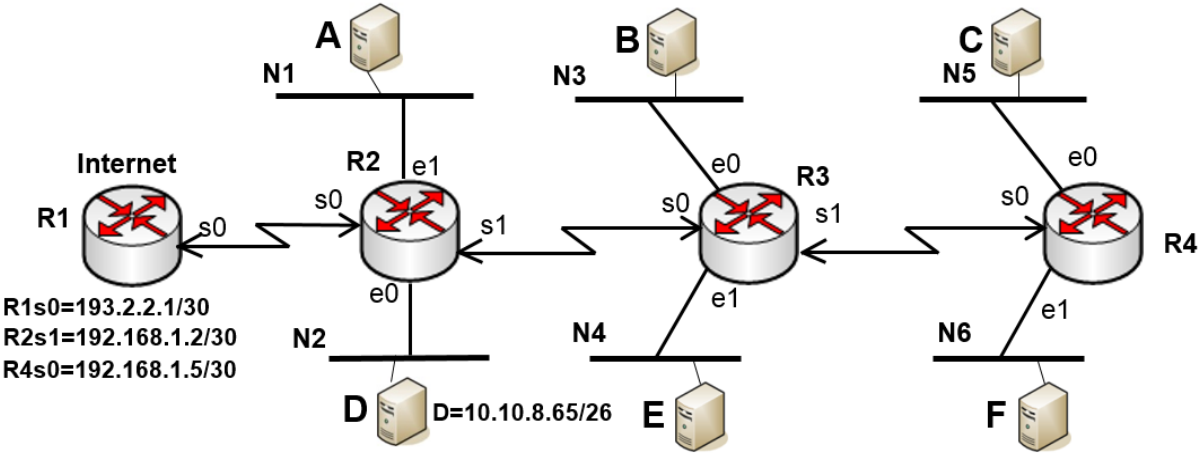


c)

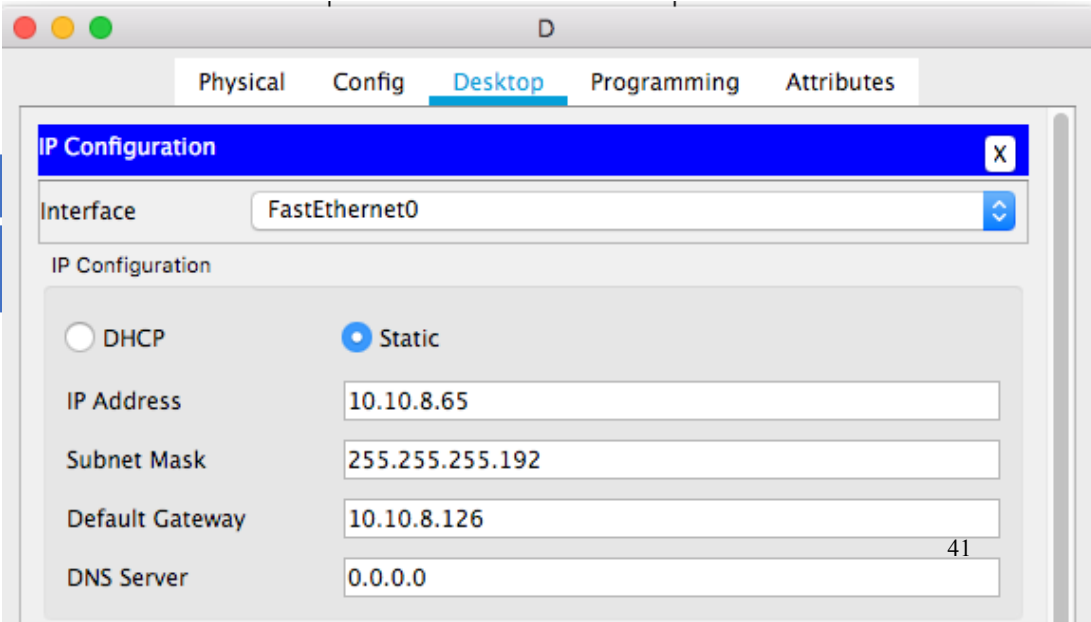
Host	IP address	CIDR	Gateway	(Network)
A	10.10.8.1	/26	10.10.8.62	10.10.8.0
B	10.10.8.129	/26	10.10.8.190	10.10.8.128
C	10.10.9.1	/25	10.10.9.126	10.10.9.0
D	10.10.8.65	/26	10.10.8.126	10.10.8.64
E	10.10.8.193	/26	10.10.8.254	10.10.8.192
F	10.10.9.129	/25	10.10.9.254	10.10.9.128

Exercício 4.12- Considere a seguinte rede da figura. A gama de endereços da rede 10.10.8.0/23 encontra-se distribuída pelas sub-redes N1 a N6. Os endereços das ligações série deverão estar de acordo com os valores indicados assim como com o endereço IP do PC “D”. Os PCs A a F estão configurados nas redes a que estão ligados incluindo o *default gateway* (router, na figura, de ligação à Internet).

c) Atribua endereços IP e respectivas máscaras e indique o *default gateway* das máquinas das redes indicadas, para estarem de acordo com os endereços atribuídos na alínea b) *Atribua os endereços mais baixos aos PCs e os mais elevados aos routers.*



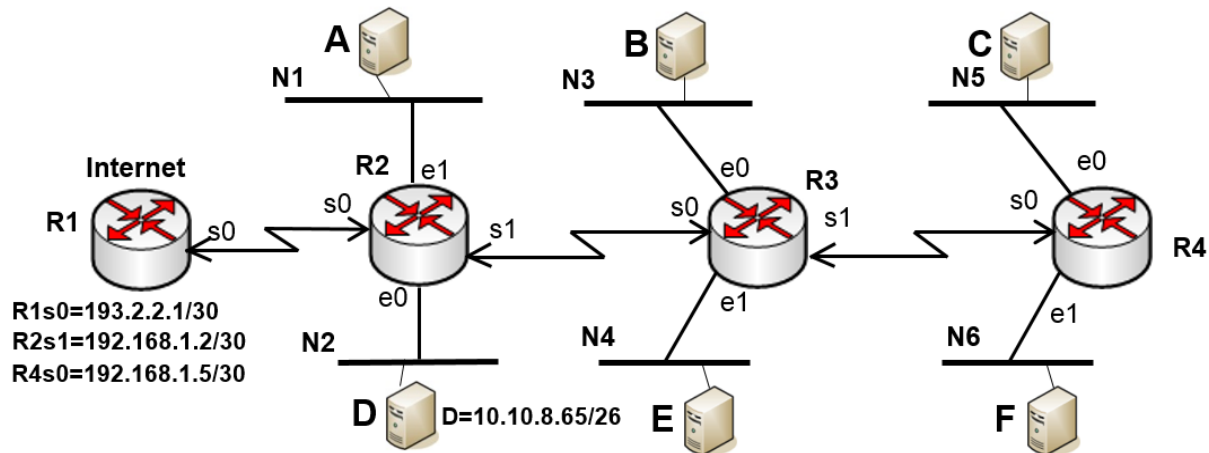
Host	IP address	CIDR	Gateway
D	10.10.8.65	/26	10.10.8.126



Exercício 4.12- Considere a seguinte rede da figura. A gama de endereços da rede 10.10.8.0/23 encontra-se distribuída pelas sub-redes N1 a N6. Os endereços das ligações série deverão estar de acordo com os valores indicados assim como com o endereço IP do PC “D”. Os PCs A a F estão configurados nas redes a que estão ligados incluindo o *default gateway* (router, na figura, de ligação à Internet).

d) Indique a tabela de encaminhamento completa do router **R2** indicando a rota de *default gateway*.

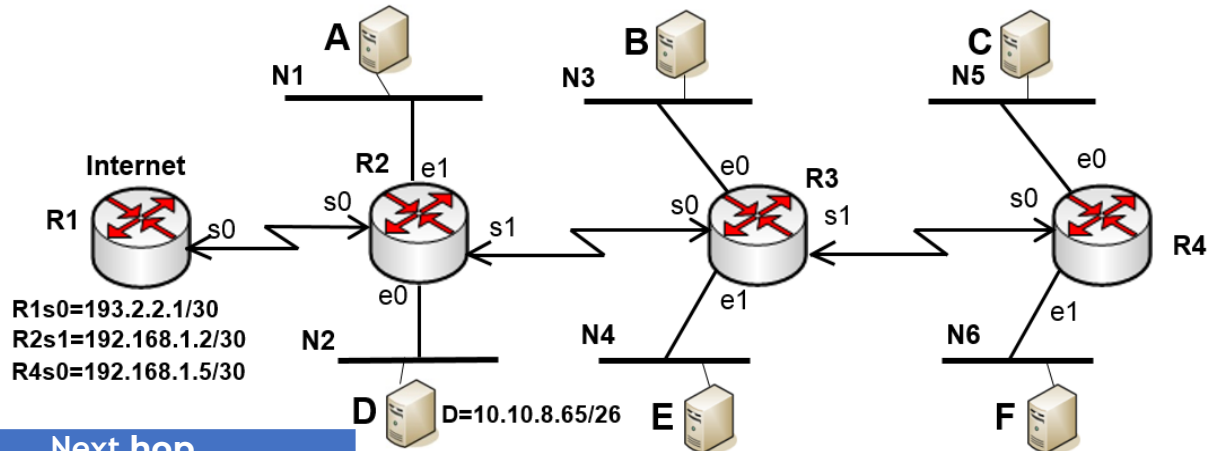
d)



