

Capítulo 4: Camada de rede

Objetivos do capítulo:

- entender os princípios por trás dos serviços da camada de rede:
 - modelos de serviço da camada de rede
 - repasse *versus* roteamento
 - como funciona um roteador
 - roteamento (seleção de caminho)
 - lidando com escala
 - tópicos avançados: IPv6, mobilidade
- instanciação, implementação na Internet

Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

Redes de datagrama

- sem estabelecimento de chamada na camada de rede
- roteadores: sem estado sobre conexões fim a fim
 - sem conceito em nível de rede da "conexão"
- pacotes repassados usando endereço do hospedeiro de destino
 - pacotes entre mesmo par origem-destino podem tomar caminhos diferentes

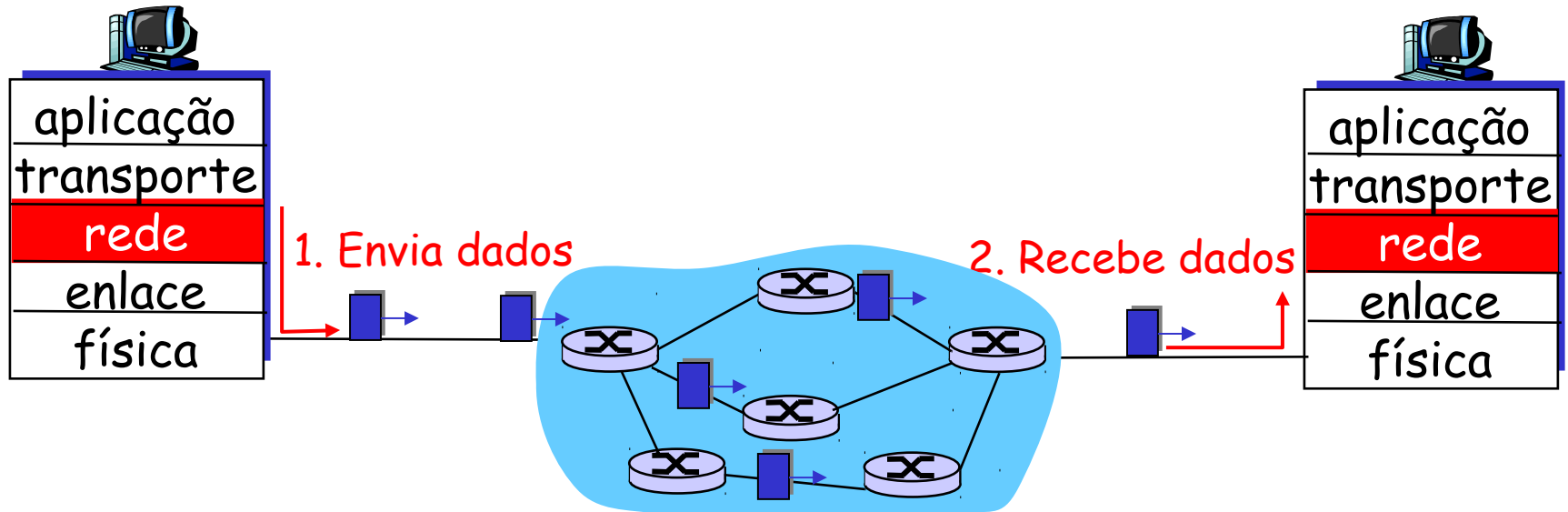


Tabela de repasse

4 bilhões de entradas
possíveis

Faixa de endereços de destino

Interface de enlace

11001000 00010111 00010000 00000000

até

0

11001000 00010111 00010111 11111111

11001000 00010111 00011000 00000000

até

1

11001000 00010111 00011000 11111111

11001000 00010111 00011001 00000000

até

2

11001000 00010111 00011111 11111111

senão

3

Concordância do prefixo mais longo

<u>Concordância do prefixo</u>	<u>Interface do enlace</u>
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
senão (default)	3

Prefixos e Máscaras

- Tamanho do prefixo determinado por IP + máscara
- IP = 200.1.1.1 (11001000.00000001.00000001.00000001)
 - Máscara: 255.255.255.0/24 → 11001000.00000001.00000001.00000000
 - Máscara: 255.255.0.0/16 → 11001000.00000001
 - Máscara: 255.128.0.0/9 → 11001000.0

Exemplos

11001000 00010111 00010110 10100001

Qual interface?

