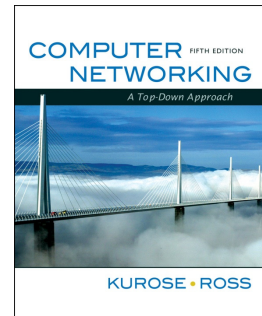


Capítulo 4 Camada de Rede



A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) in substantially unaltered form, that you mention their source (after all, we'd like people to use our book!)
- If you post any slides in substantially unaltered form on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

All material copyright 1996-2009
J.F Kurose and K.W. Ross, All Rights Reserved
Tradução para Português: Noélia Correia, UAIG, Portugal

*Computer Networking:
A Top Down Approach
5th edition.*

Jim Kurose, Keith Ross
Addison-Wesley, April
2009.

Camada de Rede 4-1

Capítulo 4: Camada de Rede

Objectivos:

- entender princípios por trás dos serviços da camada de rede:
 - modelos de serviço da camada de rede
 - *forwarding vs routing*
 - como funciona um *router*
 - *routing* (selecção de caminho)
 - lidar com escala
 - tópicos avançados: IPv6, mobilidade
- instanciação, implementação na Internet

Camada de Rede 4-2

Capítulo 4: Camada de Rede

- 4.1 **Introdução**
- 4.2 Circuitos virtuais e redes datagrama
- 4.3 O que existe dentro de um *router*
- 4.4 IP: Internet Protocol
 - Formato do datagrama
 - Endereçamento IPv4
 - ICMP
 - IPv6
- 4.5 Algoritmos de *routing*
 - Estado de ligação
 - Vector distância
 - *Routing* hierárquico
- 4.6 *Routing* na Internet
 - RIP
 - OSPF
 - BGP
- 4.7 *Routing broadcast e multicast*

Camada de Rede 4-3

Capítulo 4: Camada de Rede

- 4.1 Introdução
- 4.2 Circuitos virtuais e redes datagrama
- 4.3 O que existe dentro de um *router*
- 4.4 **IP: Internet Protocol**
 - Formato do datagrama
 - Endereçamento IPv4
 - **ICMP**
 - IPv6
- 4.5 Algoritmos de *routing*
 - Estado de ligação
 - Vector distância
 - *Routing* hierárquico
- 4.6 *Routing* na Internet
 - RIP
 - OSPF
 - BGP
- 4.7 *Routing broadcast e multicast*

Camada de Rede 4-63

ICMP: Internet Control Message Protocol

- ❑ usado por *hosts & routers* para trocarem informação da camada de rede
 - reportar erros: *host, rede, porto, protocolo* inacessível
 - *echo request/reply* (usado pelo ping)
- ❑ camada de rede "acima" do IP:
 - Msgs ICMP transportadas em datagramas IP
- ❑ **Mensagem ICMP:** tipo, código mais primeiros 8 bytes do datagrama IP que causou erro

Tipo	Código	descrição
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Camada de Rede 4-64

Traceroute e ICMP

- ❑ Fonte envia série de segmentos UDP para destino
 - Primeiro tem TTL =1
 - Secundo tem TTL=2, etc.
 - Número de porto improvável
- ❑ Quando n-ésimo datagrama chega ao n-ésimo *router*:
 - *Router* descarta datagrama
 - Envia para fonte uma mensagem ICMP (tipo 11, code 0)
 - Mensagem inclui nome do *router* & endereço IP
- ❑ Quando mensagem ICMP chega, fonte calcula RTT
- ❑ Traceroute faz isto 3 vezes
- Critério de paragem
- ❑ Segmento UDP eventualmente chega ao *host* destino
- ❑ Destino retorna pacote ICMP "*dest port unreachable*" (tipo 3, código 3)
- ❑ Quando fonte recebe este ICMP, pára.

Camada de Rede 4-65

Capítulo 4: Camada de Rede

- 4.1 Introdução
- 4.2 Circuitos virtuais e redes datagrama
- 4.3 O que existe dentro de um *router*
- 4.4 IP: Internet Protocol
 - Formato do datagrama
 - Endereçamento IPv4
 - ICMP
 - IPv6
- 4.5 Algoritmos de *routing*
 - Estado de ligação
 - Vector distância
 - *Routing* hierárquico
- 4.6 *Routing* na Internet
 - RIP
 - OSPF
 - BGP
- 4.7 *Routing broadcast e multicast*

Camada de Rede 4-66

IPv6

- **Motivação inicial:** espaço de endereços de 32-bits rapidamente ficará esgotado.
- **Motivação adicional:**
 - formato *header* ajuda a acelerar processamento/*forwarding*
 - alterações no *header* para facilitar QoS
- Formato datagrama IPv6:**
 - *header* de tamanho fixo com 40 bytes
 - não é permitida fragmentação

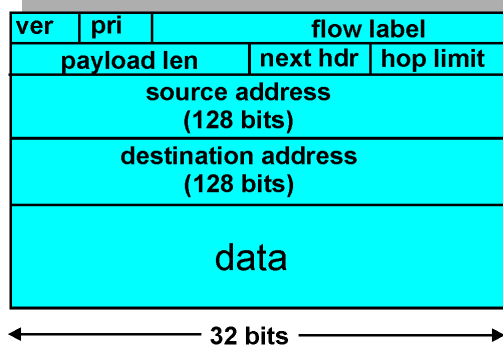
Camada de Rede 4-67

Header IPv6 (cont)

Priority: identifica prioridade entre datagrams no fluxo

Flow Label: identifica datagramas no mesmo "fluxo."
(conceito de "fluxo" não está definido).

