centro universitário

Redes de Computadores

Camada de Rede - III

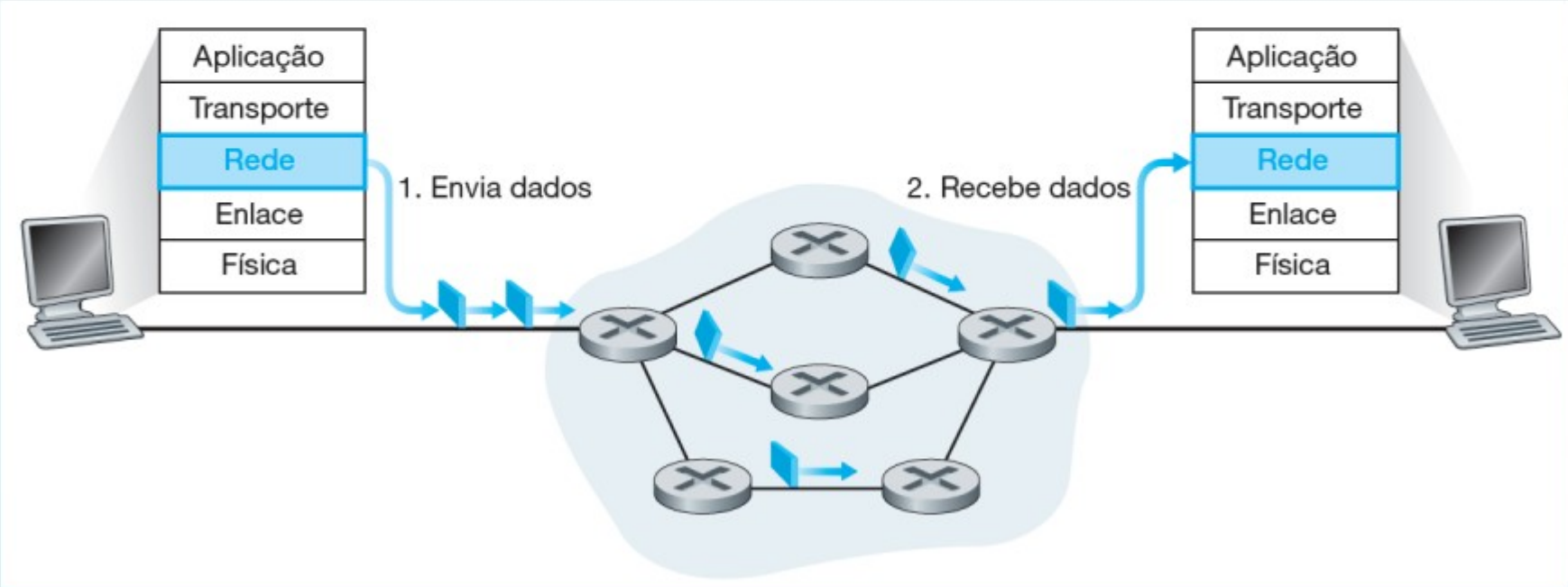
Prof. Me. Ricardo Girnis Tombi

Conteúdo adaptado de:

Redes de Computadores e a Internet. Ed. Pearson
J. F Kurose e K. W. Ross

Redes de datagramas (pacotes)

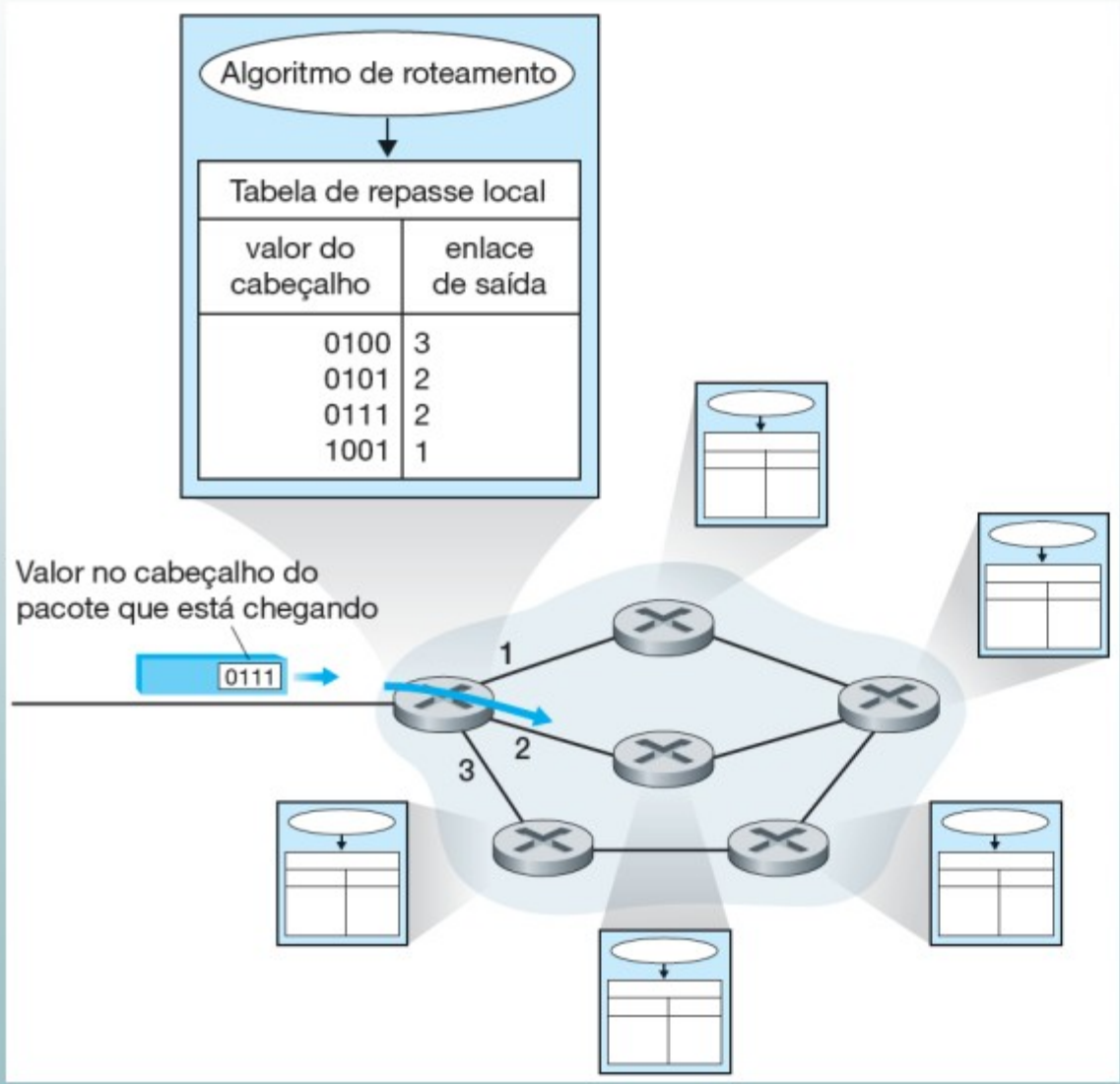
Em uma rede de datagramas, toda vez que um sistema final quer enviar um pacote, ele marca o pacote com o endereço do sistema final de destino e então o envia para dentro da rede.