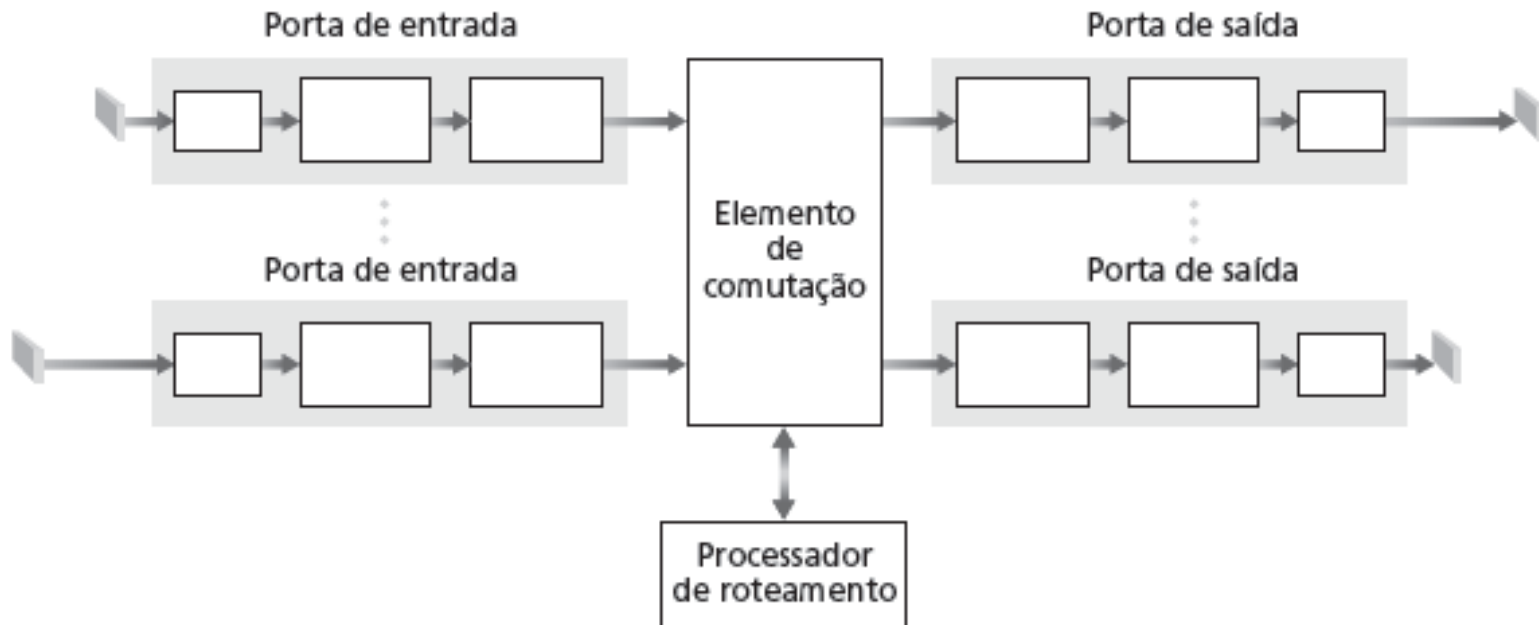
11001000 00010111 00011000 10101010

Qual interface?

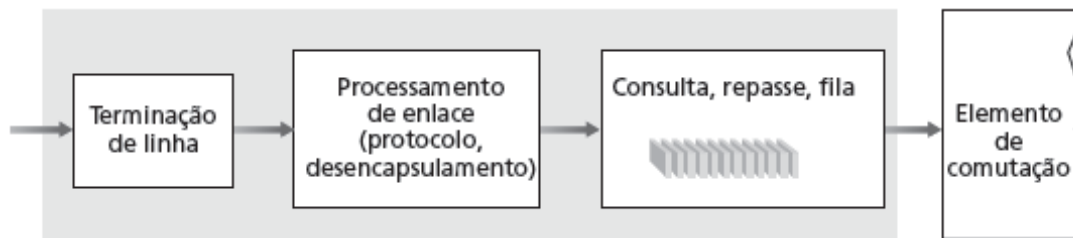
Visão geral da arquitetura do roteador

Duas funções principais do roteador:

- executar algoritmos/protocolo de roteamento (RIP, OSPF, BGP)
- *repassar* datagramas do enlace de entrada para saída



Funções da porta de entrada



Camada física:
recepção por bit

Camada de enlace
de dados:
p. e., Ethernet
ver Capítulo 5

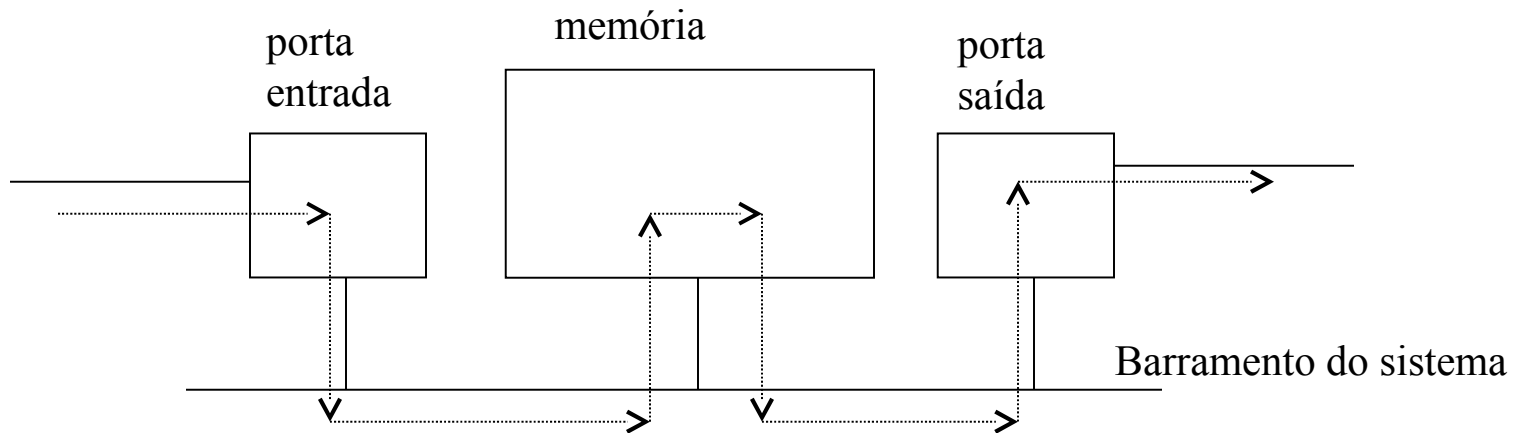
Comutação descentralizada:

- ❑ dado destino do datagrama, porta de saída de pesquisa usando tabela de repasse na memória da porta de entrada
- ❑ objetivo: processamento completo da porta de entrada na 'velocidade de linha'
- ❑ fila: se datagramas chegarem mais rápido que taxa de repasse no elemento de comutação

Comutação por memória

Roteadores de primeira geração:

- ❑ computadores tradicionais com a comutação via controle direto da CPU
- ❑ pacote copiado para a memória do sistema
- ❑ velocidade limitada pela largura de banda da memória (2 travessias de barramento por datagrama)