Next header: identifica protocolo da camada acima



Camada de Rede 4-68

Outras diferenças relativamente ao IPv4

- ❑ *Checksum*: retirado para diminuição do tempo de processamento em cada salto.
- ❑ *Options*: permitidas, mas fora do *header*, indicatado por campo "Next Header"
- ❑ *ICMPv6*: nova versão do ICMP
 - tipos de mensagens adicionais, e.g. "Packet Too Big"
 - funções para gestão de grupos *multicast*

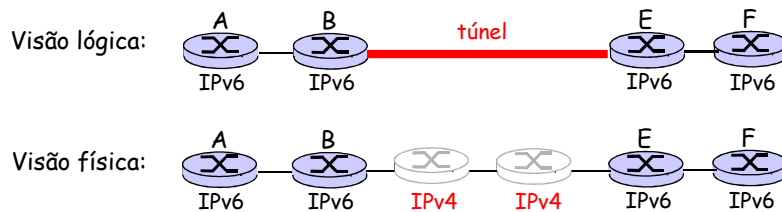
Camada de Rede 4-69

Transição de IPv4 para IPv6

- ❑ Não se consegue actualizar todos os *routers* simultaneamente
 - não existem "*flag days*"
 - Como irá operar a rede com routers IPv4 e IPv6 misturados?
- ❑ **Tunneling**: IPv6 transportado como *payload* em datagrama IPv4 entre routers IPv4

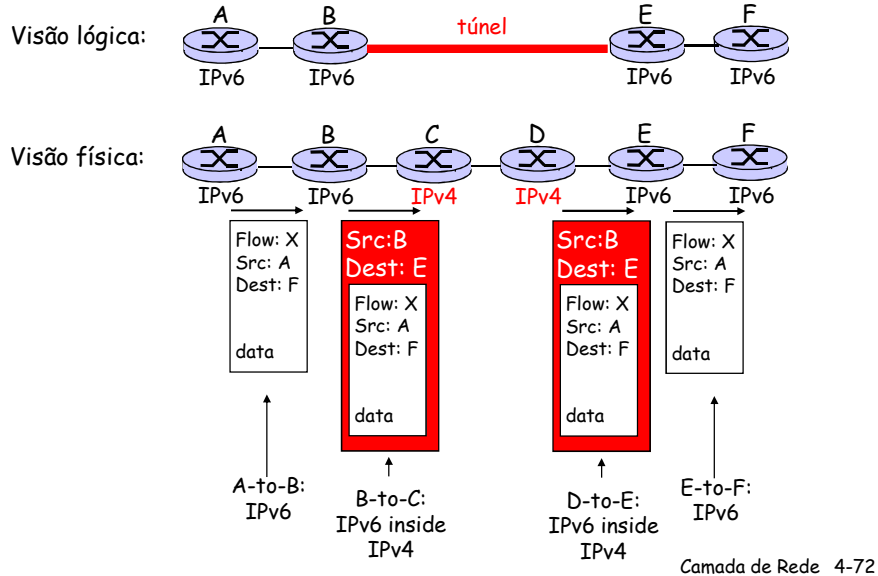
Camada de Rede 4-70

Tunneling



Camada de Rede 4-71

Tunneling

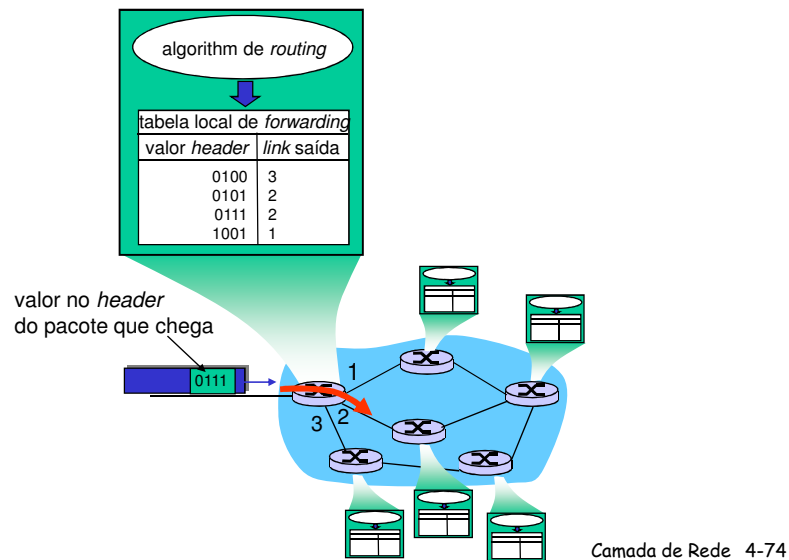


Capítulo 4: Camada de Rede

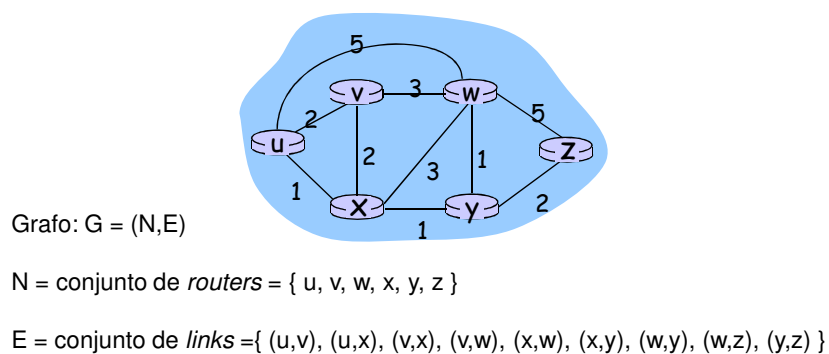
- 4.1 Introdução
- 4.2 Circuitos virtuais e redes datagrama
- 4.3 O que existe dentro de um *router*
- 4.4 IP: Internet Protocol
 - Formato do datagrama
 - Endereçamento IPv4
 - ICMP
 - IPv6
- 4.5 Algoritmos de *routing*
 - Estado de ligação
 - Vector distância
 - *Routing* hierárquico
- 4.6 *Routing* na Internet
 - RIP
 - OSPF
 - BGP
- 4.7 *Routing broadcast e multicast*

Camada de Rede 4-73

Interação entre *routing*, *forwarding*



Abstração usando grafos

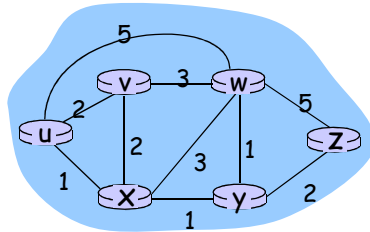


Nota: Abstração usando grafos é útil noutros contextos de rede

Exemplo: P2P, onde N é o conjunto de *peers* e E é o conjunto conexões TCP

Camada de Rede 4-75

Abstração usando grafos: custos



• $c(x, x')$ = custo do *link* (x, x')

- ex: $c(w, z) = 5$

• custo pode ser sempre 1, pode estar relacionado inversamente com largura de banda, ou pode estar relacionado de forma directa com congestionamento

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

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

Algoritmo de *routing*: algoritmo que encontra caminho de menor custo

Camada de Rede 4-76

Classificação dos algoritmos de *routing*

Informação global ou descentralizada?

Global:

- todos os *routers* têm informação completa relativa a topologia, custo de *links*
- algoritmos "estado de ligação" (*link state*)

Descentralizada:

- *router* conhece vizinhos fisicamente ligados, custos de *links* para vizinhos
- processo iterativo de cálculo, troca de info com vizinhos
- algoritmos "vector distância" (*distance vector*)

Estático ou dinâmico?