Repasse e Roteamento

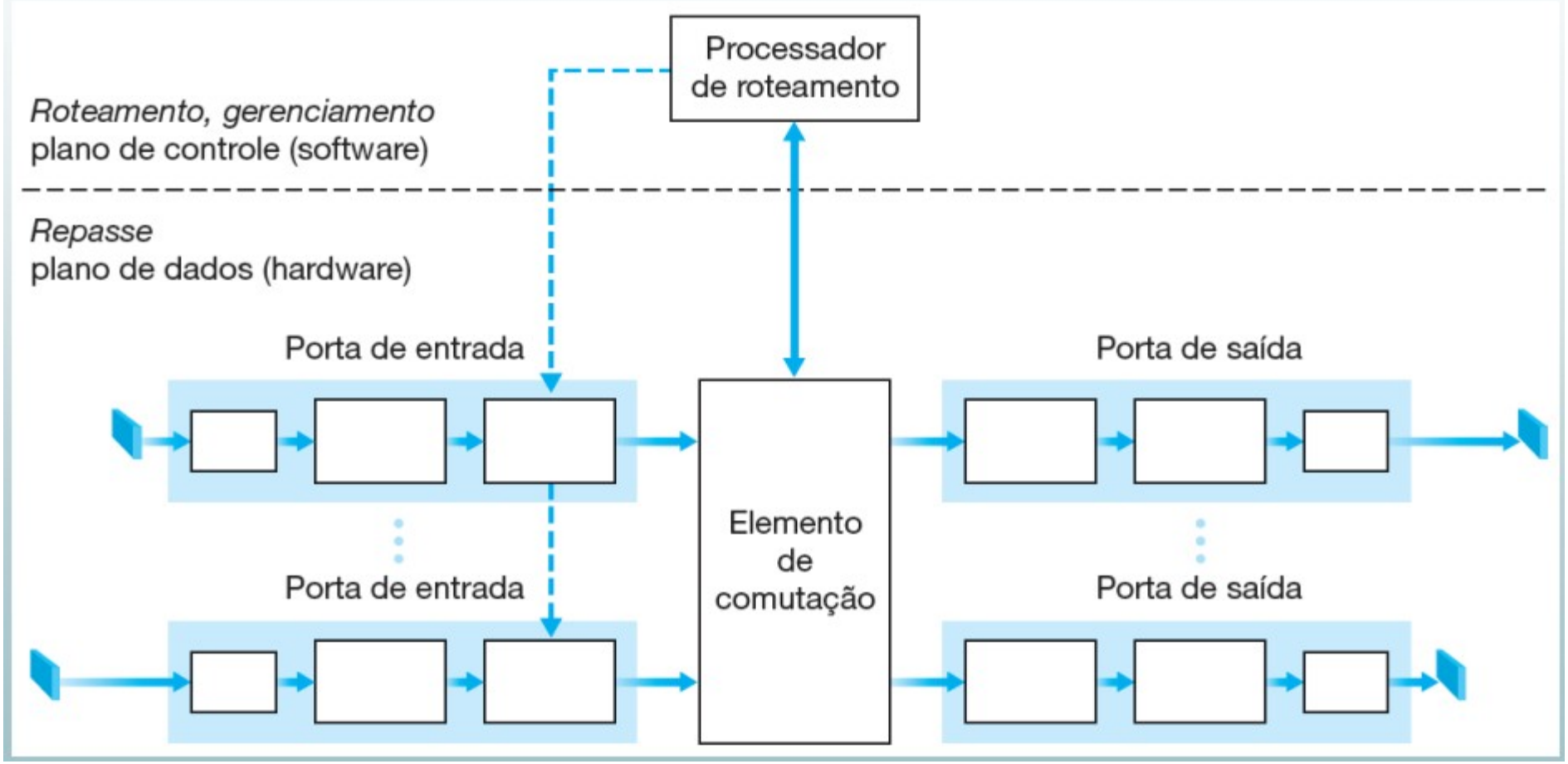
- O papel da camada de rede é transportar pacotes de um hospedeiro remetente a um hospedeiro destinatário.
- **Repasse**. Quando um pacote chega ao enlace de entrada de um roteador, este deve conduzi-lo até o enlace de saída apropriado.
- **Roteamento**. A camada de rede deve determinar a rota ou o caminho tomado pelos pacotes ao fluírem de um remetente a um destinatário.