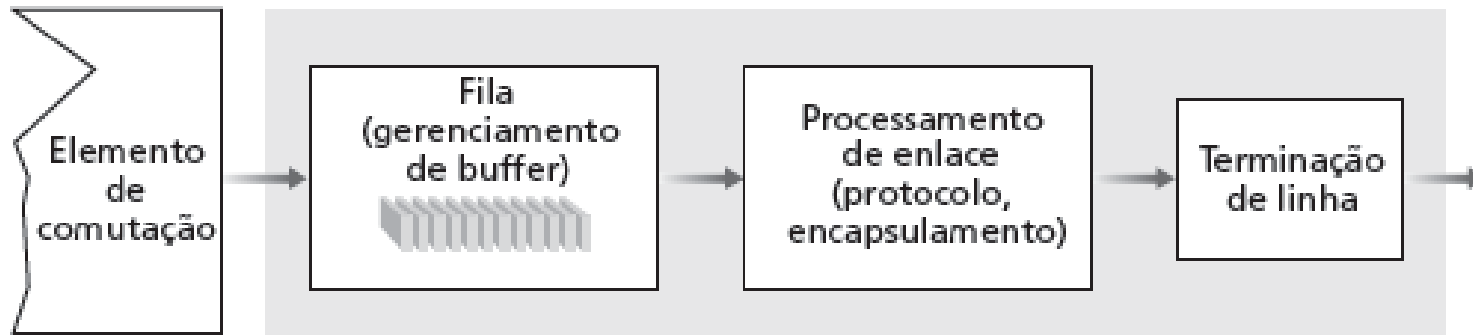
Comutação por um barramento

- ❑ datagrama da memória da porta de entrada à memória da porta de saída por um barramento compartilhado
- ❑ **disputa pelo barramento:** velocidade da comutação limitada pela largura de banda do barramento

Comutação por uma rede de interconexão

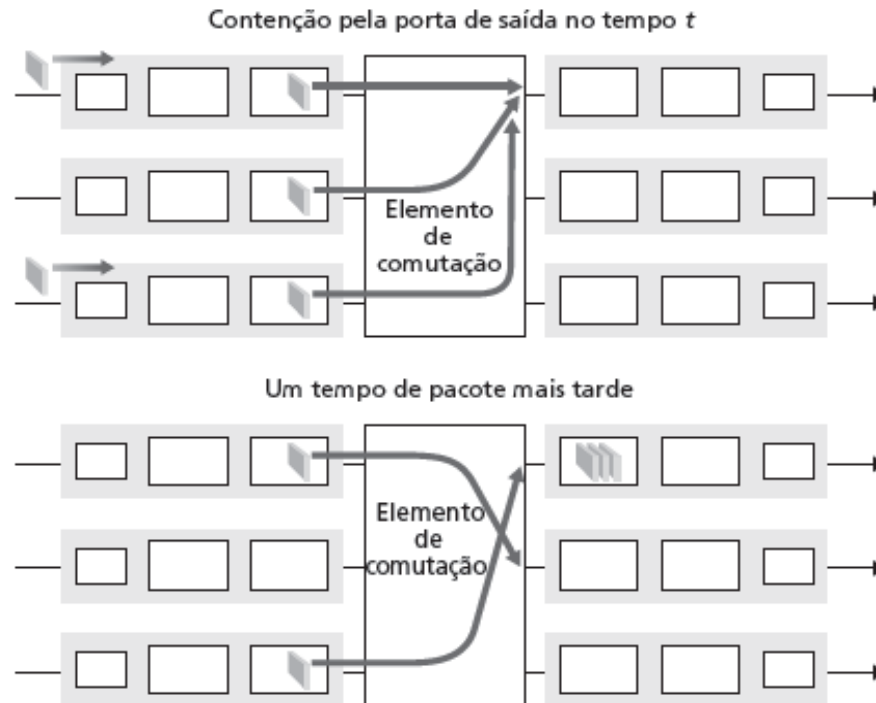
- ❑ contorna limitações de largura de banda do barramento
- ❑ redes de interconexão permite conexão entre mais de uma interface simultaneamente

Portas de saída



- *Buffering* exigido quando os datagramas chegam do elemento de comutação mais rápido que a taxa de transmissão
- *Disciplina de escalonamento* escolhe entre os datagramas enfileirados para transmissão

Enfileiramento na porta de saída



- buffering quando a taxa de chegada via comutador excede a velocidade da linha de saída
- *enfileiramento (atraso) e perda devidos a estouro de buffer na porta de saída!*

Exercícios Recomendados (1, 3, 4, 8,9,10,11)

Considere uma rede de datagramas que usa endereços de hospedeiros de 8 bits. Suponha que roteadores usem compatibilização com o prefixo mais longo e tenha as tabelas de repasse abaixo.

Para cada uma das quatro interfaces, forneça a faixa associada de endereços de hospedeiros de destino e o número de endereços na faixa para cada roteador.

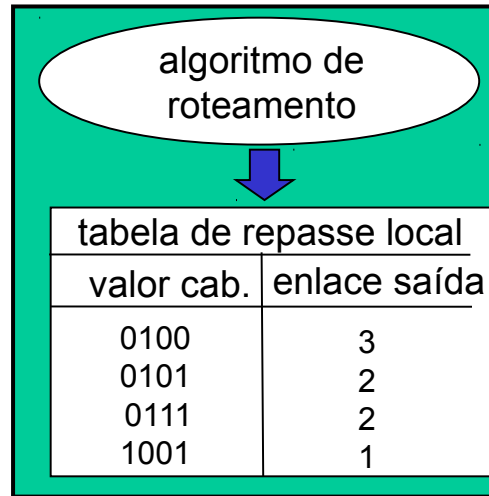
<u>Tabela Roteador A</u>	
Prefixo	Interface
00	0
010	1
011	2
10	2
11	3