Estático:

- caminhos mudam pouco ao longo do tempo

Dinâmico:

- Caminhos mudam mais rapidamente
 - actualização periódica
 - em resposta a alterações nos custos dos *links*

Camada de Rede 4-77

Capítulo 4: Camada de Rede

- 4.1 Introdução
- 4.2 Circuitos virtuais e redes datagrama
- 4.3 O que existe dentro de um *router*
- 4.4 IP: Internet Protocol
 - Formato do datagrama
 - Endereçamento IPv4
 - ICMP
 - IPv6
- 4.5 Algoritmos de *routing*
 - Estado de ligação
 - Vector distância
 - *Routing* hierárquico
- 4.6 *Routing* na Internet
 - RIP
 - OSPF
 - BGP
- 4.7 *Routing broadcast e multicast*

Camada de Rede 4-78

Um algoritmo "estado de ligação"

Algoritmo de Dijkstra

- topologia de rede, custos de *links* conhecidos por todos os nós
 - conseguido através de "*broadcasts*" do estado dos *links*
 - nós têm todos mesma info
- calcula caminhos de menor custo de um nó fonte (*source*) para todos os outros nós
 - dá a *forwarding table* para esse nó
- iterativo: após k iterações, conhece caminho de menor custo para k destinos

Notação:

- $c(x,y)$: custo do *link* do nó x para y; $= \infty$ se não são vizinhos
- $D(v)$: valor actual do custo do caminho desde a fonte até v
- $p(v)$: nó antecessor no caminho desde a fonte até v
- N' : conjunto de nós que o caminho de menor custo já conhece definitivamente

Camada de Rede 4-79

Algoritmo de Dijkstra

```

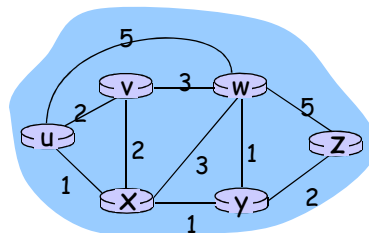
1 Initialization:
2  $N' = \{u\}$ 
3 for all nodes  $v$ 
4   if  $v$  adjacent to  $u$ 
5     then  $D(v) = c(u,v)$ 
6   else  $D(v) = \infty$ 
7
8 Loop
9   find  $w$  not in  $N'$  such that  $D(w)$  is a minimum
10  add  $w$  to  $N'$ 
11  update  $D(v)$  for all  $v$  adjacent to  $w$  and not in  $N'$  :
12     $D(v) = \min( D(v), D(w) + c(w,v) )$ 
13    /* new cost to  $v$  is either old cost to  $v$  or known
14       shortest path cost to  $w$  plus cost from  $w$  to  $v$  */
15 until all nodes in  $N'$ 

```

Camada de Rede 4-80

Algoritmo de Dijkstra: exemplo

Passo	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	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



Camada de Rede 4-81

Algoritmo de Dijkstra: exemplo (2)

Árvore de caminhos mais curtos com origem em u

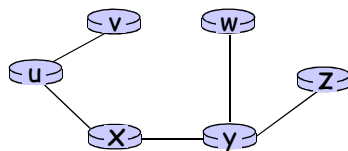


Tabela de forwarding em u:

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

Camada de Rede 4-82

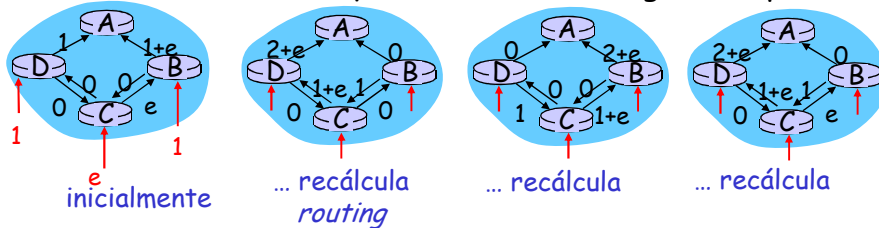
Algoritmo de Dijkstra, discussão

Complexidade algorítmica: n nós

- Cada iteração: necessita de verificar todos os nós, w , que não estão em N
- $n(n+1)/2$ comparações: $O(n^2)$
- existem implementações mais eficientes: $O(n \log n)$

Podem ocorrer oscilações:

- ex: custo do *link* = quantidade de tráfego transportado



Camada de Rede 4-83

Capítulo 4: Camada de Rede

- 4.1 Introdução
- 4.2 Circuitos virtuais e redes datagrama
- 4.3 O que existe dentro de um *router*
- 4.4 IP: Internet Protocol
 - Formato do datagrama
 - Endereçamento IPv4
 - ICMP
 - IPv6
- 4.5 Algoritmos de *routing*
 - Estado de ligação
 - **Vector distância**
 - *Routing* hierárquico
- 4.6 *Routing* na Internet
 - RIP
 - OSPF
 - BGP
- 4.7 *Routing broadcast e multicast*

Camada de Rede 4-84

Algoritmo Vector Distância

Equação Bellman-Ford (programação dinâmica)

Definir

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

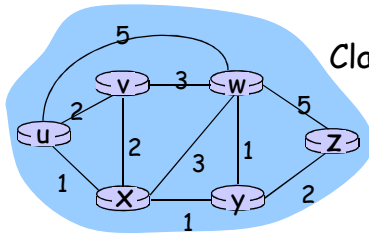
Então

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

onde min incide sobre todos os vizinhos v de x

Camada de Rede 4-85

Exemplo Bellman-Ford



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

equação B-F diz:

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

O nó que obtém mínimo é o próximo salto no caminho mais curto → tabela de *forwarding*

Camada de Rede 4-86

Algoritmo Vector Distância

- $D_x(y)$ = estimativa de menor custo de x para y
- Nó x sabe custo para cada vizinho v: $c(x,v)$
- Nó x mantém vector distância $D_x = [D_x(y): y \in N]$
- Nó x também mantém os vectores de distância dos seus vizinhos
 - Para cada vizinho v, x mantém $D_v = [D_v(y): y \in N]$

Camada de Rede 4-87

Algoritmo Vector Distância (4)

Ideia base:

- De tempos a tempos, cada nó envia o seu vector distância estimado aos seus vizinhos
- Assíncrono
- Quando um nó x recebe novo vector distância (DV) estimado de vizinho, actualiza a sua DV usando a equação B-F:
$$D_x(y) \leftarrow \min_v \{c(x,v) + D_v(y)\} \quad \text{para cada nó } y \in N$$
- De forma natural a estimativa $D_x(y)$ converge para o verdadeiro menor valor $d_x(y)$