Repasse e Roteamento



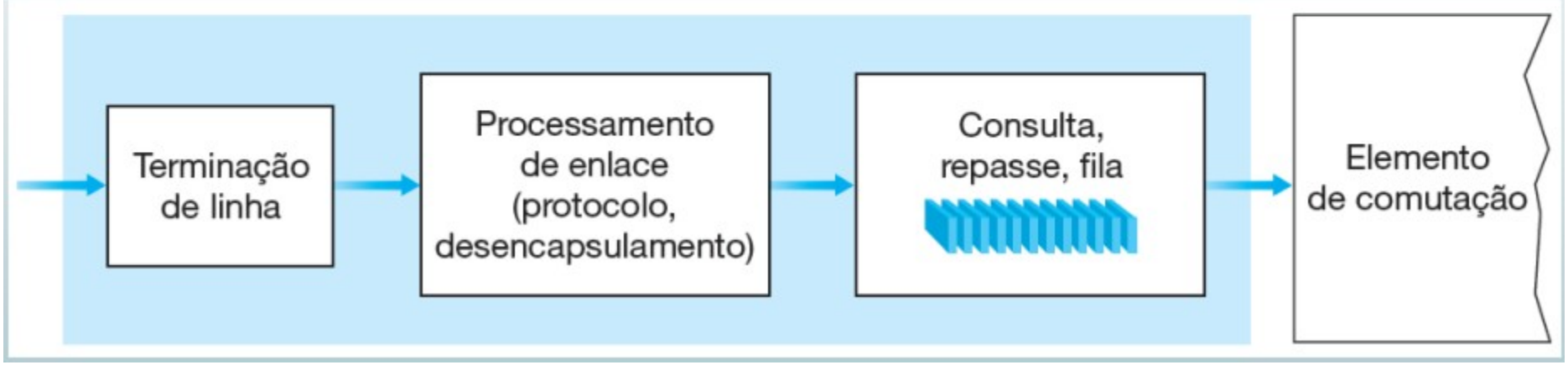
Algoritmos de roteamento determinam valores em tabelas de repasse.