Network Address	CIDR	Next Hop	(Type)	(Interface)
10.10.8.0 (N1)	/26	---	Connected	R2e1
10.10.8.64 (N2)	/26	---	Connected	R2e0
10.10.8.128 (N3)	/26	192.168.1.1 (R3s0)	Static	R2s1
10.10.8.192 (N4)	/26	192.168.1.1 (R3s0)	Static	R2s1
10.10.9.0 (N5)	/25	192.168.1.1 (R3s0)	Static	R2s1
10.10.9.128 (N6)	/25	192.168.1.1 (R3s0)	Static	R2s1
193.2.2.0 (R1-R2)	/30	---	Connected	R2s0
192.168.1.0 (R2-R3)	/30	---	Connected	R2s1
192.168.1.4 (R3-R4)	/30	192.168.1.1(R3s0)	Static	R2s1
0.0.0.0 (default)	/0	193.2.2.1 (R1s0)	Static	R2s0

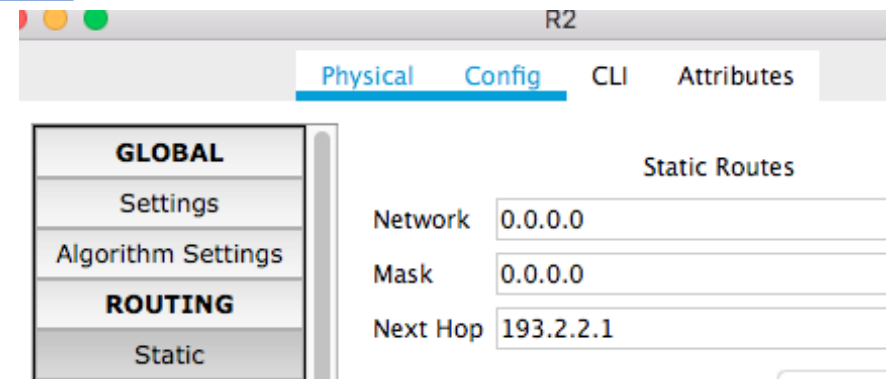
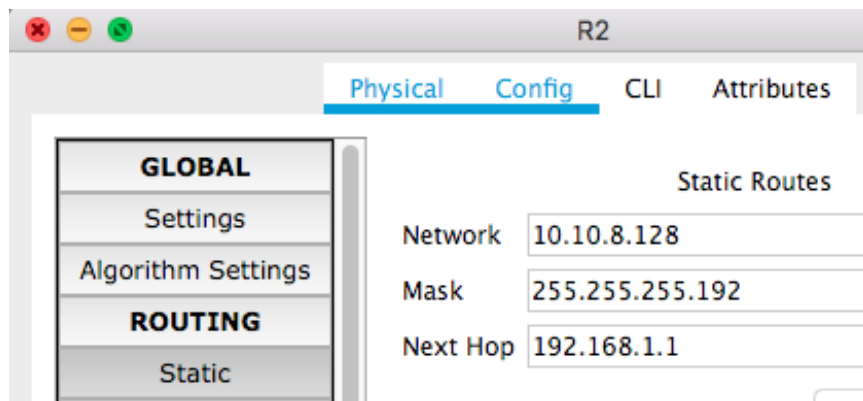
Exercício 4.12- Considere a seguinte rede da figura. A gama de endereços da rede 10.10.8.0/23 encontra-se distribuída pelas sub-redes N1 a N6. Os endereços das ligações série deverão estar de acordo com os valores indicados assim como com o endereço IP do PC “D”. Os PCs A a F estão configurados nas redes a que estão ligados incluindo o *default gateway* (router, na figura, de ligação à Internet).

d) Indique a tabela de encaminhamento completa do router **R2** indicando a rota de *default gateway*.



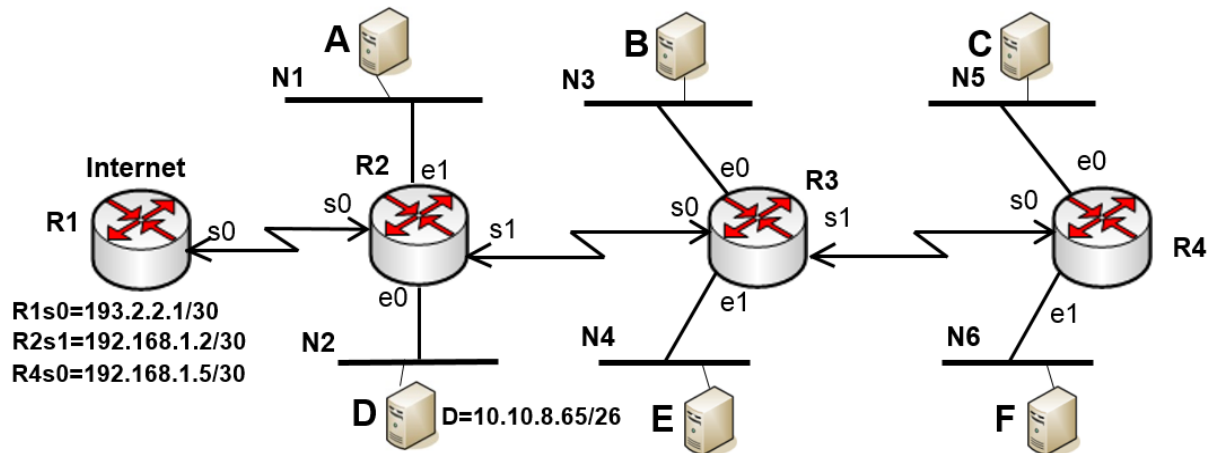
d)

Network	CIDR	Next hop
10.10.8.128 (N3)	/26	192.168.1.1 (R3s0)
0.0.0.0 (default)	/0	193.2.2.1 (R1s0)



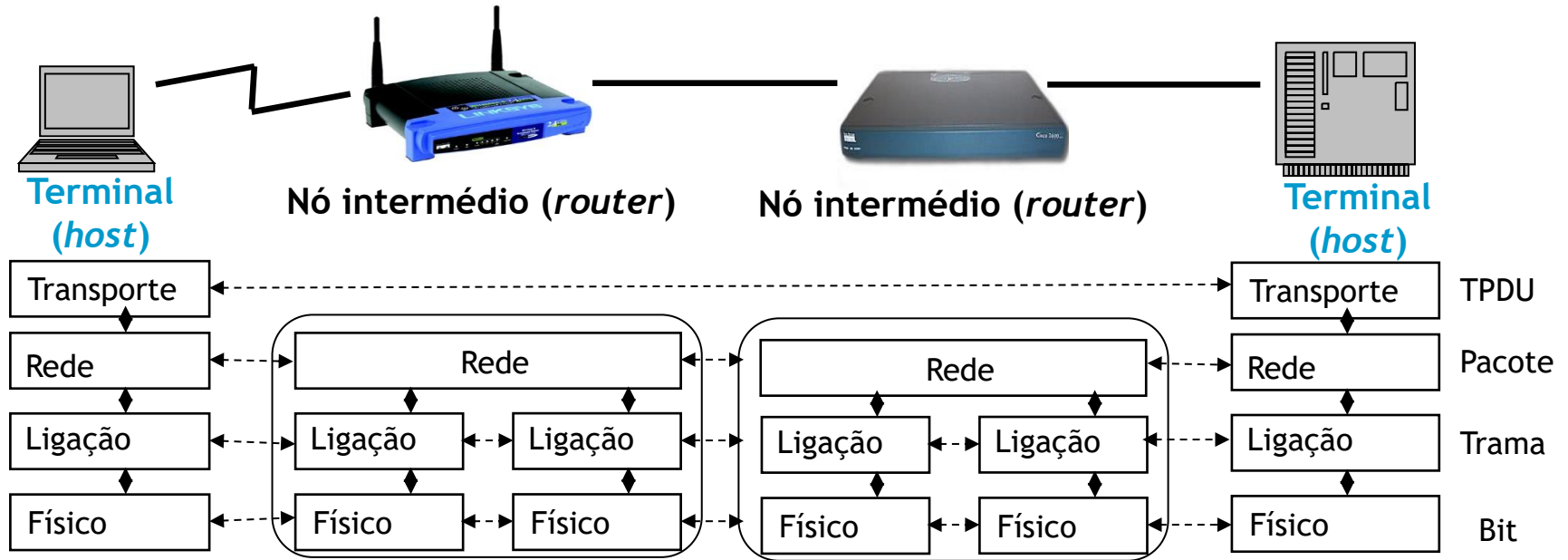
Exercício 4.12- Considere a seguinte rede da figura. A gama de endereços da rede 10.10.8.0/23 encontra-se distribuída pelas sub-redes N1 a N6. Os endereços das ligações série deverão estar de acordo com os valores indicados assim como com o endereço IP do PC “D”. Os PCs A a F estão configurados nas redes a que estão ligados incluindo o *default gateway* (router, na figura, de ligação à Internet).

e) Indique tabela de encaminhamento do router R2 devidamente resumizada.