Camada de Rede 4-88

Algoritmo Vector Distância (5)

Iterativo, assíncrono: cada iteração local causada por:

- alteração do custo de um *link* local
- mensagem de actualização de DV vinda de vizinho

Distribuído:

- cada nó notifica apenas vizinhos quando sua DV muda
 - vizinhos notificam os seus vizinhos se necessário

Cada nó:

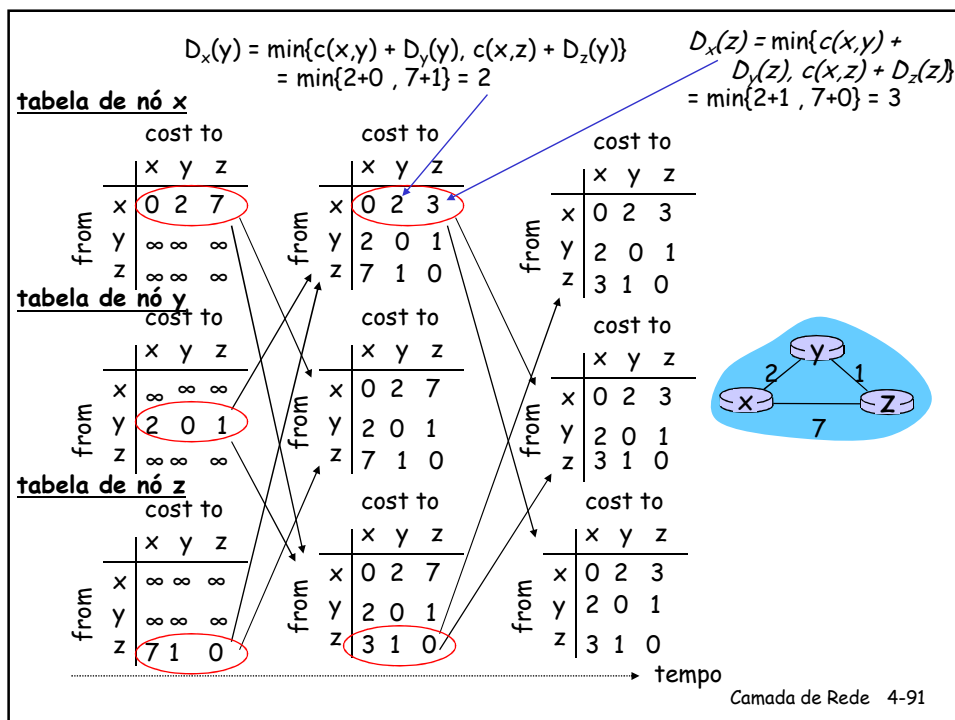
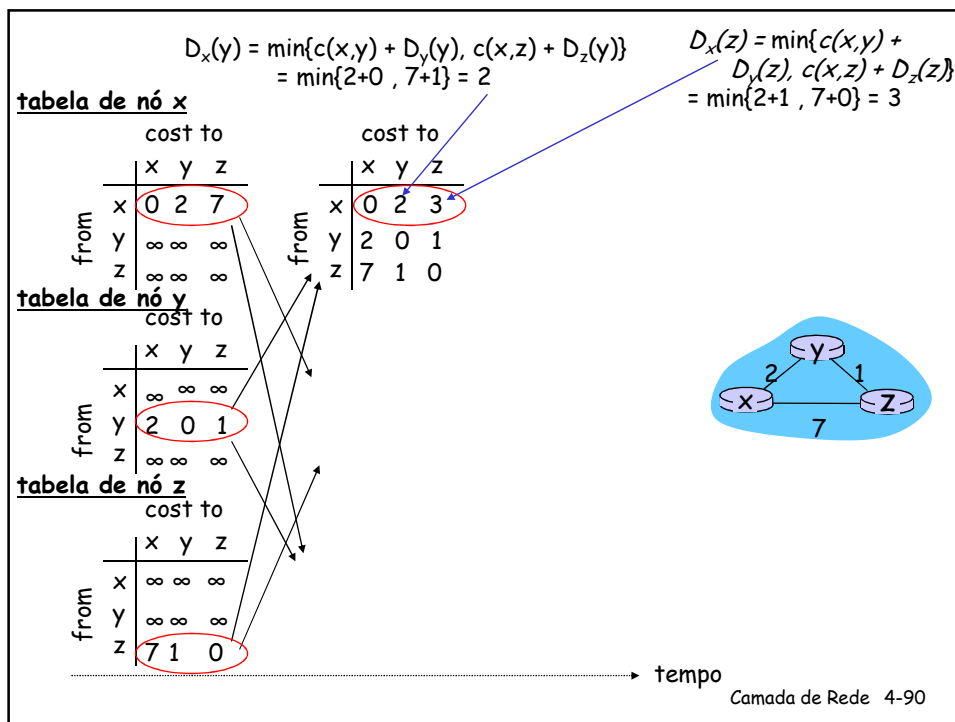
```
graph TD; A[Cada nó:] --> B[espera por (alteração do custo de um link local ou msg de vizinho)]; B --> C[recalcula estimativa]; C --> D[se DV para qualquer destino mudou, notifica vizinhos]; D --> A;
```

espera por (alteração do custo de um *link* local ou msg de vizinho)

recalcula estimativa

se DV para qualquer destino mudou, *notifica* vizinhos

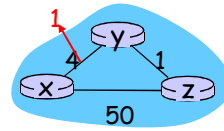
Camada de Rede 4-89



Vector Distância: alterações no custo de *link*

Alterações no custo de *link*:

- ❑ nó detecta alteração de custo de *link* local
- ❑ actualiza info *routing*, recalcula vector distância
- ❑ se DV muda, notifica vizinhos



"boas
notícias
viajam
depressa"

No instante t_0 , y detecta alteração no custo de *link*, actualiza o seu DV, e informa os vizinhos.

No instante t_1 , z recebe a actualização de y e actualiza a sua tabela. Calcula o novo menor custo para x e envia aos seus vizinhos o seu DV.

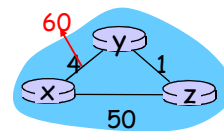
No instante t_2 , y recebe actualização de z e actualiza a sua tabela de distâncias. Os menores custos de y não mudam e, por isso, y não envia qualquer mensagem a z .