<u>Tabela Roteador B</u>	
Prefixo	Interface
1	0
10	1
1110	2
Senão	3

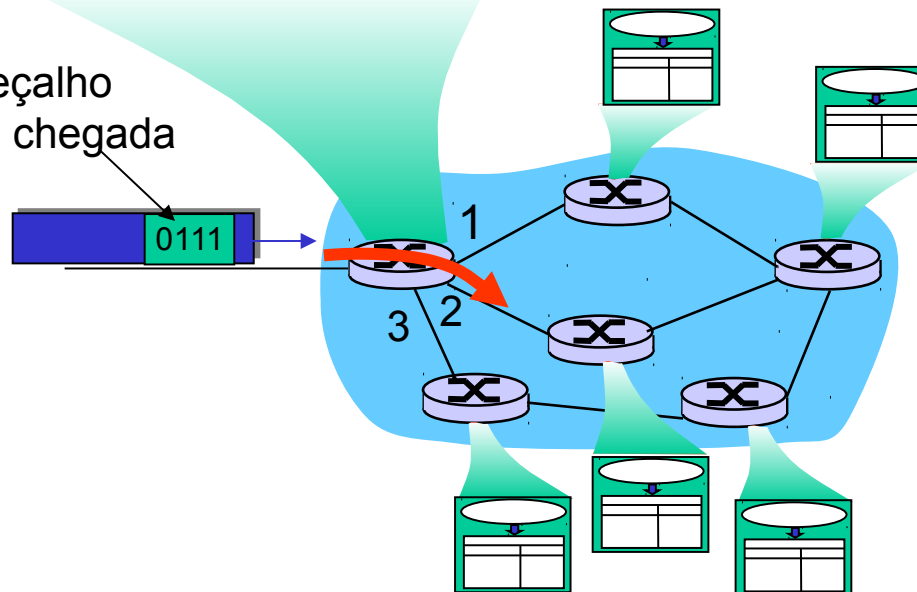
Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

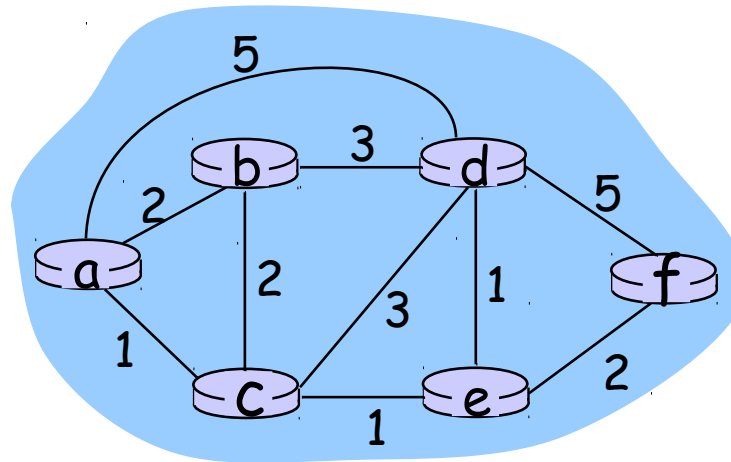
Interação entre roteamento e repasse



valor no cabeçalho do pacote de chegada



Abstração de grafo

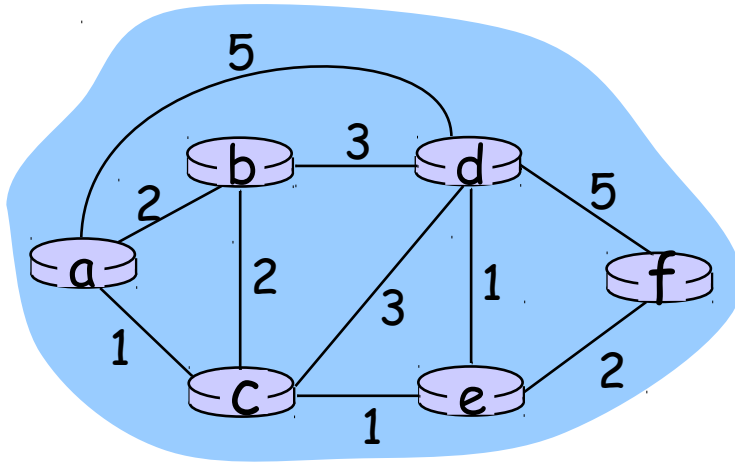


Grafo: $G = (N,E)$

$N =$ conjunto de roteadores = $\{ a, b, c, d, e, f \}$

$E =$ conjunto de enlaces = $\{ (a,b), (a,c), (a,d), (b,c), (b,d), (c,d), (c,e), (d,e), (d,f), (e,f) \}$

Abstração de grafo: custos



- $c(x,x')$ = custo do enlace (x,x')
 - p. e., $c(d,f) = 5$
- custo poderia ser sempre 1, ou inversamente relacionado à largura ou inversamente relacionado ao congestionamento

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

Pergunta: Qual é o caminho de menor custo entre a e f?

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

Classificação do algoritmo de roteamento

informação global ou descentralizada?

global:

- ❑ todos os roteadores têm topologia completa, informação de custo do enlace
- ❑ algoritmos de "estado do enlace" (Link State - LS)

descentralizada:

- ❑ roteador conhece vizinhos conectados fisicamente, custos de enlace para vizinhos
- ❑ processo de computação iterativo, troca de informações com vizinhos
- ❑ algoritmos de "vetor de distância" (Distance Vector - DV)

Estático ou dinâmico?

estático:

- ❑ rotas mudam lentamente com o tempo

dinâmico:

- ❑ rotas mudam mais rapidamente
 - atualização periódica
 - em resposta a mudanças no custo do enlace

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

Algoritmo de vetor de distância

Equação de Bellman-Ford (programação dinâmica)

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

onde min assume todos os vizinhos v de x

Algoritmo de Estado de Enlace

Algoritmo de Djikstra

$$d_x(y) = \min_v \{d_x(v) + c(v,y)\}$$

onde min assume todos os vizinhos v de y

Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

Algoritmo de roteamento de estado do enlace

algoritmo de Dijkstra

- topologia e 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**
- 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 $D(v) = \infty$

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' :