Arquitetura de roteador



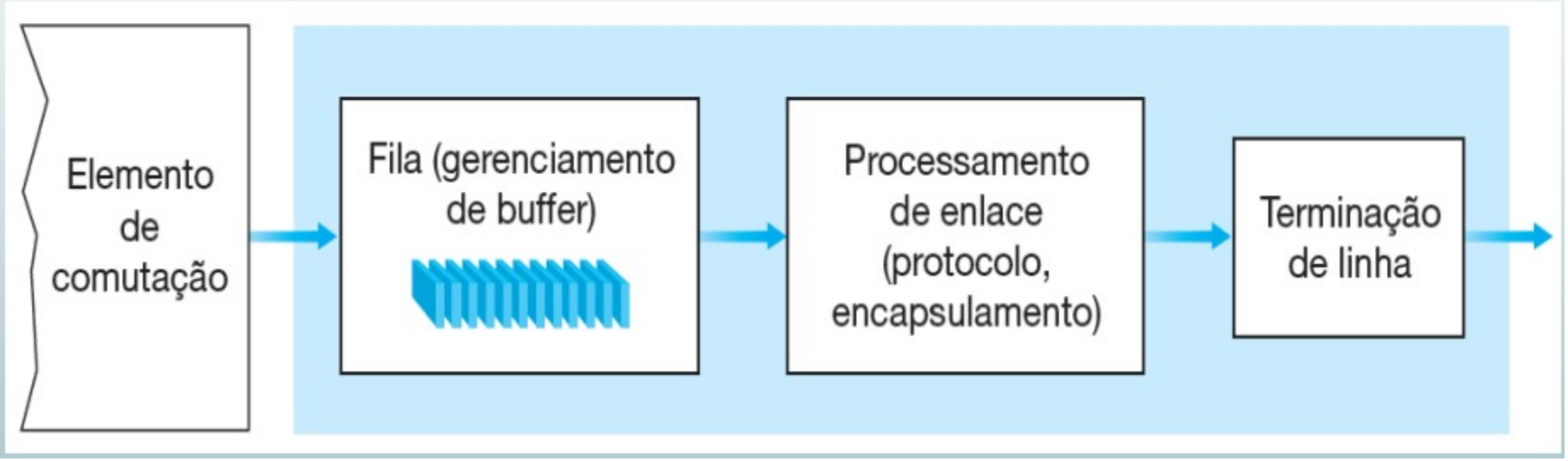
Processamento de entrada

- Processamento de porta de entrada

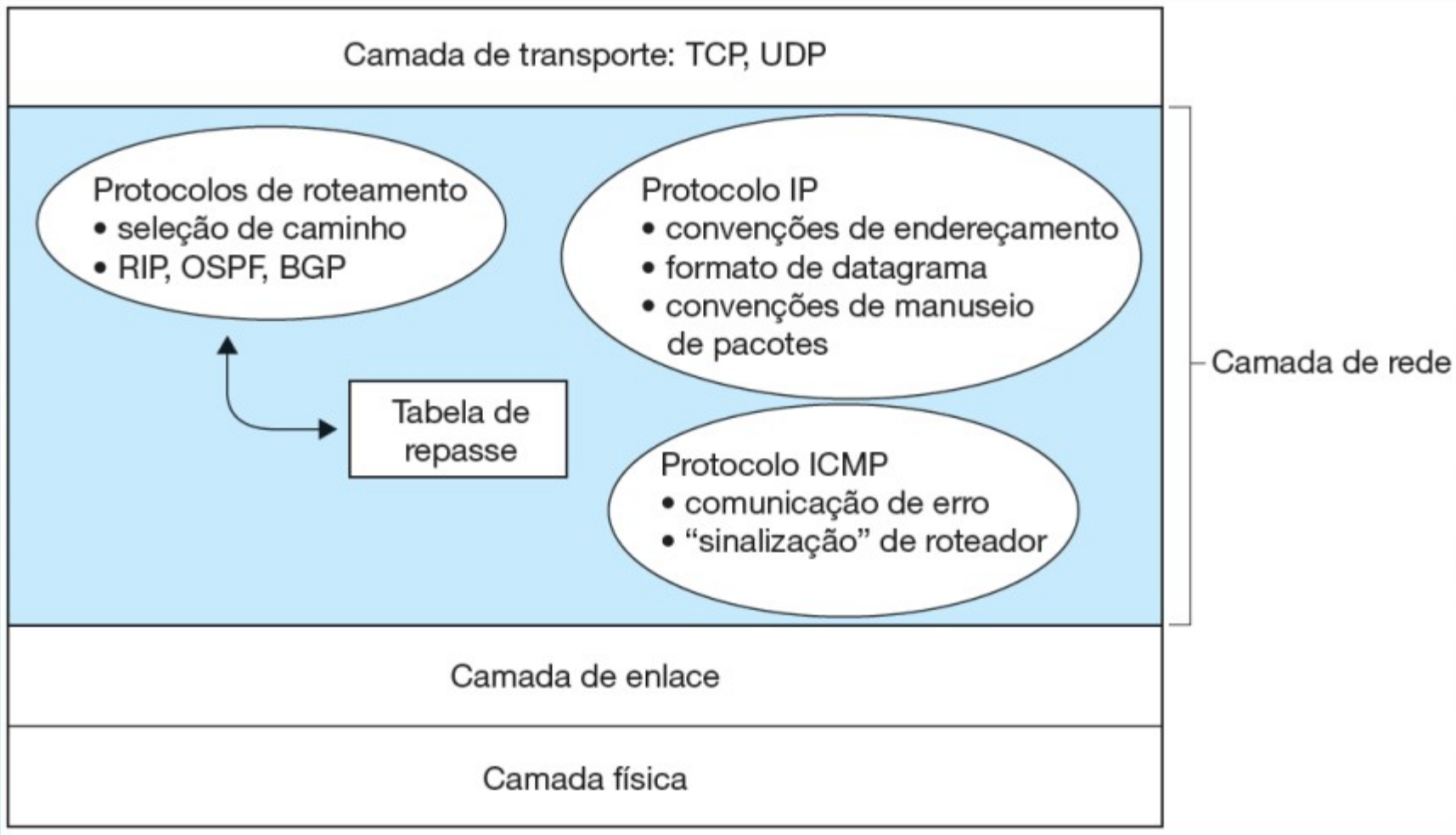


Processamento de saída

- Processamento de porta de saída



Camada de Rede



Algoritmos de roteamento

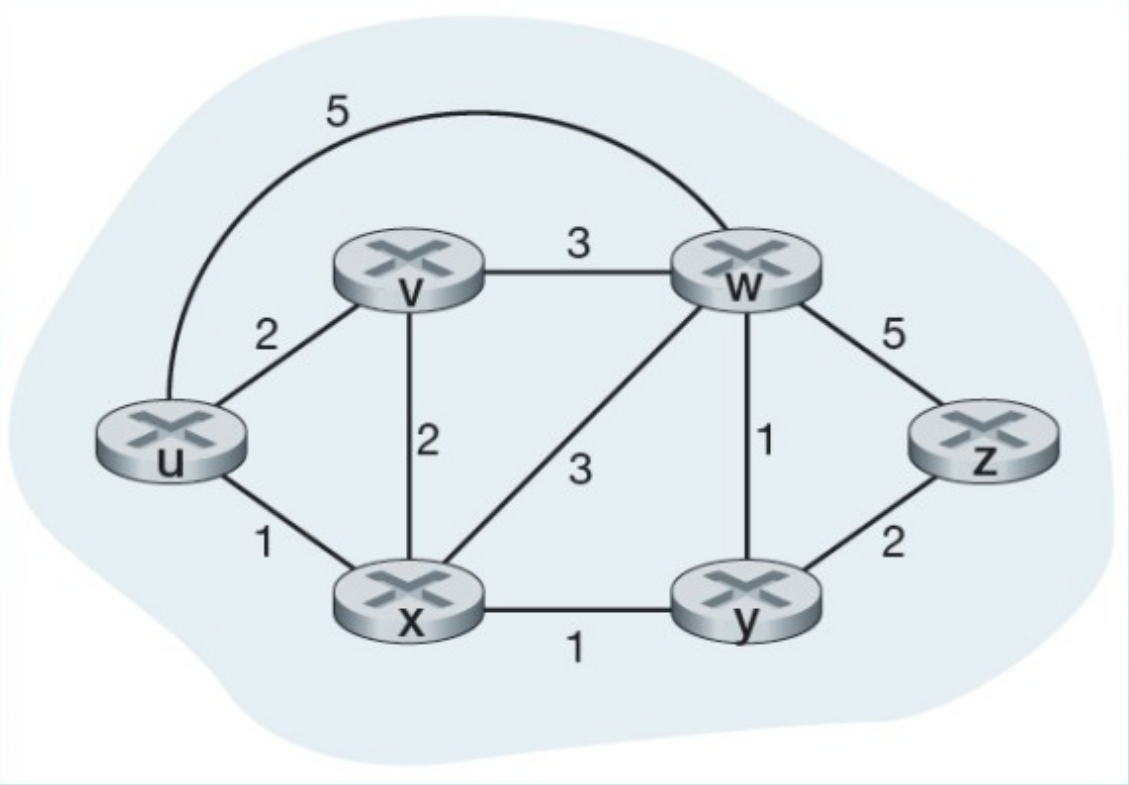
- Em geral um hospedeiro está ligado diretamente a um roteador, o **roteador *default*** para esse hospedeiro.

- Denominamos **roteador de origem** o roteador *default* do hospedeiro de origem e **roteador de destino** o roteador *default* do hospedeiro de destino.