Camada de Rede 4-92

Vector Distância: alterações no custo de *link*

Alteração no custo de *link*:

- ❑ boas notícias viajam depressa
- ❑ más notícias viajam lentamente - problema da "contagem para infinito"!
- ❑ 44 iterações até que algoritmo estabilize: ver texto



Poisoned reverse (envenenamento inverso):

- ❑ Se Z encaminha através de Y para chegar até X :
 - Z diz a Y que a sua (de Z) distância até X é infinita (para que Y não encaminhe para X através de Z)
- ❑ será que isto resolve por completo o problema da contagem para infinito?

Camada de Rede 4-93

Comparação entre algoritmos LS e DV

Complexidade das mensagens

- **LS:** com n nós, E links, $O(nE)$ msgs enviadas
- **DV:** troca somente entre vizinhos
 - tempo de convergência varia

Velocidade de convergência

- **LS:** algoritmo $O(n^2)$ requer $O(nE)$ msgs
 - pode ter oscilações
- **DV:** tempo de convergência varia
 - podem existir *loops* no encaminhamento
 - problema da contagem para infinito

Robustez: o que acontece se o *router* funciona mal?

LS:

- nó pode anunciar custo incorrecto de um *link*
- cada nó calcula apenas a sua tabela

DV:

- Nó pode anunciar custo incorrecto de um *caminho*
- a tabela de cada nó é usada por outros
 - erro propaga-se através da rede

Camada de Rede 4-94

Capítulo 4: Camada de Rede

- 4.1 Introdução
- 4.2 Circuitos virtuais e redes datagrama
- 4.3 O que existe dentro de um *router*
- 4.4 IP: Internet Protocol
 - Formato do datagrama
 - Endereçamento IPv4
 - ICMP
 - IPv6
- 4.5 Algoritmos de *routing*
 - Estado de ligação
 - Vector distância
 - *Routing hierárquico*
- 4.6 *Routing* na Internet
 - RIP
 - OSPF
 - BGP
- 4.7 *Routing broadcast e multicast*

Camada de Rede 4-95

Routing hierárquico

Nosso estudo de *routing* até agora - idealização

- todos *routers* idênticos
- rede "*flat*"
- ... não é verdade na prática

escala: com 200 milhões de destinos:

- não é possível armazenar todos os destinos nas tabelas de *routing*!
- a troca de tabelas de *routing* iria entupir os *links*!

autonomia administrativa

- Internet = rede de redes
- cada administrador de rede poderá querer controlar o *routing* na sua rede

Camada de Rede 4-96

Routing hierárquico

- agregação de *routers* em regiões, "*autonomous systems*" (AS)

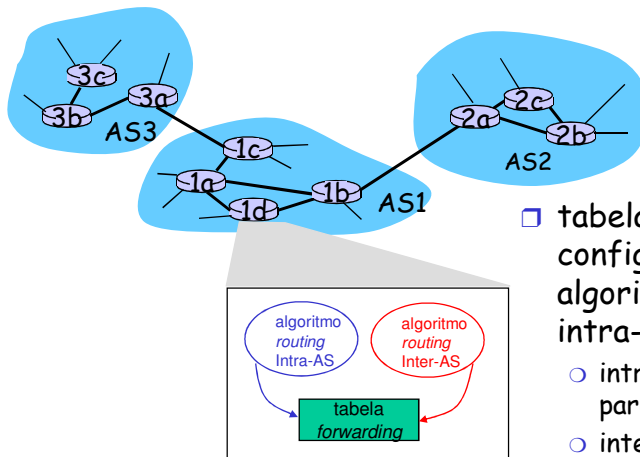
- *routers* no mesmo AS correm mesmo protocolo de *routing*
 - protocolo de *routing* intra-domínio (intra-AS)
 - *routers* em diferentes AS podem correr protocolos de *routing* intra-AS diferentes

Router gateway

- *Link* directo a *router* noutra AS

Camada de Rede 4-97

ASs interligados



- tabela de *forwarding* configurada por ambos algoritmos de *routing* intra- e inter-AS
 - intra-AS coloca entradas para destinos internos
 - inter-AS & intra-AS colocam entradas para destinos externos

Camada de Rede 4-98

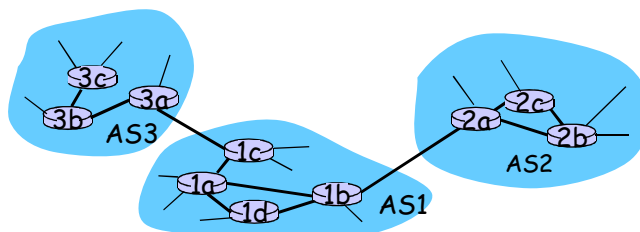
Tarefas inter-AS

- suponhamos que *router* em AS1 recebe datagrama com destino para fora de AS1:
 - *router* deverá fazer *forward* de pacote para *router gateway*, mas qual?

AS1 tem de:

1. saber que destinos podem ser acedidos através de AS2, e quais através de AS3
2. propagar esta informação de acessibilidade a todos os *routers* em AS1

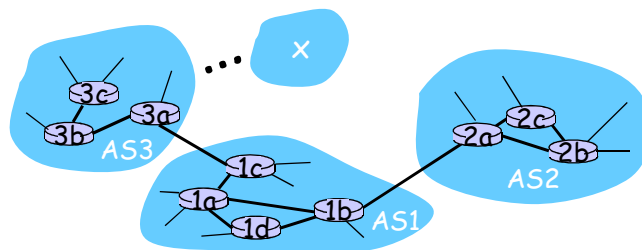
Tarefa do *routing* inter-AS!



Camada de Rede 4-99

Exemplo: Preenchimento da tabela de forwarding no router 1d

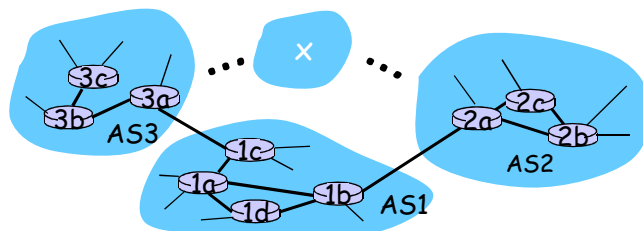
- suponhamos AS1 aprende (via protocolo inter-AS) que subrede x está acessível via AS3 (gateway 1c) mas não via AS2.
- protocolo inter-AS propaga informação de acessibilidade a todos os *routers* internos.
- *router* 1d determina através de informação de *routing* intra-AS que a sua interface I está no caminho de menor custo para 1c.
 - coloca entrada (x, I) na tabela de *forwarding*



Camada de Rede 4-100

Exemplo: Escolha entre múltiplos ASes

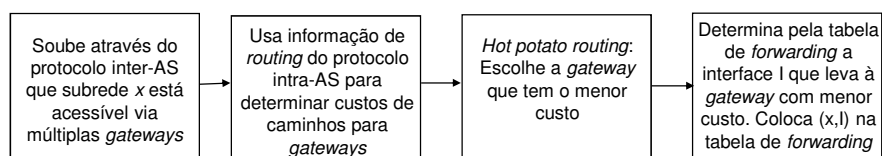
- suponhamos agora que AS1 aprende através do protocolo inter-AS que a subrede x está acessível através de AS3 e através de AS2.
- para configurar tabela de *forwarding*, *router* 1d tem de determinar para que *gateway* deverá fazer o *forward* dos packets destinados a x .
 - isto é também tarefa do protocolo de *routing* inter-AS!