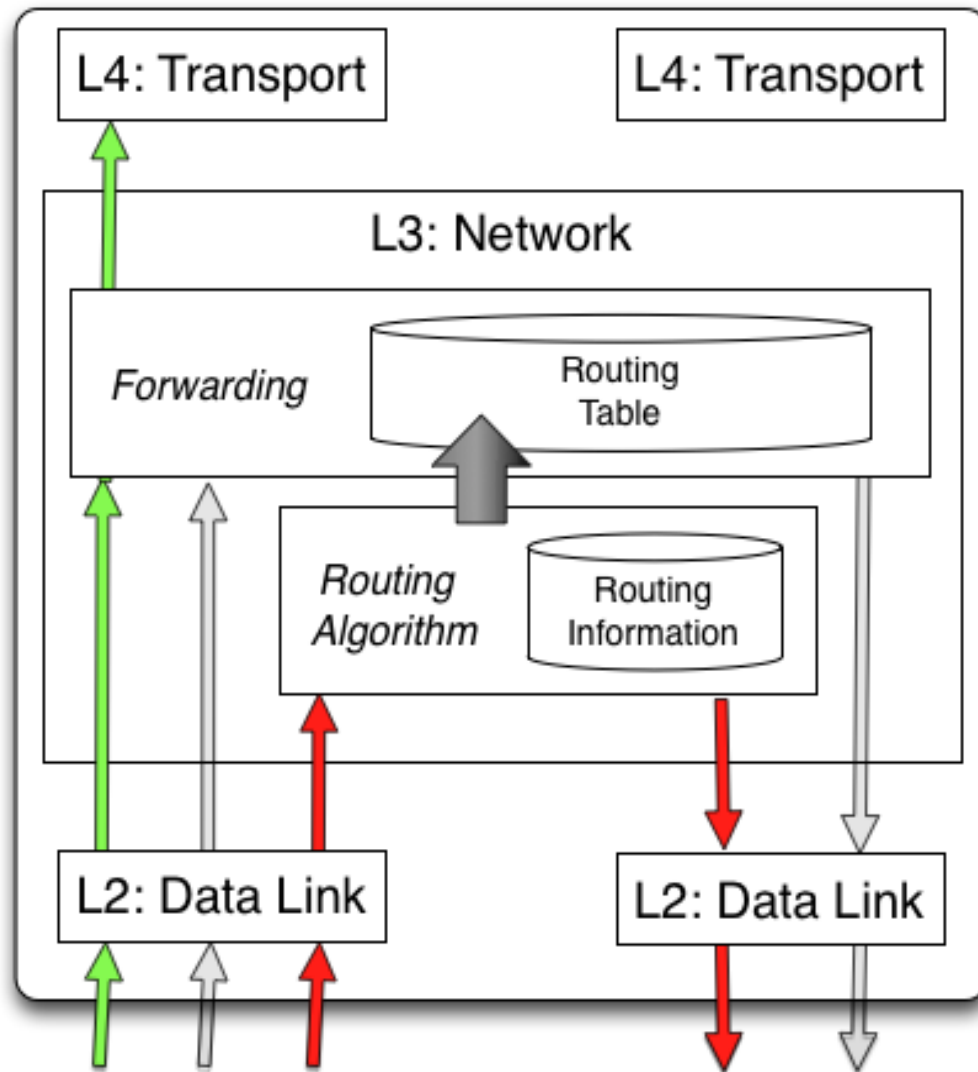
Network Address	CIDR	Next Hop	(Type)	(Interface)
10.10.8.0 (N1)	/26	---	Connected	R2e1
10.10.8.64 (N2)	/26	---	Connected	R2e0
10.10.8.128 (N3 e N4)	/25	192.168.1.1 (R3s0)	Static	R2s1
10.10.9.0 (N5 e N6)	/24	192.168.1.1 (R3s0)	Static	R2s1
193.2.2.0 (R1-R2)	/30	---	Connected	R2s0
192.168.1.0 (R2-R3)	/30	---	Connected	R2s1
192.168.1.4 (R3-R4)	/30	192.168.1.1(R3s0)	Static	R2s1
0.0.0.0 (default)	/0	193.2.2.1 (R1s0)	Static	R2s0

Recordando Nível de Rede (*Network Layer*)



- Utiliza os serviços do nível L2 (*Data Link*), presta serviços ao nível L4 (*Transport*).
Os serviços prestados ao nível transporte devem ser independentes da tecnologia da sub-rede.
O número, tipo e topologia das sub-redes que constituem o nível rede deve ser escondido ao nível transporte.
Todos os nós têm de ter pelo menos os níveis L1-L2-L3 implementados
- A unidade protocolar fundamental é o pacote (*packet*).
- O nível de rede preocupa-se com o percurso dos pacotes desde a origem através da rede até ao destino (*end-to-end*)
O percurso pode ser composto por vários nós intermédios - encaminhadores (*routers*).

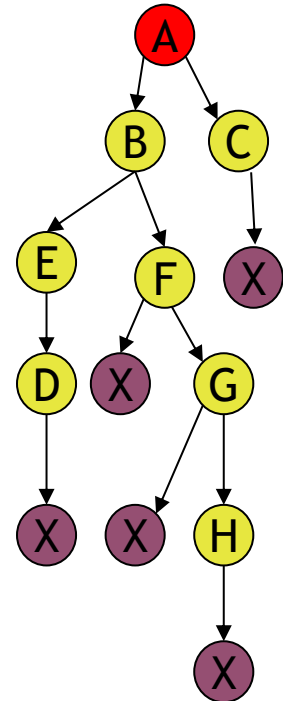
Recordando: Funcionalidades fundamentais do Nível de Rede



- **Expedição (*forwarding*)**: envio dos pacotes para o próximo nó:
 - Por inundação
 - Por consulta de uma tabela de encaminhamento (*routing table*).
(Atenção à diferença entre os processos de expedição e de routing)
- **Encaminhamento (*routing*)**: construção da tabela de encaminhamento
 - Necessita de conhecer a topologia da sub-rede (os nós) e escolher o caminho apropriado através dos nós.
 - Diferentes estratégias/algoritmos para recolher informação sobre a topologia da rede e para o cálculo do caminho através da rede.

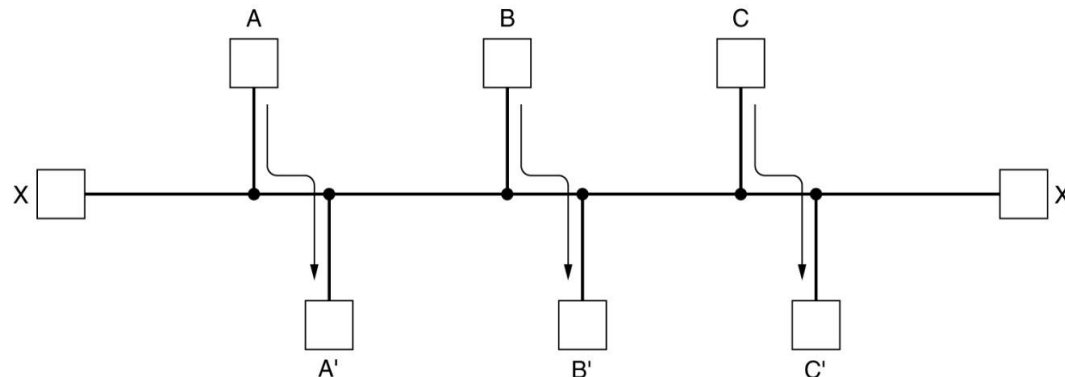
Expedição por Inundação (*Flooding*)

- Cada pacote é enviado para todas as saídas menos para aquela de onde veio (gera n cópias).
- Convém limitar o número de cópias (ilimitada no caso de ciclos na rede), várias opções:
 - Cada nó mantém uma lista, para cada nó de origem, com o número de sequência dos pacotes enviados por esse nó, um pacote que pertença à lista não é encaminhado de novo.
 - Cada pacote mantém uma lista de todos os nós que já visitou.
 - *Contador (Time-to-live, TTL) no cabeçalho que é decrementado em cada salto. Idealmente o contador deverá ser inicializado com o número de saltos para ir da origem ao destino ou, no pior caso, com o tamanho máximo da rede.*
- **Variante Inundação selectiva:** cada pacote apenas é enviado para as saídas com direcção ao destino (*aprendido*).
- A inundação ao utilizar todos os caminhos usa também e sempre o caminho mais curto.
- Muito interessantes para: aplicações no domínio militar (robustez) e em aplicações com bases de dados distribuídas que necessitam de ser actualizadas (necessidade de enviar para todos).



Algoritmos de Encaminhamento

- O nível de rede tem que escolher o caminho entre a origem e o destino podendo envolver múltiplos saltos - **Encaminhamento dos Pacotes**:
- Os algoritmos para geração das tabelas de encaminhamento podem ser:
 - Não Adaptativos (encaminhamento estático).
 - Tabelas construídas manualmente.
 - Não escalável, muito sujeito a erros.
 - Adaptativos (encaminhamento dinâmico).
 - Tabelas construídas automaticamente com informação trocada entre os nós.
 - O algoritmo para a construção das tabelas deverá ser: **correcto, simples, robusto, estável, justo e óptimo**.
Os critérios de optimalidade podem incluir: minimizar o tempo de atraso, minimizar o número de saltos, minimizar o custo financeiro, e maximizar o desempenho da rede.

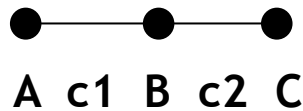


Princípio de Optimização (Bellman, 1957)

- Se o nó B pertence ao caminho óptimo do nó A para o nó C então o caminho óptimo de B para C é o mesmo quer seja de B para C quer de A para C.