- O problema de rotear um pacote do hospedeiro de origem até o hospedeiro de destino se reduz, claramente, ao problema de direcionar o pacote do roteador de origem ao roteador de destino.

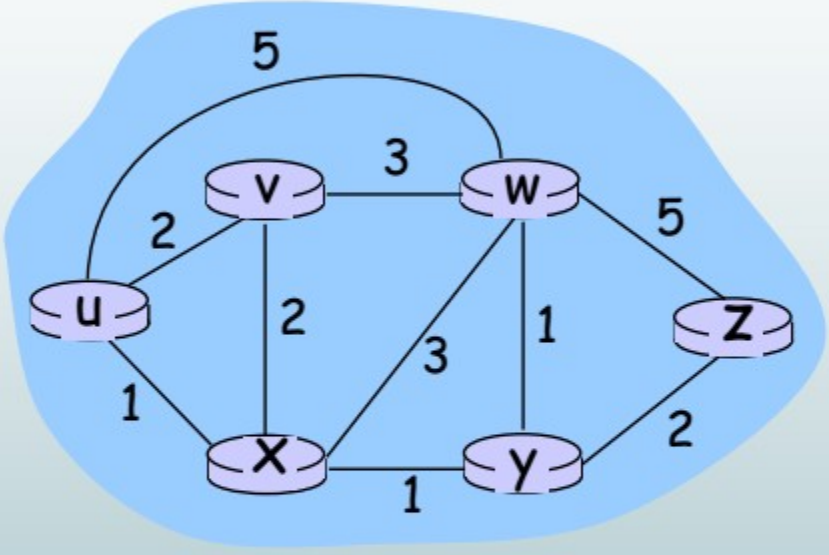
Abstração de grafo

Um grafo é usado para formular problemas de roteamento



Qual o melhor caminho de **u** para **z** ?

Abstração de grafo

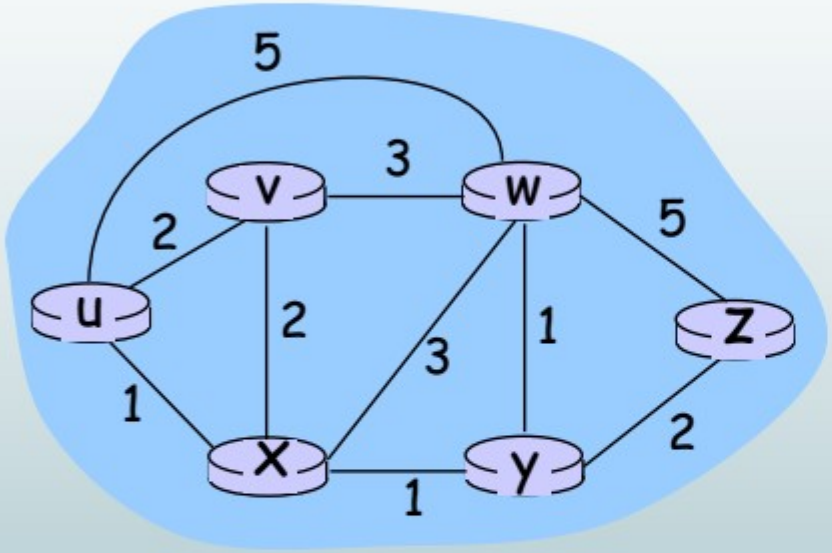


Grafo: G = (N,E)

N = conjunto de roteadores = { u, v, w, x, y, z }

E = conjunto de enlaces = { (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

Abstração de grafo



- c(x,x') = custo do enlace (x,x')

  - p. e., c(w,z) = 5

- custo poderia ser sempre 1, ou inversamente relacionado à largura ou inversamente relacionado ao congestionamento

Custo do caminho $(x_1, x_2, x_3, \ldots, x_p) = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$

Pergunta: Qual é o caminho de menor custo entre u e z?

algoritmo de roteamento: algoritmo que encontra o caminho de menor custo

Algoritmos de roteamento

- Um **algoritmo de roteamento global** calcula o caminho de menor custo entre uma origem e um destino usando conhecimento completo e global sobre a rede.

- Em um **algoritmo de roteamento descentralizado**, o cálculo do caminho de menor custo é realizado de modo iterativo e distribuído.

- Em **algoritmos de roteamento estáticos**, as rotas mudam muito devagar ao longo do tempo, muitas vezes como resultado de intervenção humana

Algoritmos de roteamento

- **Algoritmos de roteamento dinâmicos** mudam os caminhos de roteamento à medida que mudam as cargas de tráfego ou a topologia da rede.

- Em um **algoritmo sensível à carga**, custos de enlace variam dinamicamente para refletir o nível corrente de congestionamento no enlace subjacente.

Estado de enlace

algoritmo de Dijkstra

- nova topologia, custos de enlace conhecidos de todos os nós
  - realizado por "broadcast de estado do enlace"
  - todos os nós têm a mesma informação
- calcula caminhos de menor custo de um nó ("origem") para todos os outros nós
  - da tabela de repasse para esse nó
- iterativo: após k iterações, sabe caminho de menor custo para k destinos

notação:

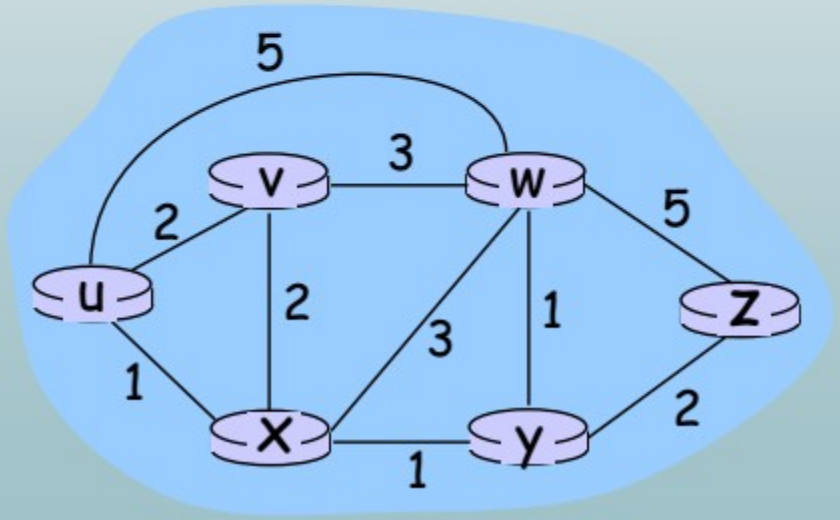
- c(x,y): custo do enlace do nó x até y; $= \infty$ se não forem vizinhos diretos
- D(v): valor atual do custo do caminho da origem ao destino v
- p(v): nó predecessor ao longo do caminho da origem até v
- N': conjunto de nós cujo caminho de menor custo é definitivamente conhecido

Algoritmo de Dijkstra

```
1  Inicialização:
2   N' = {u}
3   para todos os nós v
4     se v adjacente a u
5         então D(v) = c(u,v)
6     senão D(v) = ∞
7
8  Loop
9    acha w não em N' tal que D(w) é mínimo
10    acrescenta w a N'
11    atualiza D(v) para todo v adjacente a w e não em N' :
12       D(v) = min( D(v), D(w) + c(w,v) )
13    /* novo custo para v é custo antigo para v ou custo conhecido
14     do caminho mais curto para w + custo de w para v */
15 até todos os nós em N'
```

Algoritmo de Dijkstra

Etapa	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxv	2,u	3,y			4,y
3	uxvv		3,y			4,y
4	uxvvw					4,y
5	uxvvwz					



Algoritmo de Dijkstra

Árvore resultante do caminho mais curto a partir de u:

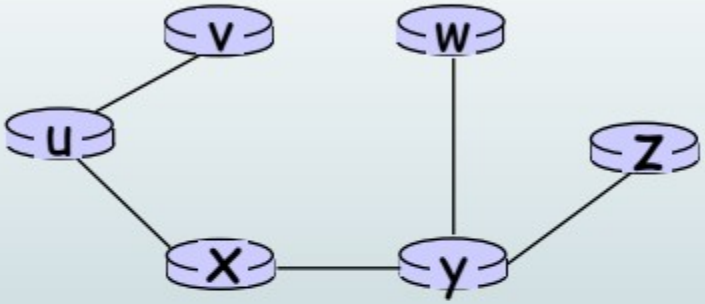


Tabela de repasse resultante em u:

destino	enlace
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Algoritmo de vetor de distância

Equação de Bellman-Ford (programação dinâmica)

Defina

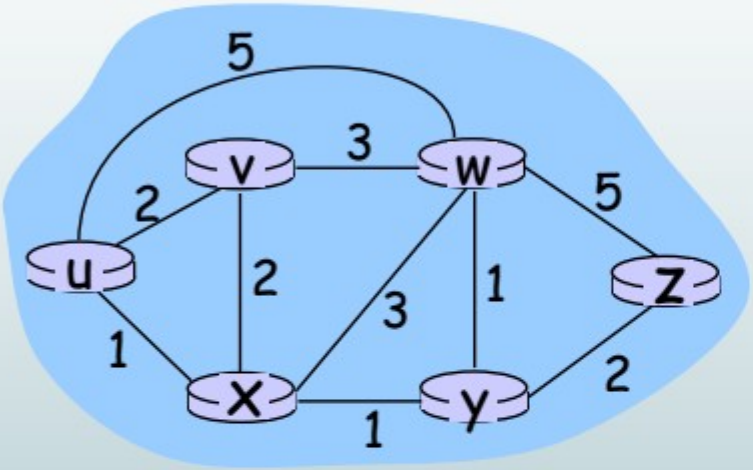
$d_x(y) :=$ custo do caminho de menor custo de x para y

Depois

$$d_x(y) = \min \{c(x,v) + d_v(y) \}$$

onde min assume todos os vizinhos v de x

Algoritmo de vetor de distância



Inicialização:

$d_v(z) = 5,\ d_x(z) = 3,\ d_w(z) = 3$

Equação B-F diz:

$$d_u(z) = \min \{ c(u,v) + d_v(z),\ c(u,x) + d_x(z),\ c(u,w) + d_w(z) \}$$
$$= \min \{2 + 5,\ 1 + 3,\ 5 + 3\} = 4$$

Portanto:

Nó que alcança mínimo é o próximo salto no caminho mais curto

➜ tabela de repasse

Algoritmo de vetor de distância

- $D_x(y)$ = estimativa do menor custo de x para y
- nó x sabe custo de cada vizinho v: $c(x,v)$
- nó x mantém vetor de distância $\mathbf{D}_x = [D_x(y): y \in N]$
- nó x também mantém vetor de distância de seus vizinhos
  - para cada vizinho v, x mantém
    $\mathbf{D}_v = [D_v(y): y \in N]$

mudanças de custo do enlace:

- nó detecta mudança de custo no enlace local
- atualiza informação de roteamento, recalcula vetor de distância
- se DV mudar, notifica vizinhos

Exemplos de implementações

- Distance Vector:

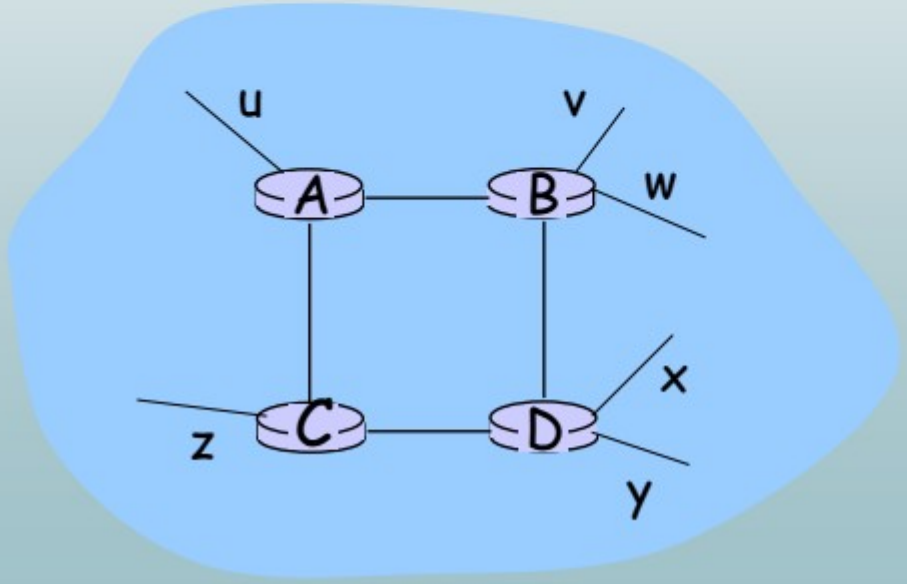
  RIP – Routing Information Protocol

- Link State:

  OSPF - Open Shortest Path First

R I P

- Algoritmo de vetor de distância
- Incluído na distribuição BSD-UNIX em 1982
- Métrica de distância: # de saltos (máx. = 15 saltos)



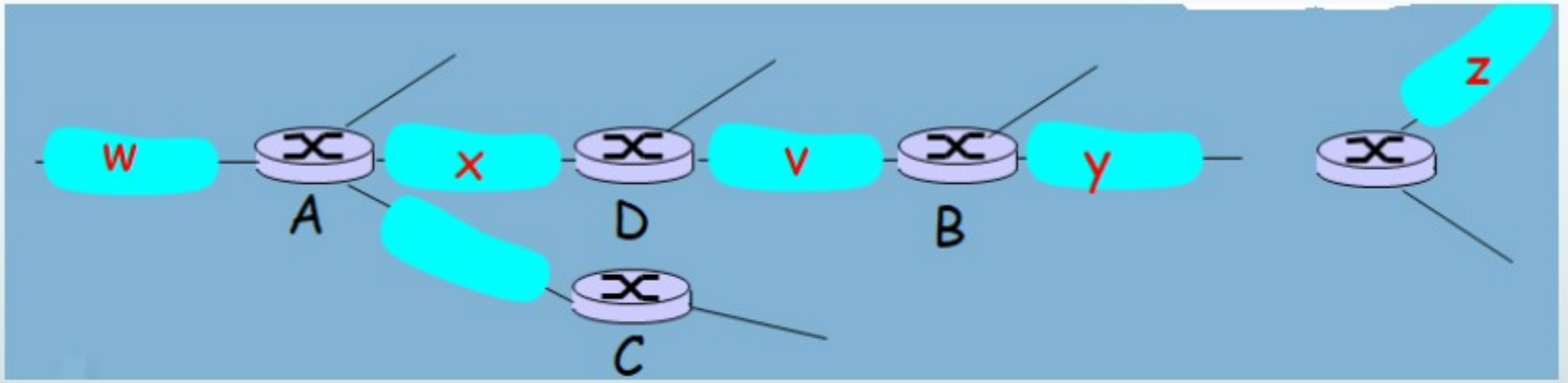
Do roteador A às sub-redes:

destino	saltos
u	1
v	2
w	2
x	3
y	3
z	2

R I P

- *vetores de distância:* trocados entre vizinhos a cada 30 s por meio de mensagem de resposta (também conhecida como anúncio)
- Cada anúncio: lista de até 25 sub-redes de destino dentro do AS

RIP



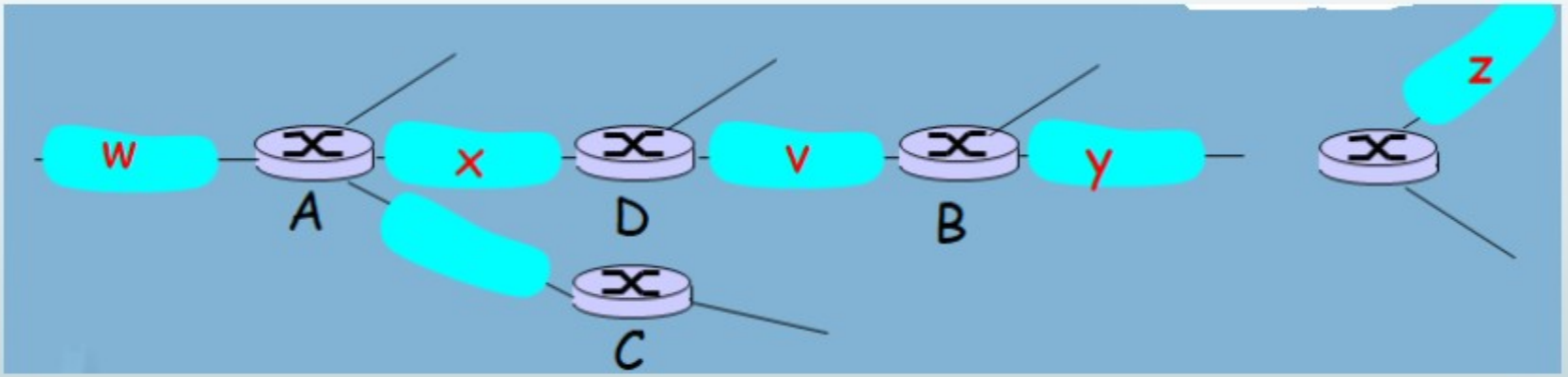
Rede de destino	Roteador seguinte	Núm. saltos até dest.
w	A	2
y	B	2
z	B	7
x	--	1
….	….	….