Camada de Rede 4-101

Exemplo: Escolha entre múltiplos ASes

- suponhamos agora que AS1 aprende através do protocolo inter-AS que a subrede **x** está acessível através de AS3 e através de AS2.
- para configurar tabela de *forwarding*, *router* 1d tem de determinar para que *gateway* deverá fazer o *forward* dos packets destinados a **x**.
 - isto é também tarefa do protocolo de *routing* inter-AS!
- **hot potato routing**: enviar pacote para o *router* mais perto.



Camada de Rede 4-102

Capítulo 4: Camada de Rede

- 4.1 Introdução
- 4.2 Circuitos virtuais e redes datagrama
- 4.3 O que existe dentro de um *router*
- 4.4 IP: Internet Protocol
 - Formato do datagrama
 - Endereçamento IPv4
 - ICMP
 - IPv6
- 4.5 Algoritmos de *routing*
 - Estado de ligação
 - Vector distância
 - *Routing* hierárquico
- **4.6 *Routing* na Internet**
 - RIP
 - OSPF
 - BGP
- 4.7 *Routing broadcast e multicast*

Camada de Rede 4-103

Intra-AS Routing

- também chamados *Interior Gateway Protocols (IGP)*
- protocolos de *routing* Intra-AS mais comuns:
 - RIP: *Routing Information Protocol*
 - OSPF: *Open Shortest Path First*
 - IGRP: *Interior Gateway Routing Protocol* (propriedade Cisco)

Camada de Rede 4-104

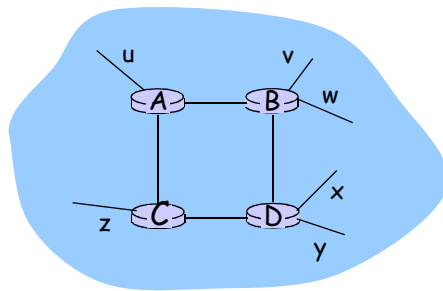
Capítulo 4: Camada de Rede

- | | |
|---|--|
| □ 4.1 Introdução | □ 4.5 Algoritmos de <i>routing</i> |
| □ 4.2 Circuitos virtuais e redes datagrama | ○ Estado de ligação |
| □ 4.3 O que existe dentro de um <i>router</i> | ○ Vector distância |
| □ 4.4 IP: Internet Protocol | ○ <i>Routing</i> hierárquico |
| ○ Formato do datagrama | □ 4.6 <i>Routing na Internet</i> |
| ○ Endereçamento IPv4 | ○ RIP |
| ○ ICMP | ○ OSPF |
| ○ IPv6 | ○ BGP |
| | □ 4.7 <i>Routing broadcast e multicast</i> |

Camada de Rede 4-105

RIP (Routing Information Protocol)

- ❑ algoritmo vector distância
- ❑ incluído na distribuição BSD-UNIX em 1982
- ❑ métrica de distância: # saltos (max = 15 saltos)



Do router A para subredes:

<u>destino</u>	<u>saltos</u>
u	1
v	2
w	2
x	3
y	3
z	2

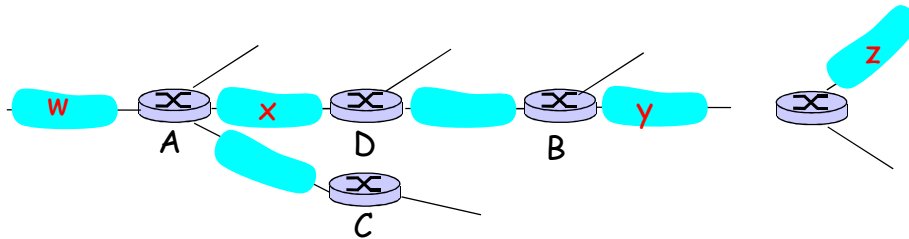
Camada de Rede 4-106

Anúncios RIP

- ❑ vectores distância: trocados entre vizinhos todos os 30 seg via *Response Message* (também chamado anúncio ou *advertisement*)
- ❑ cada anúncio: lista de até 25 subredes destino dentro do AS

Camada de Rede 4-107

RIP: Exemplo



Rede Destino	Próximo Router	Num. saltos até dest.
w	A	2
y	B	2
z	B	7
x	--	1
....		

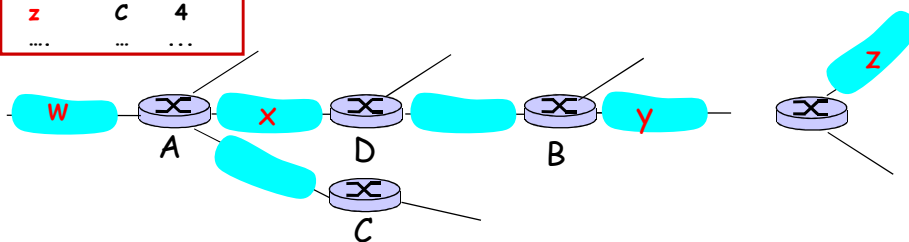
Tabela de *Routing/Forwarding* em D

Camada de Rede 4-108

RIP: Exemplo

Dest	Próx	Salto
w	-	1
x	-	1
z	C	4
....	...	...

Anúncio de
A para D



Rede Destino	Próximo Router	Num. saltos até dest.
w	A	2
y	B	2
z	B A	7 5
x	--	1
....		