Prova por absurdo:

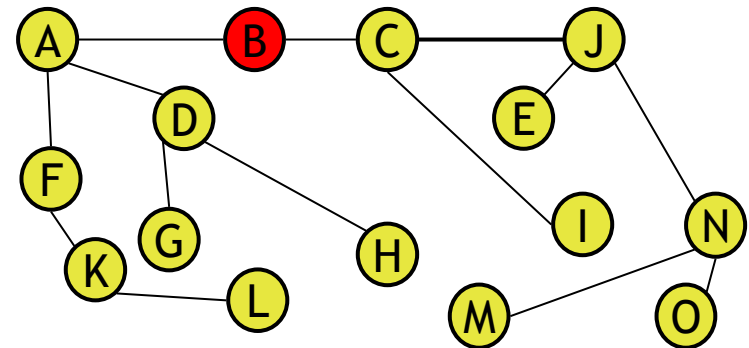
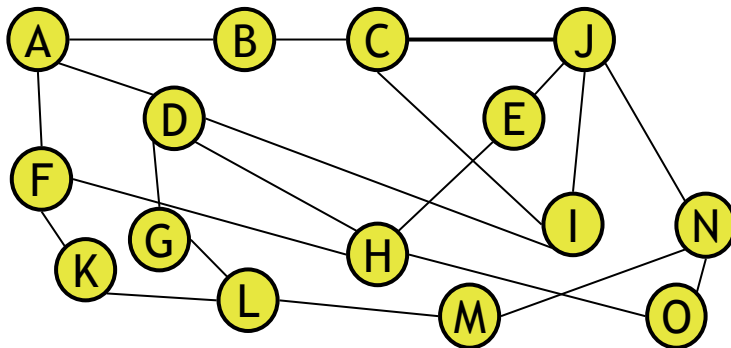
Se existir caminho melhor entre B e C que c_2 , ele seria usado conjuntamente com c_1 para melhorar o desempenho de A para C logo c_1 e c_2 não seriam óptimos



Corolário:

O cálculo do caminho mais curto pode ser feito de forma incremental, dinâmica.

- O conjunto de caminhos óptimos de todas as origens para um dado destino designa-se por árvore de escoamento “*sink tree*” (não é necessariamente única).



Algoritmo do Caminho Mais Curto de Dijkstra (1959) (1)

Algoritmo para calcular o melhor caminho para um destino a partir de qualquer nó.

Grafo (orientado): $G=(V, E)$, V nós da rede, E arcos etiquetados com o “custo”, w , entre nós.

Condição: Para todos os arcos (u,v) , $w(u,v) \geq 0$ (i.e., todos os custos são não-negativos).

DISJKSTRA (G, w, s)

/* Inicialização */

$S \leftarrow \{\}$

$Q \leftarrow V[G]$

$d_Q \setminus \{s\} \leftarrow \infty$

$d_s = 0$

while ($Q \neq \{\}$)

$u \leftarrow \min_d(Q)$

$Q \leftarrow Q \setminus \{u\}$

$S \leftarrow S \cup \{u\}$

/* Atualização dos vizinhos*/

for each $v \in \text{Adj}[u]$

if ($d_v > d_u + w(u, v)$)

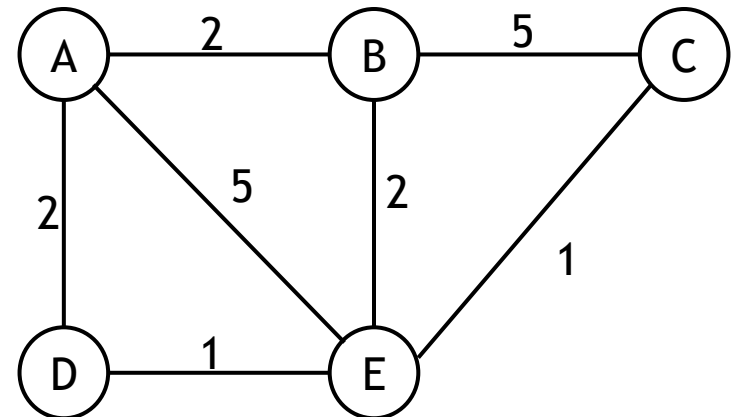
$d_v = d_u + w(u, v)$

s ; nó destino

S ; nós “resolvidos”

Q ; nós “por resolver” ordenados por d_u

d_u ; distância “actual” de u para s



Algoritmo do Caminho Mais Curto de Dijkstra (2)

DISJKSTRA ($G, w, s=A$)

$S \leftarrow \{\}$

$Q \leftarrow V[G]$

$d_Q \setminus \{s\} \leftarrow \infty$

$d_s = 0$

while ($Q \neq \{\}$)

$u \leftarrow \min_d(Q)$

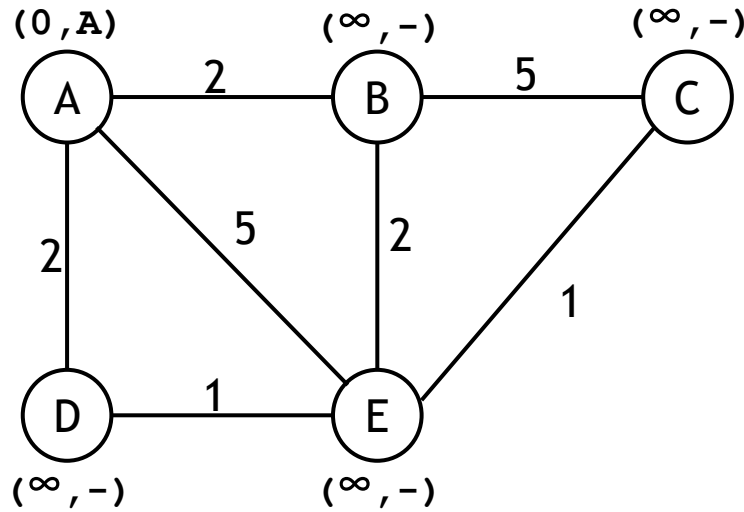
$Q \leftarrow Q \setminus \{u\}$

$S \leftarrow S \cup \{u\}$

 for each $v \in \text{Adj}_Q[u]$

 if ($d_v > d_u + w(u, v)$)

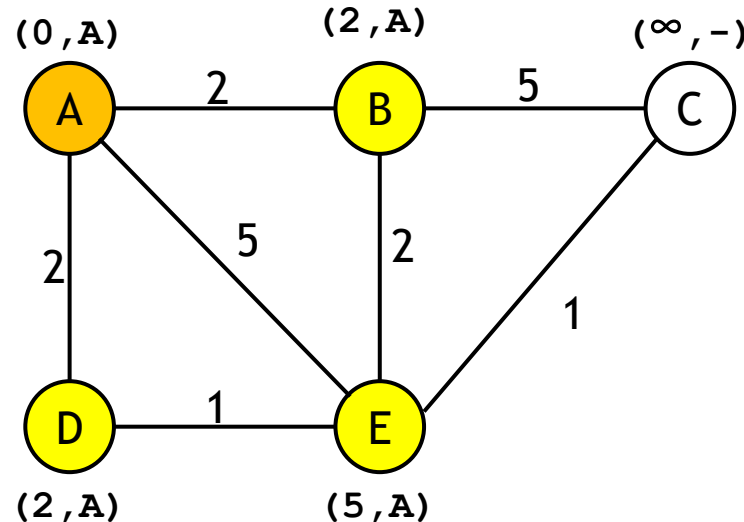
$d_v = d_u + w(u, v)$



$s = A$

$S \leftarrow \{\}$

$Q \leftarrow \{A, B, C, D, E\}$



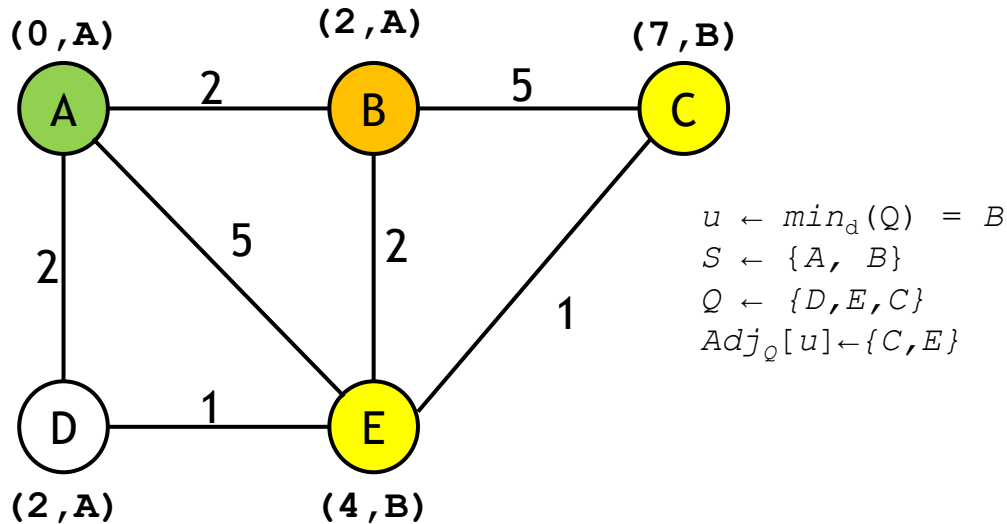
$u \leftarrow \min_d(Q) = A$

$S \leftarrow \{A\}$

$Q \leftarrow \{B, C, D, E\}$

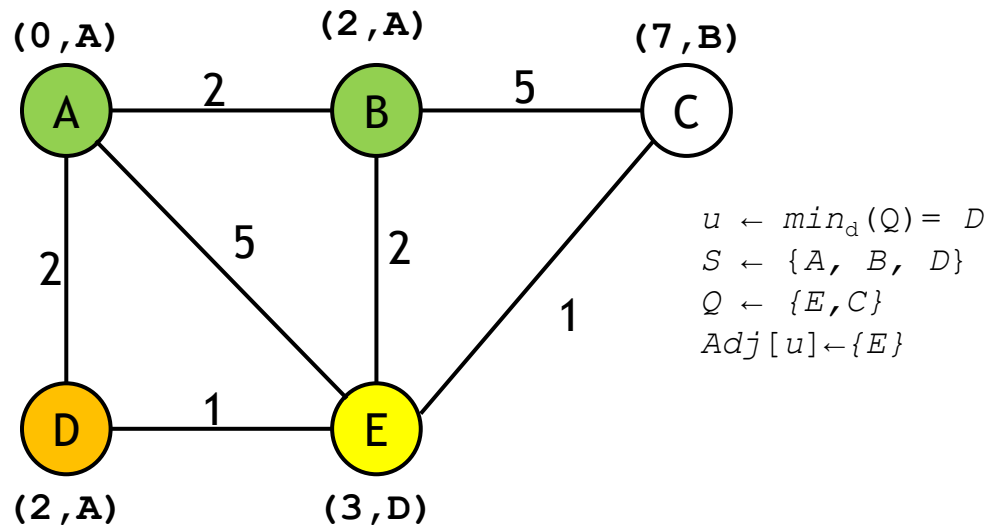
$\text{Adj}_Q[u] \leftarrow \{B, D, E\}$