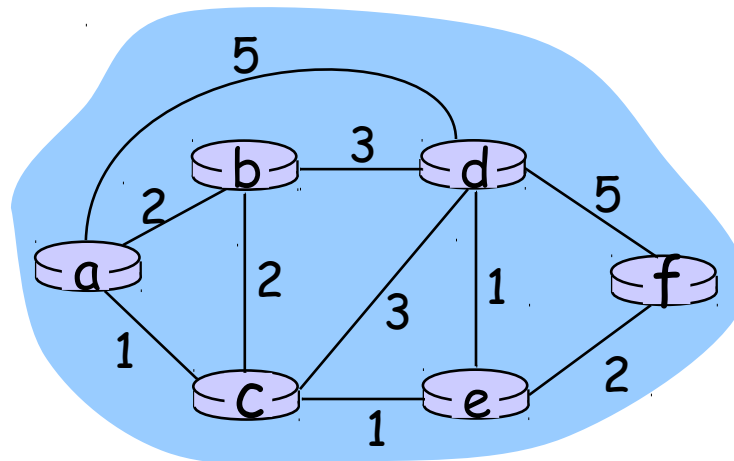
12 $D(v) = \min(D(v), D(w) + c(w,v))$

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'**

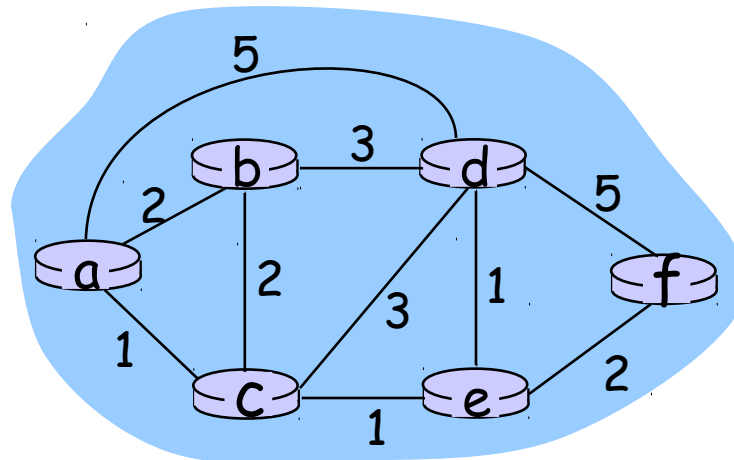
Algoritmo de Dijkstra: exemplo

Etapa	N'	D(b),p(b)	D(d),p(d)	D(c),p(c)	D(e),p(e)	D(f),p(f)
0	a	2,a	5,a	1,a	∞	∞
1	ac					
2	ace					
3	aceb					
4	acebd					
5	acebdf					



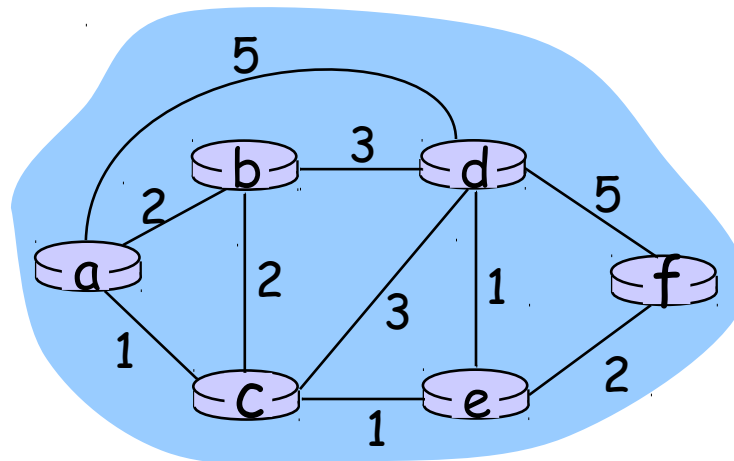
Algoritmo de Dijkstra: exemplo

Etapa	N'	D(b),p(b)	D(d),p(d)	D(c),p(c)	D(e),p(e)	D(f),p(f)
0	a	2,a	5,a	1,a	∞	∞
1	ac	2,a	4,c		2,c	∞
2	ace					
3	aceb					
4	acebd					
5	acebdf					



Algoritmo de Dijkstra: exemplo

Etapa	N'	D(b),p(b)	D(d),p(d)	D(c),p(c)	D(e),p(e)	D(f),p(f)
0	a	2,a	5,a	1,a	∞	∞
1	ac	2,a	4,c		2,c	∞
2	ace	2,a	3,e			4,e
3	aceb		3,e			4,e
4	acebd					4,e
5	acebdf					



Algoritmo de Dijkstra: exemplo (2)

árvore resultante do caminho mais curto a partir de a:

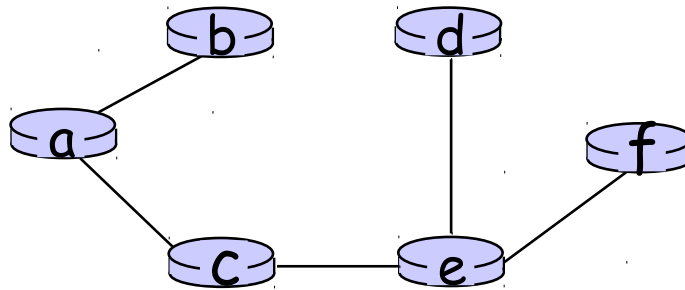


tabela de repasse resultante em a:

destino	enlace
b	(a,b)
c	(a,c)
d	(a,c)
e	(a,c)
f	(a,c)

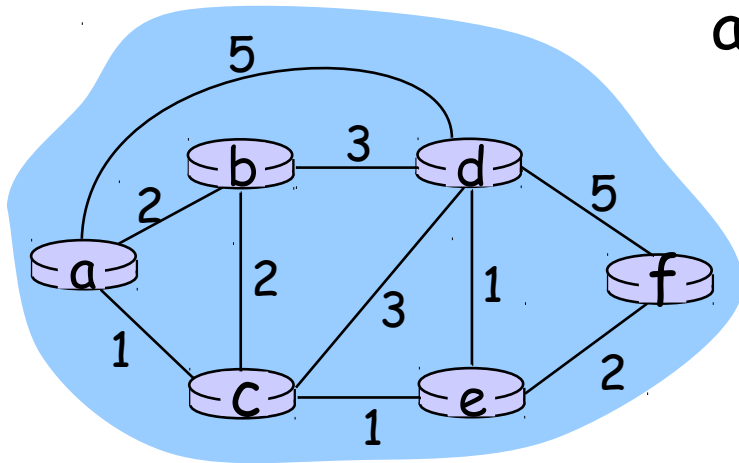
Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