Tabela *Routing/Forwarding* em D

Camada de Rede 4-109

RIP: Falha de *link* e recuperação

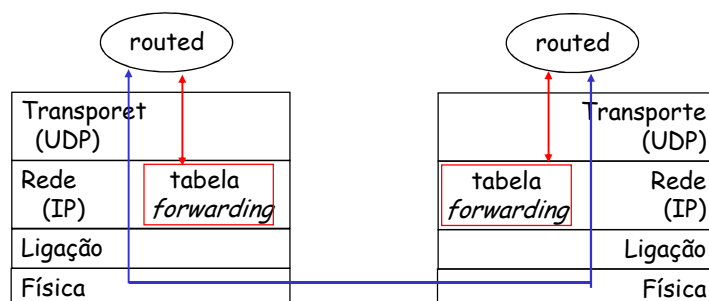
Se não é ouvido anúncio após 180 seg --> vizinho/*link* declarado morto

- caminhos via vizinho invalidados
- novos anúncios enviados a vizinhos
- vizinhos por sua vez enviam novos anúncios (se tabelas alteradas)
- informação de falha de *link* propaga-se rapidamente (?) a toda a rede
- *poison reverse* usado para prevenir *ping-pong loops* (distância infinita = 16 saltos)

Camada de Rede 4-110

RIP: Processamento de tabela

- Tabelas de *routing* RIP geridas por processo da **camada de aplicação** chamado route-d (daemon)
- anúncios (*advertisements*) enviados em pacotes UDP, repetidos periodicamente



Camada de Rede 4-111

Capítulo 4: Camada de Rede

- 4.1 Introdução
- 4.2 Circuitos virtuais e redes datagrama
- 4.3 O que existe dentro de um *router*
- 4.4 IP: Internet Protocol
 - Formato do datagrama
 - Endereçamento IPv4
 - ICMP
 - IPv6
- 4.5 Algoritmos de *routing*
 - Estado de ligação
 - Vector distância
 - *Routing* hierárquico
- 4.6 *Routing na Internet*
 - RIP
 - OSPF
 - BGP
- 4.7 *Routing broadcast e multicast*

Camada de Rede 4-112

OSPF (*Open Shortest Path First*)

- "open": disponível publicamente
- Usa algoritmo *Link State*
 - disseminação de pacotes LS
 - mapa da topologia em cada nó
 - cálculo do caminho usando algoritmo de Dijkstra
- Anúncio (*advertisement*) OSPF transporta uma entrada por *router* vizinho
- Anúncios (*advertisements*) disseminados para **todo** o AS (via *flooding*)
 - transportados em mensagens OSPF directamente sobre IP (em vez de TCP or UDP)

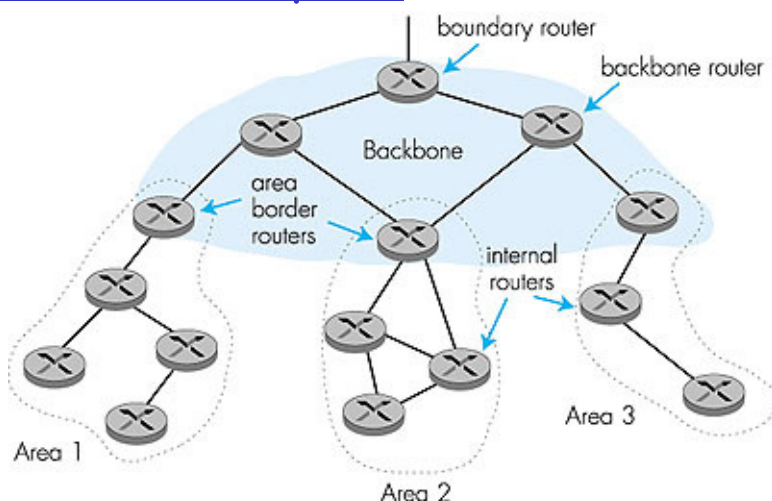
Camada de Rede 4-113

Aspectos "avançados" do OSPF (não existem no RIP)

- ❑ **segurança**: todas as mensagens OSPF são autenticadas (para prevenir intrusão maliciosa)
- ❑ permitidos múltiplos caminhos com mesmo custo (no RIP apenas um caminho)
- ❑ Para cada *link*, múltiplas métricas de custo para **TOS** diferentes (ex: custo de um *link* satélite colocado "baixo" para melhor esforço; "alto" para tempo real)
- ❑ suporte integrado para uni- e **multicast**:
 - OSPF (MOSPF) multicast usa mesma base de dados de topologia que o OSPF
- ❑ OSPF **hierárquico** em grandes domínios

Camada de Rede 4-114

OSPF hierárquico



Camada de Rede 4-115

OSPF hierárquico

- ❑ **hierarquia de dois níveis:** área local, núcleo da rede (*backbone*).
 - anúncios (*advertisements*) de estado dos *links* apenas na área
 - cada nó tem topologia detalhada da área; apenas sabe a direção (caminho mais curto) para redes noutras áreas.
- ❑ **area border routers:** "sumariam" distâncias até redes na sua área, anunciam para outros *area border routers*.
- ❑ **backbone routers:** usam *routing* OSPF apenas limitado ao núcleo da rede (*backbone*).
- ❑ **boundary routers:** ligam a outros AS's.

Camada de Rede 4-116

Capítulo 4: Camada de Rede

- | | |
|---|--|
| ❑ 4.1 Introdução | ❑ 4.5 Algoritmos de <i>routing</i> |
| ❑ 4.2 Circuitos virtuais e redes datagrama | ○ Estado de ligação |
| ❑ 4.3 O que existe dentro de um <i>router</i> | ○ Vector distância |
| ❑ 4.4 IP: Internet Protocol | ○ <i>Routing</i> hierárquico |
| ○ Formato do datagrama | ❑ 4.6 <i>Routing</i> na Internet |
| ○ Endereçamento IPv4 | ○ RIP |
| ○ ICMP | ○ OSPF |
| ○ IPv6 | ○ BGP |
| | ❑ 4.7 <i>Routing broadcast e multicast</i> |

Camada de Rede 4-117

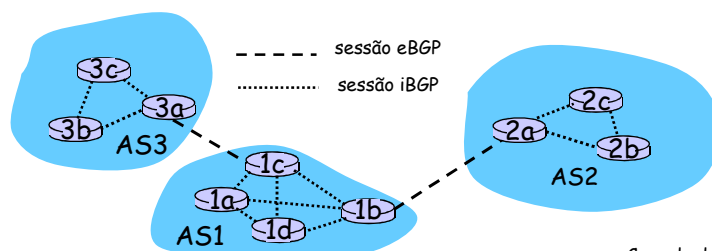
Routing inter-AS na Internet: BGP

- **BGP (*Border Gateway Protocol*)**: o standard *de facto*
- BGP fornece a cada AS uma forma de:
 1. Obter, de AS vizinhos, informação de como chegar a subredes.
 2. Propagar informação de acessibilidade para todos os *routers* internos ao AS.
 3. Determinar "bons" caminhos até subredes com base na informação de acessibilidade e política.
- permite anunciar a existência de uma subrede ao resto da Internet: "*I am here*"