Algoritmo do Caminho Mais Curto de Dijkstra (3)

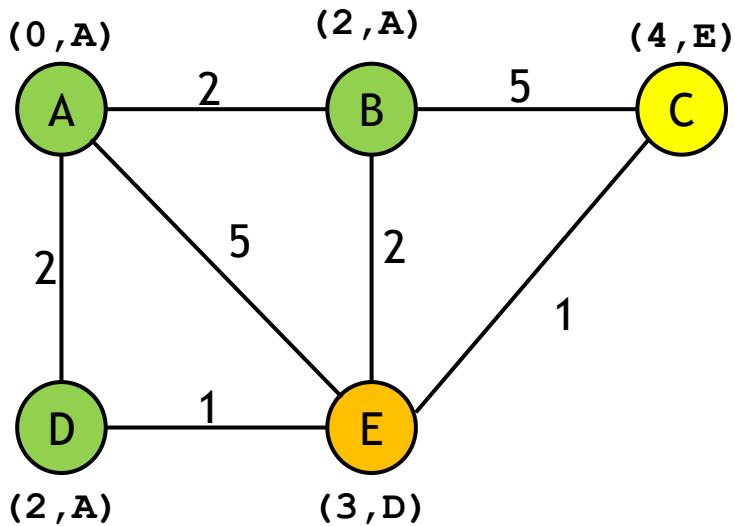


A escolha entre B e D como o próximo nó u é arbitrária.

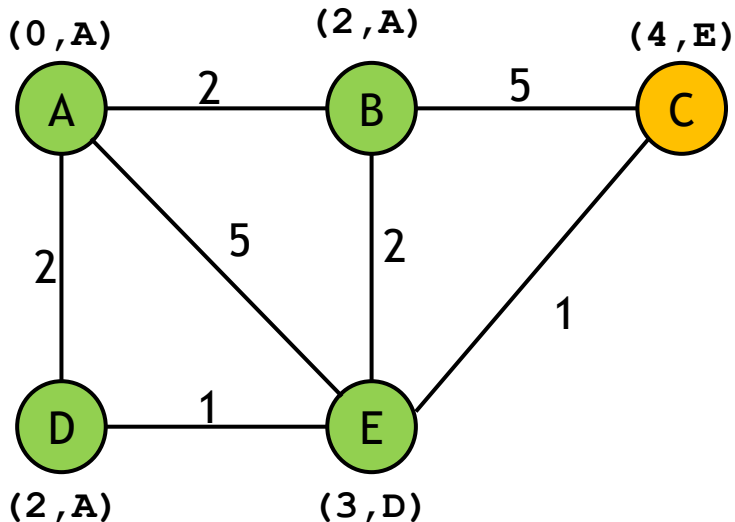
Se a escolha tivesse sido outra poderia resultar numa árvore de escoamento diferente, no entanto seriam igualmente óptimas (i.e., nenhuma seria melhor que a outra).



Algoritmo do Caminho Mais Curto de Dijkstra (4)

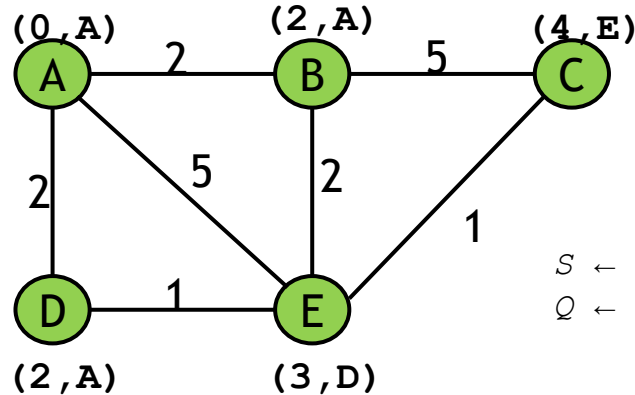


$u \leftarrow \min_d(Q) = E$
 $S \leftarrow \{A, B, D, E\}$
 $Q \leftarrow \{C\}$
 $Adj_Q[u] \leftarrow \{C\}$



$u \leftarrow \min_d(Q) = C$
 $S \leftarrow \{A, B, D, E, C\}$
 $Q \leftarrow \{\}$
 $Adj[u] \leftarrow \{\}$

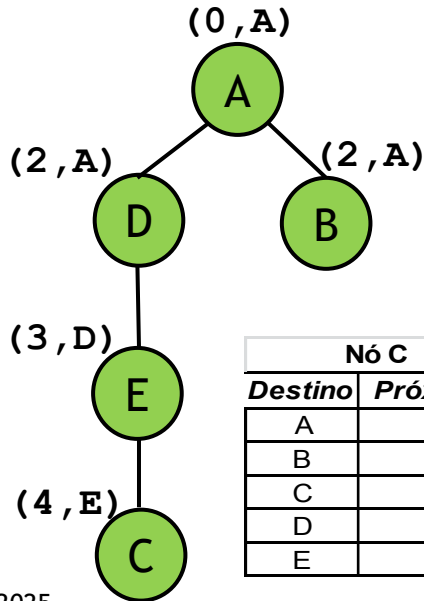
Algoritmo do Caminho Mais Curto de Dijkstra (5)



$S \leftarrow \{A, B, D, E, C\}$
 $Q \leftarrow \{\}$

Calculada a árvore de escoamento para o nó A.

Essa entrada pode ser actualizada na tabela de encaminhamento de todos os outros nós.



Nó A	
Destino	Próximo
A	A
B	-
C	-
D	-
E	-

Nó B	
Destino	Próximo
A	A
B	-
C	-
D	-
E	-

Nó C	
Destino	Próximo
A	E
B	-
C	-
D	-
E	-

Nó D	
Destino	Próximo
A	A
B	-
C	-
D	-
E	-

Nó E	
Destino	Próximo
A	D
B	-
C	-
D	-
E	-

Questão em aberto:

Como é que cada nó conhece a topologia completa da rede?

Encaminhamento Baseado no Estado da Ligação (*Link State Routing*)

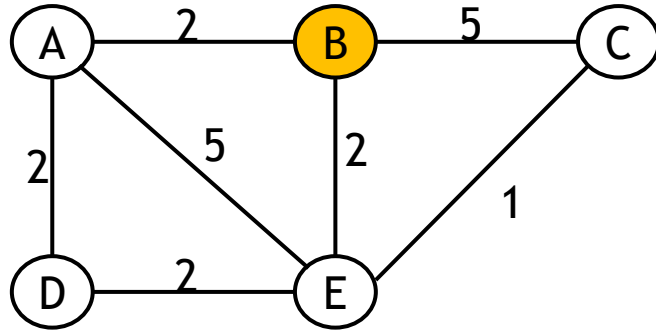
1. **Descobrir os vizinhos e seus endereços “Hello”:** \Rightarrow nomes globais e únicos.
2. **Medir o atraso ou custo para cada vizinho “Echo”:** Com resposta imediata mede o tempo de ida e volta, contanto ou não com o tempo da fila espera mede ou não a carga da rede.
3. **Construir o pacote com a informação apreendida:** periodicamente ou quando algo de importante ocorre, e.g., nó fora de serviço/em serviço.
4. **Envio do pacote para todos os nós “Inundação”:**
 - Loops \Rightarrow campo com número de sequência de 4 bytes.
 - Erros no número de sequência ou reinicialização de um nó \Rightarrow campo com a idade (*TTL*) decrementada pelos nós que percorre; se a idade = 0 o pacote é descartado.
 - Os pacotes não são transmitidos de imediato \Rightarrow pacotes duplicados e pacotes mais antigos são descartados.
 - Os pacotes são todos confirmados para protecção contra erros.
5. **Calcular o caminho mais curto:** depois de cada nó ter recebido um conjunto completo de pacotes executa um algoritmo, e.g., Dijkstra, e actualiza as suas tabelas de encaminhamento.

Problemas: dimensão das tabelas; tempo de cálculo; é necessário que os nós sejam “bem comportados”.