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 $D_x = [D_x(y): y \in N]$
- nó x também mantém vetor de distância de seus vizinhos
 - para cada vizinho v , x mantém $D_v = [D_v(y): y \in N]$

Equação de Bellman-Ford (programação dinâmica)



assuma, $d_b(f) = 5$, $d_c(f) = 3$, $d_d(f) = 3$

equação B-F diz:

$$\begin{aligned}d_a(f) &= \min \{ c(a,b) + d_b(f), \\ &\quad c(a,c) + d_c(f), \\ &\quad c(a,d) + d_d(f) \} \\ &= \min \{ 2 + 5, \\ &\quad 1 + 3, \\ &\quad 5 + 3 \} = 4\end{aligned}$$

nó que alcança mínimo é o próximo salto
no caminho mais curto → tabela de repasse

Algoritmo de vetor de distância

ideia básica:

- de tempos em tempos, cada nó envia sua própria estimativa de vetor de distância aos vizinhos
- assíncrono
- quando um nó x recebe nova estimativa DV do vizinho, ele atualiza seu próprio DV usando a equação de B-F:

$$D_x(y) \leftarrow \min_v \{c(x,v) + D_v(y)\} \quad \text{para cada nó } y \in N$$

- sob condições modestas, naturais, a estimativa $D_x(y)$ converge para o menor custo real $d_x(y)$

Algoritmo de vetor de distância

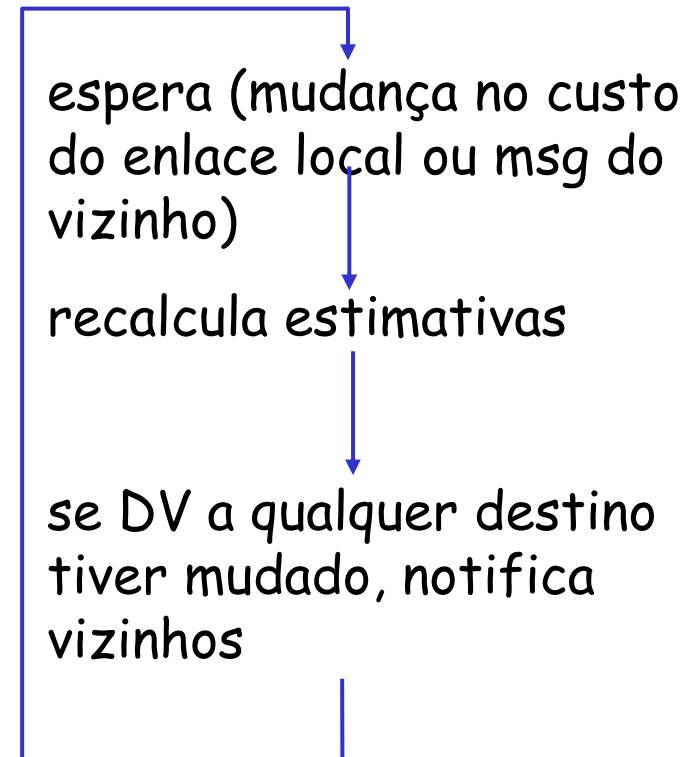
iterativo, assíncrono: cada iteração local causada por:

- ❑ mudança de custo do enlace local
- ❑ mensagem de atualização do DV do vizinho

distribuído:

- ❑ cada nó notifica os vizinhos *apenas* quando seu DV muda
 - nós, então, notificam seus vizinhos, se necessário

Cada nó:



$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$$

$$= \min\{2+0, 7+1\} = 2$$

tabela nó x

		custo para		
		x	y	z
de	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

		custo para		
		x	y	z
de	x	0	2	3
	y	2	0	1
	z	7	1	0

tabela nó y

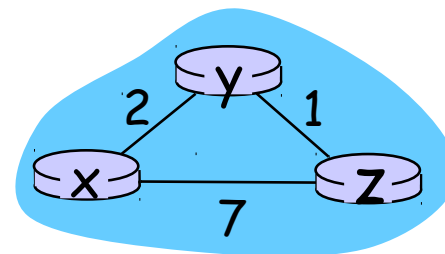
		custo para		
		x	y	z
de	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

tabela nó z

		custo para		
		x	y	z
de	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$$

$$= \min\{2+1, 7+0\} = 3$$



► tempo

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$$

$$= \min\{2+0, 7+1\} = 2$$

tabela nó x

		custo para		
		x	y	z
de	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

		custo para		
		x	y	z
de	x	0	2	3
	y	2	0	1
	z	7	1	0

		custo para		
		x	y	z
de	x	0	2	3
	y	2	0	1
	z	3	1	0

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$$

$$= \min\{2+1, 7+0\} = 3$$

tabela nó y

		custo para		
		x	y	z
de	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

		custo para		
		x	y	z
de	x	0	2	7
	y	2	0	1
	z	7	1	0

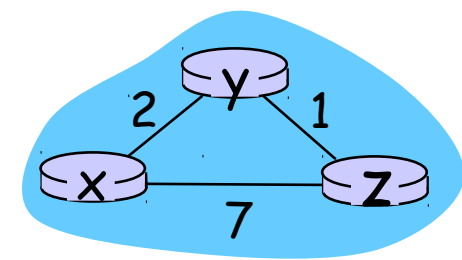
		custo para		
		x	y	z
de	x	0	2	3
	y	2	0	1
	z	3	1	0

tabela nó z

		custo para		
		x	y	z
de	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0

		custo para		
		x	y	z
de	x	0	2	7
	y	2	0	1
	z	3	1	0

		custo para		
		x	y	z
de	x	0	2	3
	y	2	0	1
	z	3	1	0

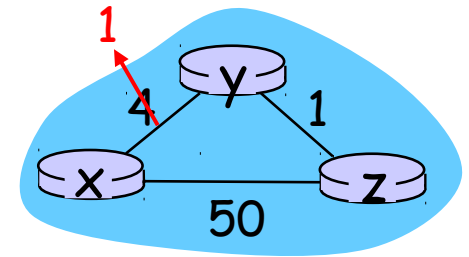


tempo →

Vetor de distância: mudanças de custo do enlace

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



“boas
notícias
correm
rápido”

no tempo t_0 , y detecta a mudança do custo do enlace, atualiza seu DV e informa aos seus vizinhos.