Camada de Rede 4-118

BGP: básico

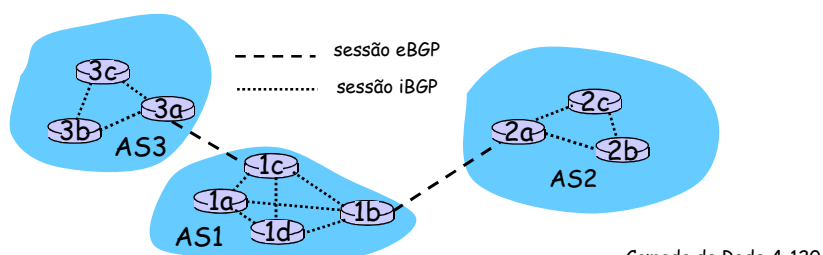
- pares de *routers* (*peers* BGP) trocam informação de encaminhamento usando conexões TCP semi-permanentes: **sessões BGP**
 - sessões BGP não necessitam de corresponder a *links* físicos.
- quando AS2 anuncia um prefixo a AS1:
 - AS2 **promete** que irá fazer o *forward* de datagramas em direção a esse prefixo.
 - AS2 pode agregar prefixos no anúncio



Camada de Rede 4-119

Distribuindo informação de acessibilidade

- ❑ usando sessão eBGP entre 3a e 1c, AS3 envia informação de acessibilidade de prefixo para AS1.
 - 1c pode então usar iBGP para distribuir nova informação de prefixo para todos os *routers* no AS1
 - 1b pode então re-anunciar nova informação de acessibilidade para AS2 através da sessão eBGP 1b-2a
- ❑ quando *router* sabe de novo prefixo, cria entrada para prefixo na sua tabela de *forwarding*.



Camada de Rede 4-120

Atributos de caminho & caminhos BGP

- ❑ prefixo anunciado inclui atributos BGP.
 - prefixo + atributos = "caminho"
- ❑ dois atributos importantes:
 - **AS-PATH**: contém ASs através dos quais o anúncio de prefixo passou: ex: AS 67, AS 17
 - **NEXT-HOP**: especifica a interface, do *router* interno ao AS corrente, usada para enviar anúncio até AS px salto (poderão existir múltiplos *links* do AS corrente até AS-próximo-salto)
- ❑ quando *router gateway* recebe anúncio de caminho, usa **política de importação** para aceitar/declinar.

Camada de Rede 4-121

BGP: seleção de caminho

- ❑ *router* pode aprender mais do que 1 caminho para mesmo prefixo. *Router* terá de seleccionar caminho.
- ❑ regras para eliminação:
 1. valor de atributo de preferência local: decisão política
 2. AS-PATH mais curto
 3. router NEXT-HOP mais próximo: *routing hot potato*
 4. critério adicional

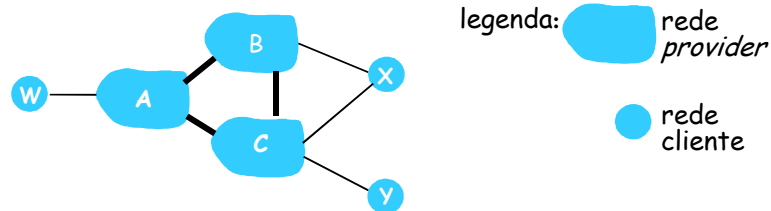
Camada de Rede 4-122

BGP: mensagens

- ❑ Mensagens BGP trocadas usando TCP.
- ❑ Mensagens BGP:
 - **OPEN**: abre conexão TCP até *peer* e autentica emissor
 - **UPDATE**: anuncia novo caminho (ou retira antigo)
 - **KEEPALIVE** mantém conexão *alive* na ausência de UPDATES; também faz o ACK de OPEN request
 - **NOTIFICATION**: reporta erros em msg anterior; também usado para fechar conexão

Camada de Rede 4-123

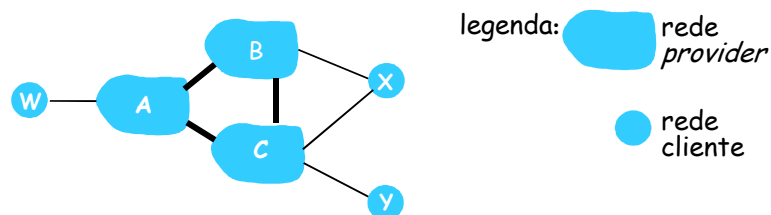
BGP: política de routing



- A,B,C são *redes providers*
- X,W,Y são clientes (de redes provider)
- X é *dual-homed*: ligada a duas redes
 - X não quer encaminhar de B para C via X
 - .. então X não irá anunciar a B um caminho para C

Camada de Rede 4-124

BGP: política de routing (2)



- A anuncia caminho AW a B
- B anuncia caminho BAW a X
- Deverá B anunciar caminho BAW a C?
 - Nem pensar! B não "lucra" por encaminhar CBAW uma vez que nem W nem C são clientes de B
 - B quer forçar C a encaminhar para w via A
 - B quer encaminhar *apenas* para/de seus clientes!

Camada de Rede 4-125

Porquê encaminhamento Intra- e Inter-AS diferente?

Política:

- ❑ Inter-AS: administração quer controlo sobre como é encaminhado o tráfego, quem encaminha pela sua rede
- Intra-AS: administração única, por isso não são necessárias decisões políticas

Escala:

- ❑ *routing* hierárquico poupa tamanho de tabelas, reduz tráfego de actualização

Desempenho:

- ❑ Intra-AS: pode focar-se em desempenho
- ❑ Inter-AS: política pode dominar sobre desempenho

Camada de Rede 4-126

Capítulo 4: Camada de Rede

- ❑ 4.1 Introdução
- ❑ 4.2 Circuitos virtuais e redes datagrama
- ❑ 4.3 O que existe dentro de um *router*
- ❑ 4.4 IP: Internet Protocol
 - Formato do datagrama
 - Endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de *routing*
 - Estado de ligação
 - Vector distância
 - *Routing* hierárquico
- ❑ 4.6 *Routing* na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 *Routing broadcast e multicast*

Camada de Rede 4-127