Exemplos: *Open Shortest Path First (OSPF), Intermediate System - Intermediate System (IS-IS).*

Link State Routing (2)

Pacotes construídos pelos diferentes nós:



Source: A
Seq: 15
Age: 6
B, 2
D, 2
E, 5

Source: B
Seq: 10
Age: 6
A, 2
C, 5
E, 2

Source: C
Seq: 14
Age: 6
B, 5
E, 1

Source: D
Seq: 16
Age: 6
A, 2
E, 2

Source: E
Seq: 12
Age: 6
A, 5
B, 2
C, 1
D, 2

Melhorias para a redução de mensagens desnecessárias no processo de *flooding*.

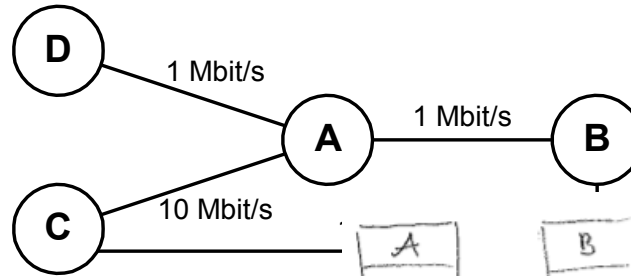
- Não enviar mensagens com número de sequência inferior.
- Esperar algum tempo antes de fazer *flooding* (não enviar por onde se recebeu).
 - Ex. estado no nó B ao fim de 3 e 6 unidades de tempo.

Source	Seq.	Age	SendTo			ConfirmTo		
			A	C	E	A	C	E
A	15	6	0	1	1	1	0	0
C	14	6	1	1	0	0	0	1
D	15	5	0	0	0	0	0	0
E	12	6	1	1	0	0	0	1

Source	Seq.	Age	SendTo			ConfirmTo		
			A	C	E	A	C	E
A	15	6	0	1	0	1	0	1
C	14	6	1	0	0	0	1	1
D	16	6	0	1	0	1	0	1
E	12	6	0	0	0	1	1	1

Exercício 4.1

Considere a seguinte rede na qual se utiliza o algoritmo de encaminhamento baseado no estado das ligações (*Link State Routing*). O comprimento de cada ligação é de 10 km e o atraso de propagação é de 5 μ s/km. Assuma que o custo de cada ligação é definido pelo tempo de transferência (i.e., tempo de transmissão mais tempo de propagação) e o tamanho médio igual a 125 bytes.



$$t_{tx1} = \frac{L}{R_{b1}} = \frac{125 \times 8}{1 \times 10^6} = 1 \times 10^{-3} \text{ seg} = 1 \text{ mseg}$$

$$t_{tx20} = \frac{L}{R_{b10}} = \frac{125 \times 8}{10 \times 10^6} = 0,1 \text{ mseg}$$

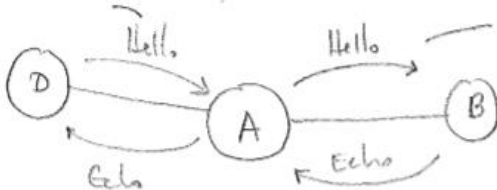
Indique a estrutura e os valores para o primeiro pacote com origem em um dos nós.

A	B	C	D
Seq = 0	Seq = 0	Seq = 0	Seq = 0
Age = 3	Age = 3	Age = 3	Age = 3
B, 1050	A, 1050	A, 150	A, 1050

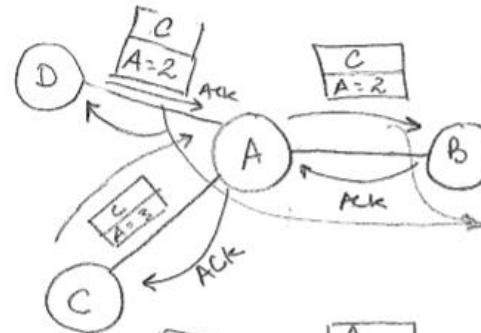
⇒ pacotes referidos como updates

Indique todos os tipos de pacotes recebidos e enviados pelo nó A.

Também ocorre com B e C

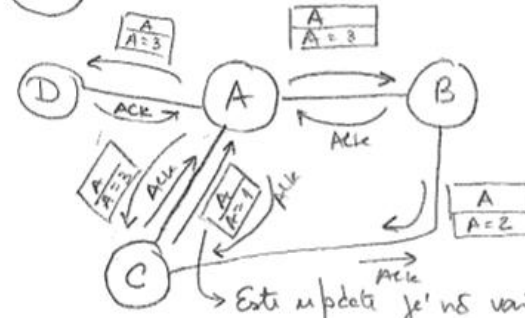


Existem também para...



→ o mesmo acontece com pacotes (updates) criados nos nós B e D

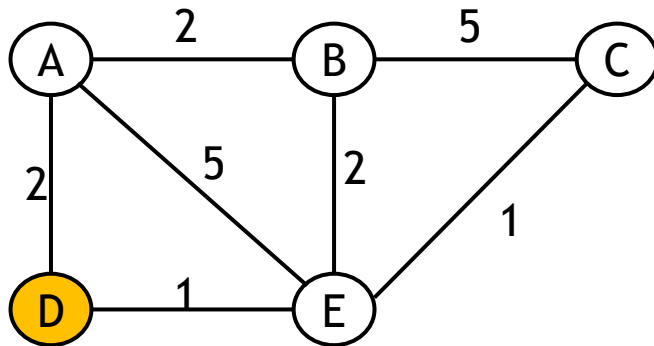
Resultantes do processo de flooding



Este update já não vai ser reenviado por A

Encaminhamento Baseado no Vector Distância (*Distance Vector Routing*)

- A informação de encaminhamento em cada nó é uma tabela (vector) com a melhor distância para cada destino e a linha a usar (em vez da distância pode ser o nº de saltos, o comprimento da fila de espera, o atraso ou outro).
- De **x** em **x** segundos (e.g., 30s) cada nó envia uma lista dos “atrasos” estimados para cada destino e recebe a mesma informação dos seus vizinhos.
- Com base nestas tabelas decide qual o vizinho mais vantajoso para encaminhar um determinado pacote para um dado destino, i.e., cria a tabela de encaminhamento.



Ex.: ARPANET, Internet (RIP-Routing Information Protocol), DECnet, Novell's IPX AppleTalk, Cisco

Informação
no nó D

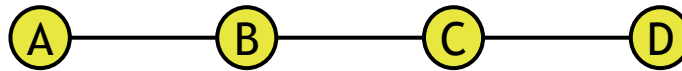
VECTORES RECEBIDOS (A, E) **NOVA ESTIMATIVA**

	A	E	TEMPO	SAIDA
A	0	4	2	A
B	2	2	3	E
C	7	1	2	E
D	2	1	0	D
E	5	0	1	E
	2	1		

Custo medido
pelo nó D

Distance Vector Routing: Count-to-Infinity

- Propaga as boas notícias depressa, mas as más são propagadas devagar (Problema *Count-to-Infinity*).



Distância ao nó A

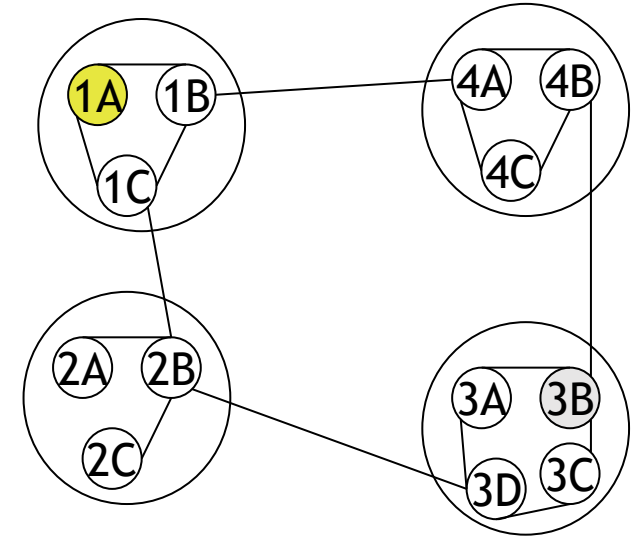
B	C	D	
∞	∞	∞	nó A fora de serviço
∞	∞	∞	nó A em serviço
1	∞	∞	após primeira lista
1	2	∞	após segunda lista

Distância ao nó A

B	C	D	
1	2	3	nó A em serviço
1	2	3	nó A fora de serviço
3	2	3	após primeira lista
3	4	3	após segunda lista
5	4	5	após terceira lista
5	6	5	após quarta lista
7	6	7	após quinta lista

Encaminhamento Hierárquico

- Numa rede de grandes dimensões não é viável que todos os nós conheçam todos os nós.
- Divide-se a rede em níveis. Quantos ? É outro problema.
- Cada sub-rede tem conhecimento dos seus nós e a saída para as restantes sub-redes.
- O encaminhamento pode ser sub-óptimo para alguns destinos, e.g., de 1A para 3B

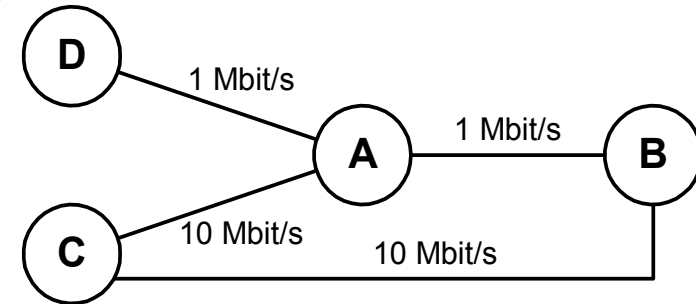


Dest	Linha	Salto
1A	-	-
1B	1B	1
1C	1C	1
2A	1C	3
2B	1C	2
2C	1C	3
3A	1C	4
3B	1B	4
3C	1C	4
3D	1C	3
4A	1B	2
4B	1B	3
4C	1B	3

Dest	Linha	Salto
1A	-	-
1B	1B	1
1C	1C	1
2	1C	2
3	1C	3
4	1B	2

Exercício 4.2

Considere a seguinte rede na qual se utiliza o algoritmo de encaminhamento baseado no vector distância (*Distance Vector Routing*). O comprimento de cada ligação é de 10 km e o atraso de propagação é de 5 μ s/km. Assuma que o custo de cada ligação é definido pelo tempo de transferência (i.e., tempo de transmissão mais tempo de propagação) de uma mensagem de tamanho médio igual a 125 bytes.