tabela de roteamento/repasse em D

RIP

Destino	Próx.	saltos
w	-	1
x	-	1
z	C	4
….	…	...

anúncio de A para D



Rede de destino até dest.	Roteador seguinte	Núm. saltos
w	A	2
y	B	2
z	~~B~~ A	~~7~~ 5
x	--	1
….	….	....

tabela de roteamento/repasse em D

O S P F

- “open”: publicamente disponível
- usa algoritmo Link State
  - disseminação de pacote LS
  - mapa de topologia em cada nó
  - cálculo de rota usando algoritmo de Dijkstra
- anúncio OSPF transporta uma entrada por roteador vizinho
- anúncios disseminados ao AS inteiro (com inundação)

Roteamento hierárquico

Nosso estudo de roteamento até aqui – o ideal:

- todos os roteadores idênticos
- rede "achatada"

… *não* acontece na prática

escala: com 200 milhões de destinos:

- não pode armazenar todos os destinos nas tabelas de roteamento!
- troca de tabela de roteamento atolaria os enlaces!

autonomia administrativa

- Internet = rede de redes
- cada administrador de rede pode querer controlar o roteamento em sua própria rede

Roteamento hierárquico

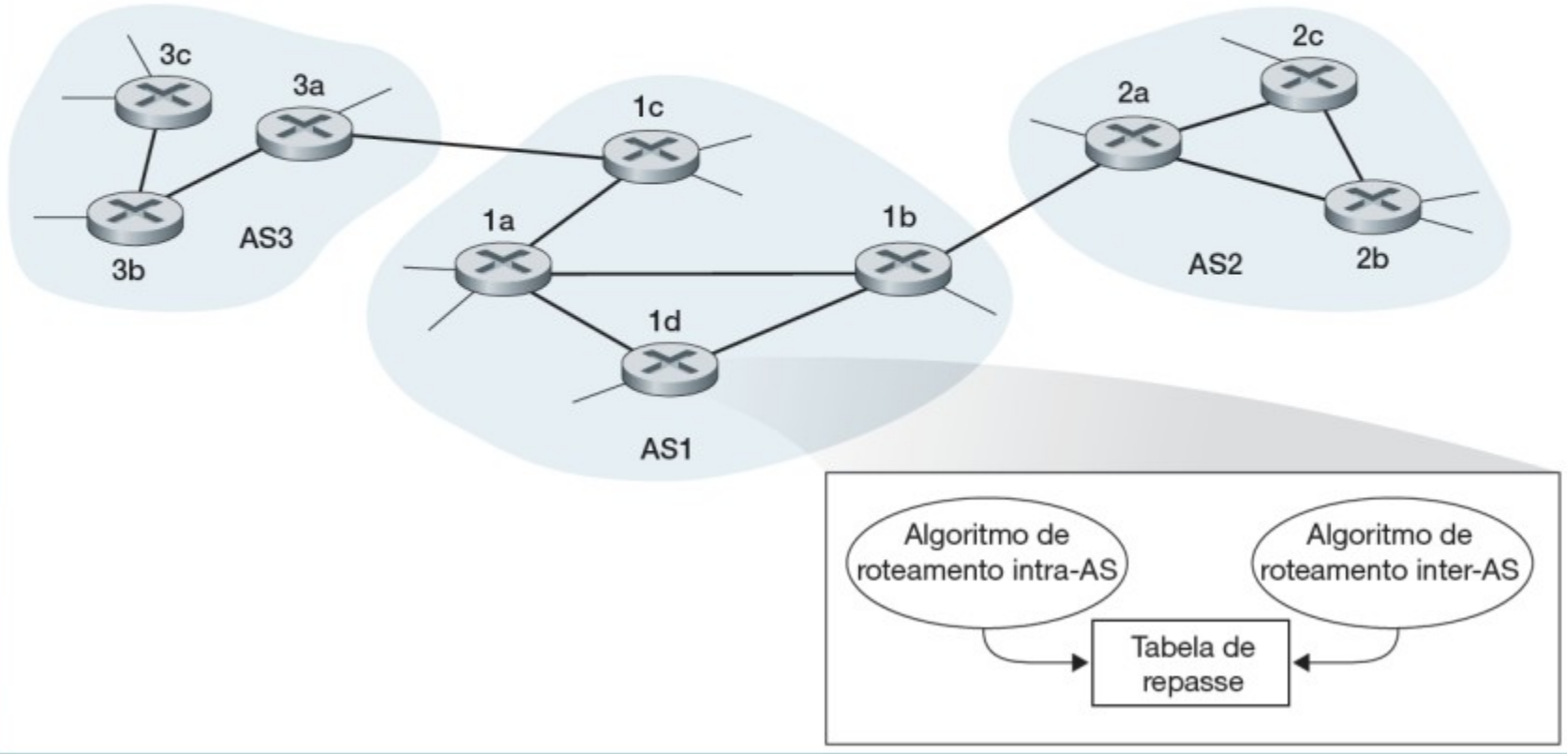
- Roteadores agregados em regiões, "sistemas autônomos" (AS)

- Roteadores no mesmo AS rodam o mesmo protocolo de roteamento
  - protocolo de roteamento "intra-AS"
  - roteadores em ASes diferentes podem executar protocolo de roteamento intra-AS diferente

Roteador de borda

- Enlace direto com roteador em outro AS

Exemplo de sistemas autônomos interconectados



Roteamento intra-AS na Internet

- Um protocolo de roteamento intra-AS é usado para determinar como é rodado o roteamento dentro de um sistema autônomo (AS).

- Historicamente, dois protocolos de roteamento têm sido usados para roteamento dentro de um sistema autônomo na Internet:

1. o protocolo de informações de roteamento, **RIP** (Routing Information Protocol) e

2. o **OSPF** (Open Shortest Path First).

PERGUNTAS ?