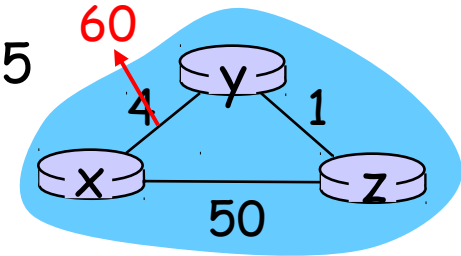
no tempo t_1 , z recebe a atualização de y e atualiza sua tabela. Calcula um novo custo mínimo para x e envia seu DV aos vizinhos.

no tempo t_2 , y recebe a atualização de z e atualiza sua tabela de distância. Menores custos de y não mudam, e daí y não envia qualquer mensagem a z .

Vetor de distância: mudanças de custo do enlace

mudanças de custo do enlace:

- ❑ antes da mudança temos: $D_y(x) = 4$ e $D_z(x) = 5$
- ❑ y detecta a mudança e atualiza $D_y(x)=6$ e repassa para z
- ❑ z atualiza $D_z(x)=7$ e repassa para y
- ❑ y atualiza $D_y(x)=8$ e repassa para z
- ❑



solução:

- ❑ ao repassar vetor de distância aos vizinhos, se utiliza o vizinho como rota, mude valor para infinito
 - y detecta a mudança e atualiza $D_y(x)=6$, mas repassa $D_y(x)=\infty$ para z
- ❑ z atualiza $D_z(x)=50$ e repassa para y
- ❑ y atualiza $D_y(x)=51$ e repassa $D_y(x)=\infty$ para z

Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

Roteamento hierárquico

nosso estudo de roteamento até aqui - o ideal:

- todos os roteadores idênticos
- rede "achatada"

... *não* acontece na prática

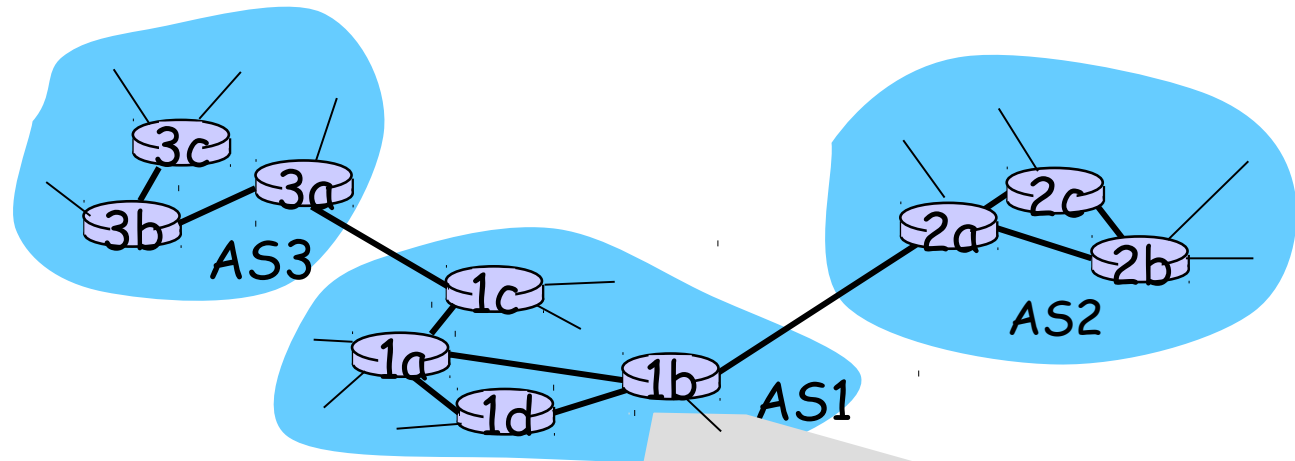
escala: com 200 milhões de destinos:

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

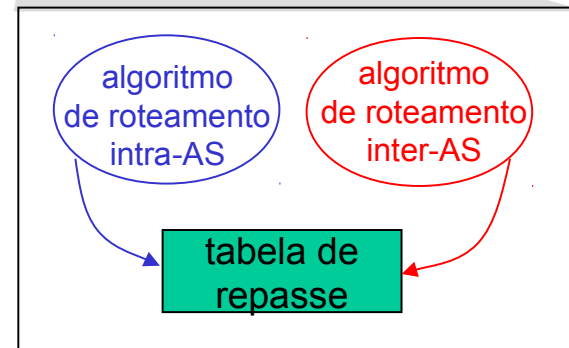
autonomia administrativa

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

Sistemas Autônomos interconectados

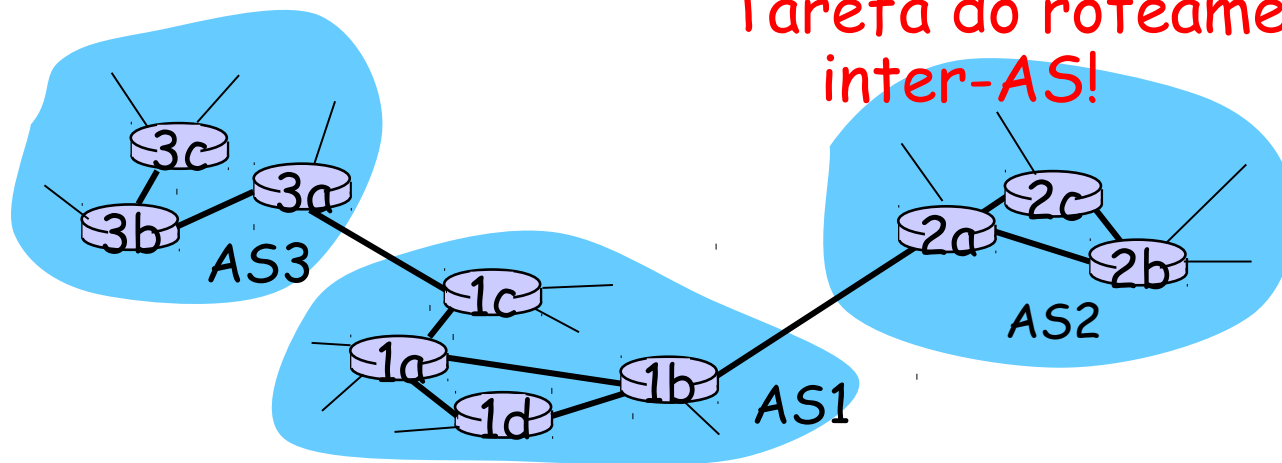


- tabela de repasse configurada por algoritmo de roteamento intra e inter-AS
 - intra-AS define entradas para destinos internos
 - inter-AS & intra-AS definem entradas para destinos externos



Tarefas inter-AS

- suponha que roteador no AS1 recebe datagrama destinado para fora do AS1:
 - roteador deve encaminhar pacote ao roteador de borda, mas qual?



AS1 deve:

1. descobrir quais destinos são alcançáveis por AS2 e quais por AS3
2. propagar essa informação de acessibilidade a todos os roteadores no AS1

Tarefa do roteamento inter-AS!

Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

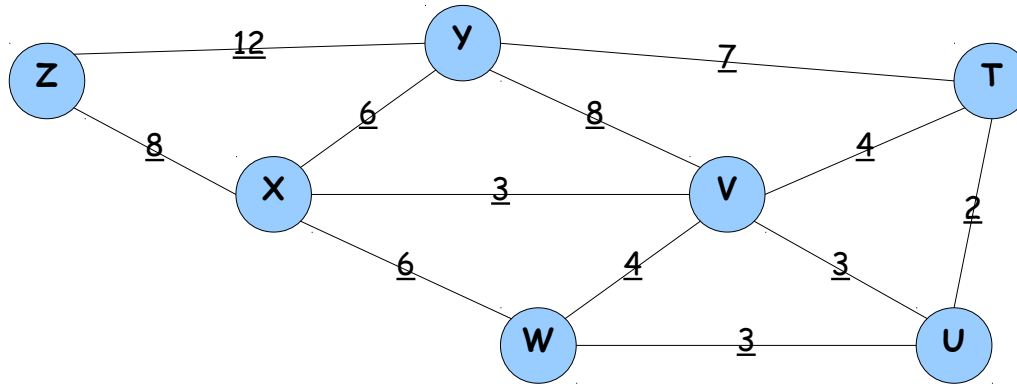
Roteamento intra-AS

- ❑ também conhecido como **Interior Gateway Protocols (IGP)**
- ❑ protocolos de roteamento intra-AS mais comuns:
 - RIP: Routing Information Protocol
 - OSPF: Open Shortest Path First
 - IGRP: Interior Gateway roteamento Protocol (proprietário da Cisco)

Roteamento inter-AS da Internet: BGP

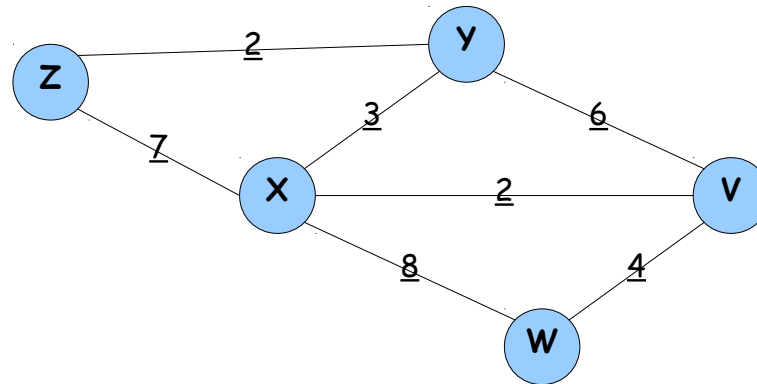
- ❑ **BGP (Border Gateway Protocol)**
- ❑ BGP oferece a cada AS um meio de:
 1. obter informação de acessibilidade da sub-rede a partir de ASs vizinhos.
 2. propagar informação de acessibilidade a todos os roteadores internos ao AS.
 3. determinar rotas "boas" para sub-redes com base na informação e política de acessibilidade.
- ❑ permite que a sub-rede anuncie sua existência ao resto da Internet: *"Estou aqui"*

Exercícios Recomendados (20, 21, 24, 25, 26, 27, 28, 29, 30, 31 e 32)



Considere a rede acima com os custos de enlace indicados.

- Use o algoritmo do caminho mais curto de Dijkstra para calcular o caminho mais curto entre cada par de nó.
- Use uma versão síncrona do algoritmo de vetor de distâncias para calcular o caminho mais curto entre cada par de nó. Suponha que inicialmente cada nó conheça apenas o custo para seus vizinhos imediatos.



Considere a rede acima com os custos de enlace indicados.

- Use uma versão síncrona do algoritmo de vetor de distâncias para calcular o caminho mais curto entre cada par de nó. Suponha que inicialmente cada nó conheça apenas o custo para seus vizinhos imediatos.
- Considerando o resultado do item anterior, use uma versão assíncrona do algoritmo de vetor de distâncias, onde os nós enviem suas tabelas imediatamente após realizar alguma atualização. Considere que o custo $c(z,x)$ seja alterado para 4.