- Mostre a evolução das tabelas de encaminhamento assim como os vectores trocados até se atingir um ponto de estabilidade.
- Indique como poderia, mesmo sem calcular os vectores trocados e a actualização das tabelas de encaminhamento, determinar o número de trocas de vectores que garante:
 - A conectividade do nó D com todos os outros nós.
 - Que é conhecido por D o próximo nó do caminho óptimo, bem como o seu custo, para todos os outros nós.
- Compare, justificando, os algoritmos de encaminhamento *Link State Routing* e *Distance Vector Routing* em termos do tempo que demoram as tabelas de encaminhamento a adaptar-se a alterações na rede quando uma ligação deixa de estar em actividade.

Exercício 4.2 a)

Passo I. **Medidas** por cada nó.

(Colocar o custo em cada uma das ligações)

Nó A					
Destino	Vectores Recebidos			Tabela Local	
	B	C	D	Custo	Próximo
A				0	A
B				1050	B
C				150	C
D				1050	D
Medido	1050	150	1050		

Nó B				
Destino	Vectores Recebidos		Tabela Local	
	A	C	Custo	Próximo
A			1050	A
B			0	B
C			150	C
D			∞	-
Medido	1050	150		

Nó C				
Destino	Vectores Recebidos		Tabela Local	
	A	B	Custo	Próximo
A			150	A
B			150	B
C			0	C
D			∞	-
Medido	150	150		

Nó D			
Destino	Vectores	Tabela Local	
	A	Custo	Próximo
A		1050	A
B		∞	-
C		∞	-
D		0	D
Medido	1050		

(Os vectores são os custos já aprendidos por cada nó e enviados para os seus **vizinhos**)

A	
A	0
B	1050
C	150
D	1050

B	
A	1050
B	0
C	150
D	∞

C	
A	150
B	150
C	0
D	∞

D	
A	1050
B	∞
C	∞
D	0

Passo II. Actualização baseada nos **vectores** recebidos.

Nó A					
Destino	Vectores Recebidos			Tabela Local	
	B	C	D	Custo	Próximo
A	1050	150	1050	0	A
B	0	150	∞	300	C
C	150	0	∞	150	C
D	∞	∞	0	1050	D
Medido	1050	150	1050		

Nó B				
Destino	Vectores Recebidos		Tabela Local	
	A	C	Custo	Próximo
A	0	150	300	C
B	1050	150	0	B
C	150	0	150	C
D	1050	∞	2100	A
Medido	1050	150		

Nó C				
Destino	Vectores Recebidos		Tabela Local	
	A	B	Custo	Próximo
A	0	1050	150	A
B	1050	0	150	B
C	150	150	0	C
D	1050	∞	1200	A
Medido	150	150		

Nó D			
Destino	Vectores	Tabela Local	
	A	Custo	Próximo
A	0	1050	A
B	1050	2100	A
C	150	1200	A
D	1050	0	D
Medido	1050		

(Os vectores são os custos já aprendidos por cada nó e enviados para os seus **vizinhos**)

A	
A	0
B	300
C	150
D	1050

B	
A	300
B	0
C	150
D	2100

C	
A	150
B	150
C	0
D	1200

D	
A	1050
B	2100
C	1200
D	0

Passo III. Actualização baseada nos **vectores** recebidos.

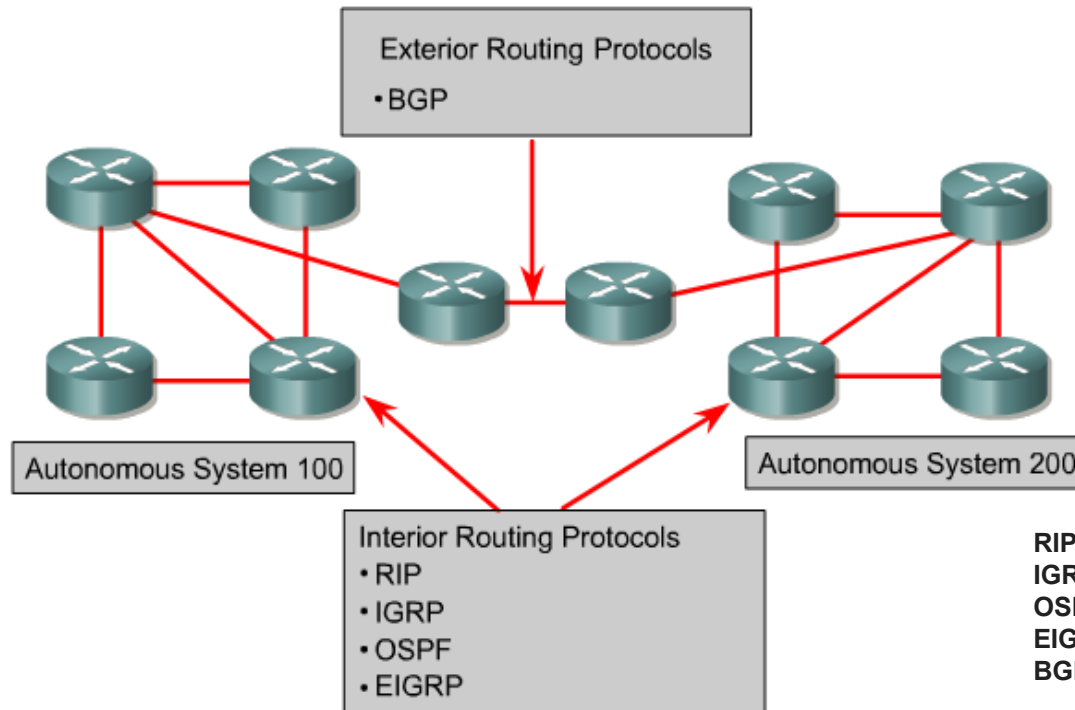
Nó A					
Destino	Vectores Recebidos			Tabela Local	
	B	C	D	Custo	Próximo
A	300	150	1050	0	A
B	0	150	2100	300	C
C	150	0	1200	150	C
D	2100	1200	0	1050	D
Medido	1050	150	1050		

Nó B				
Destino	Vectores Recebidos		Tabela Local	
	A	C	Custo	Próximo
A	0	150	300	C
B	300	150	0	B
C	150	0	150	C
D	1050	1200	1350	C
Medido	1050	150		

Nó C				
Destino	Vectores Recebidos		Tabela Local	
	A	B	Custo	Próximo
A	0	300	150	A
B	300	0	150	B
C	150	150	0	C
D	1050	2100	1200	A
Medido	150	150		

Nó D			
Destino	Vectores	Tabela Local	
	A	Custo	Próximo
A	0	1050	A
B	300	1350	A
C	150	1200	A
D	1050	0	D
Medido	1050		

Visão Geral dos Protocolos de Routing



RIP - Routing Information Protocol
IGRP - Interior Gateway Routing Protocol
OSPF - Open Shortest Path First
EIGRP - Enhanced IGRP
BGP - Border Gateway Protocol

- RIP - Protocolo de *routing* interior com base no vector distância
- IGRP - Protocolo de *routing* interior da Cisco com base no vector distância
- OSPF - Protocolo de *routing* interior com base no estado das ligações
- EIGRP - Protocolo de *routing* interior da Cisco com base no vector distância
- BGP - Protocolo de *routing* exterior com base no vector distância

Routing dentro de sistemas autónomos (AS - Autonomous Systems): 3 tipos

- Static routing
 - Apenas usado em domínios muito pequenos
- Distance vector routing
 - **RIP: Routing Information Protocol**
 - Menos usado, mas, apesar das suas limitações, é bastante fácil de configurar em domínios pequenos
 - **IGRP: Interior Gateway Routing Protocol** (substituído pelo EIGRP)
 - Protocolo proprietário da Cisco que tenta resolver alguns problemas do RIP
- Link-state routing
 - **OSPF: Open Shortest Path First**
 - Largamente usado em grandes redes
 - **IS-IS: Intermediate System- Intermediate System**
 - Largamente usado por ISPs
 - Multiprotocolo (OSPF usa apenas IP)

RIP (1)

- Baseado no algoritmo Distance Vector Routing
- Popularizado pela distribuição BSD-UNIX em 1982
- Versão 1 (1988): <https://tools.ietf.org/html/rfc1058>
- Versão 2 (1998): <https://tools.ietf.org/html/rfc2453>
- Métricas de distância : nº de hops (max = 15 hops)
- Tipos de mensagens
 - Request
 - Pede a tabela de encaminhamento a nós vizinhos. Indica se deve obter uma dada subrede, ou toda a tabela
 - Response
 - Enviado por um dispositivo, para anunciar informação local da sua tabela de encaminhamento. Estes anúncios com os vectores de distância são enviados nas seguintes circunstâncias:
 - Automaticamente e aleatoriamente a cada 30 seg (aproximadamente)
 - Como resposta a um Request gerado por outro nó RIP
 - Se o triggered update for suportado, sempre que um custo mudar

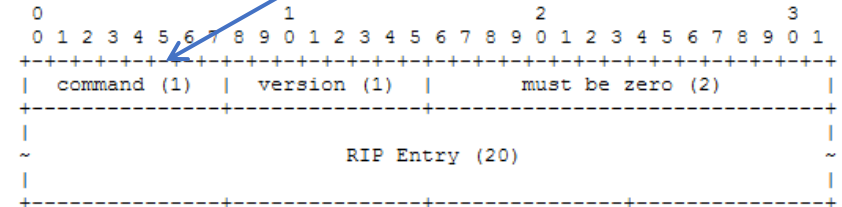
RIP (2)

•Response:

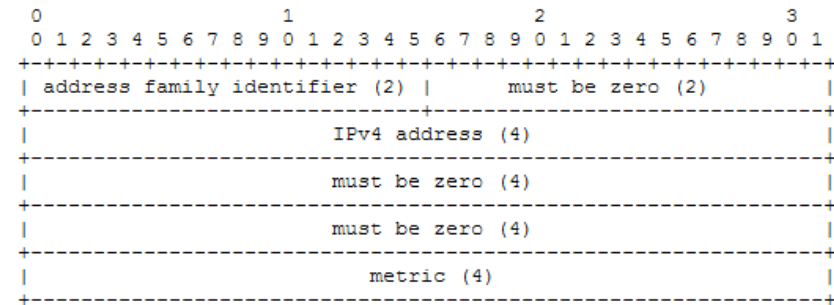
- enviados via segmentos UDP, limitados a 512 bytes
- pode ser apenas um sumário da tabela de encaminhamento do router
 - lista até 25 destinos de rede dentro do AS
- Para cada rede destino, a mensagem:
 - Indica o prefixo IP
 - Contem a distância entre o seu nó e o destino
- A tabela local é actualizada se com os anúncios (Response) recebidos for possível determinar um caminho mais curto

The RIP packet format is:

command - request /response



There may be between 1 and 25 (inclusive) RIP entries. A RIP-1 entry has the following format:

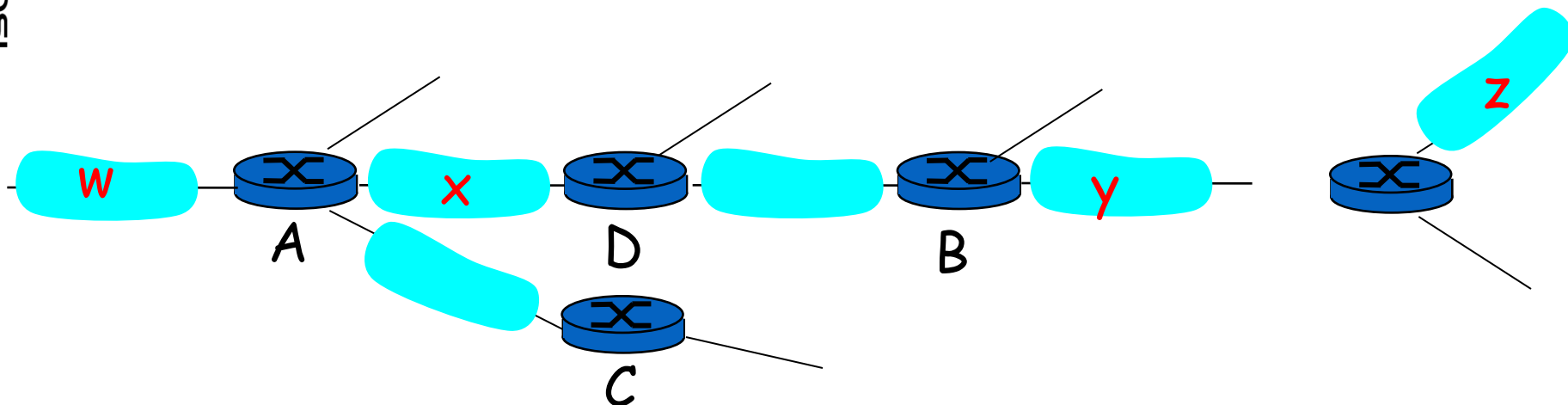


Field sizes are given in octets. Unless otherwise specified, fields contain binary integers, in network byte order, with the most-significant octet first (big-endian). Each tick mark represents one bit.

RIP (3)

- Temporizadores (*timers*)
 - O RIP usa vários temporizadores na sua operação:
 - **Update interval** - é o intervalo de tempo entre *routing updates*, por defeito é 30 s.
 - **Timeout interval** - se um caminho não é actualizado durante um intervalo de tempo igual ao *timeout interval* (e.g., 180 s), então esse caminho é marcado como inválido e é removido da tabela de encaminhamento. Um caminho também pode ser retirado da tabela se a métrica chegar a infinito (i.e., 16 hops). No entanto, o caminho inválido permanece na tabela durante um determinado intervalo para que os vizinhos possam ser notificados que o caminho foi retirado. Este intervalo de tempo é o *hold-down interval*. Só depois do *hold-down interval* expirar é que o caminho é eliminado da tabela.
 - **Hold-down interval** - intervalo de tempo que o caminho inválido tem que permanecer na tabela de encaminhamento.

RIP: Exemplo (1)



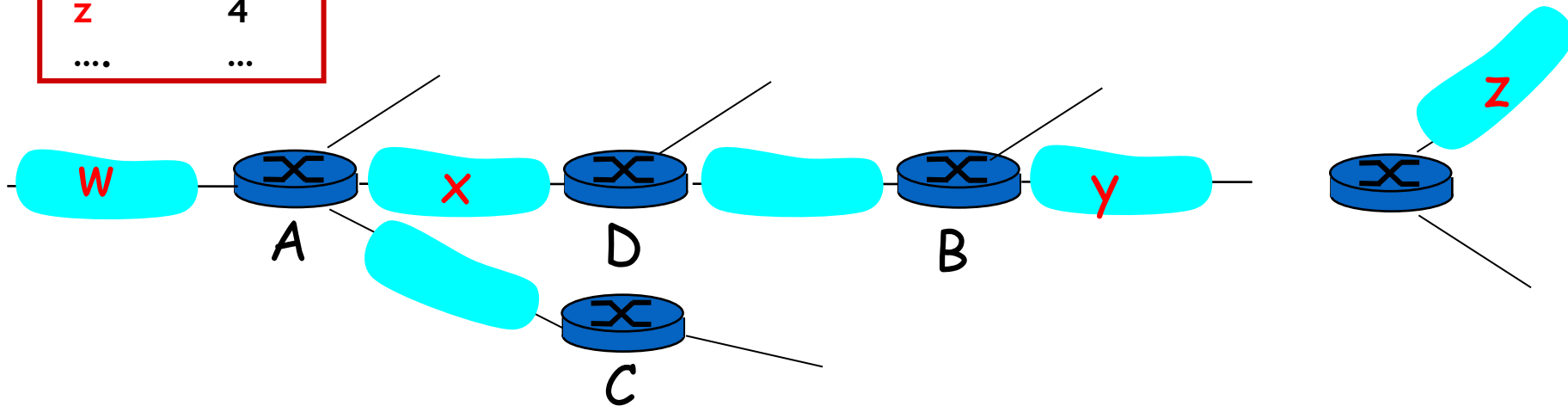
Rede destino	Próximo router	Num. de hops ao dest.
W	A	2
Y	B	2
Z	B	7
X	--	1
...

Tabela de encaminhamento em D

RIP: Exemplo (2)

Dest.	hops
w	-
x	-
z	4
...	...

Anúncio de A para D



Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	B A	7 5
x	--	1
...

Tabela de encaminhamento em D

RIP: Falha de ligação e Recuperação

- Como lidar com falhas nas ligações?
 - Existe um temporizador
 - se nenhum anúncio é escutado após 180s --> vizinho/ligação declarada em baixo
 - Rotas via vizinhos invalidadas
 - Tabela de encaminhamento recalculada
 - Novos anúncios devem ser enviados aos vizinhos
 - Vizinhos por sua vez enviam novos anúncios (se as tabelas mudarem)
- Como evitar o problema da contagem para o infinito? (recordar slide 59)
 - Limitar a distância máxima entre *routers*
 - infinito > 15 (considera-se infinito um custo igual a 16)
 - Split horizon
 - O router A não anuncia ao router B os caminhos pelos quais envia pacotes via B
 - Split horizon with poison reverse
 - Informação de custos, enviada a vizinhos, de caminhos que passem por eles é indicada como infinito (custo = 16)
 - Route poison
 - Mal uma rota falha, envia imediatamente custo infinito, e não espera 180s

RIP - limitações

- Limite no custo dos caminhos - caminho com custo máximo de 15.
Diâmetro máximo da rede!
- Atualizações de tabelas de forma intensiva - problema em ligações de capacidade reduzida.
- Convergência relativamente lenta - algoritmo de vector de distâncias; depende de temporizadores para iniciar os anúncios.

CP5 e CP6: Nível de Rede e Nível de Rede na Internet (Exercícios)

- Sebenta de Exercícios e Aplicações
 - Capítulo 4
- Questões de provas escritas
 - Ver plataforma de e-learning com provas de anos anteriores.
- Mini-teste
 - Mini-teste para avaliação na plataforma de e-learning.