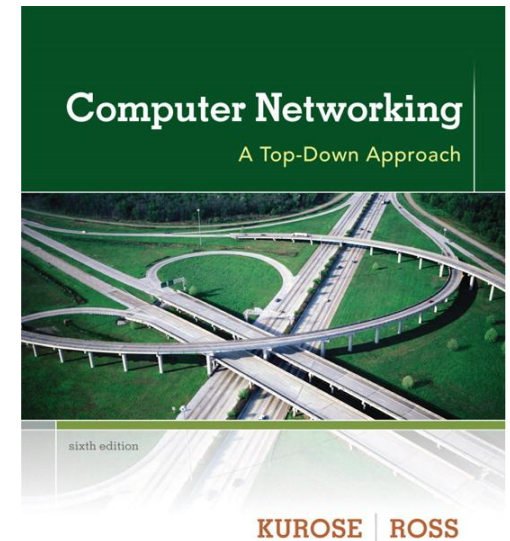


Roteamento

Adaptado e traduzido do material do Livro
“*Computer Networking: A Top-Down Approach*”



*Computer
Networking: A
Top Down
Approach*
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

Agenda

Roteamento

- Estático
- Dinâmico

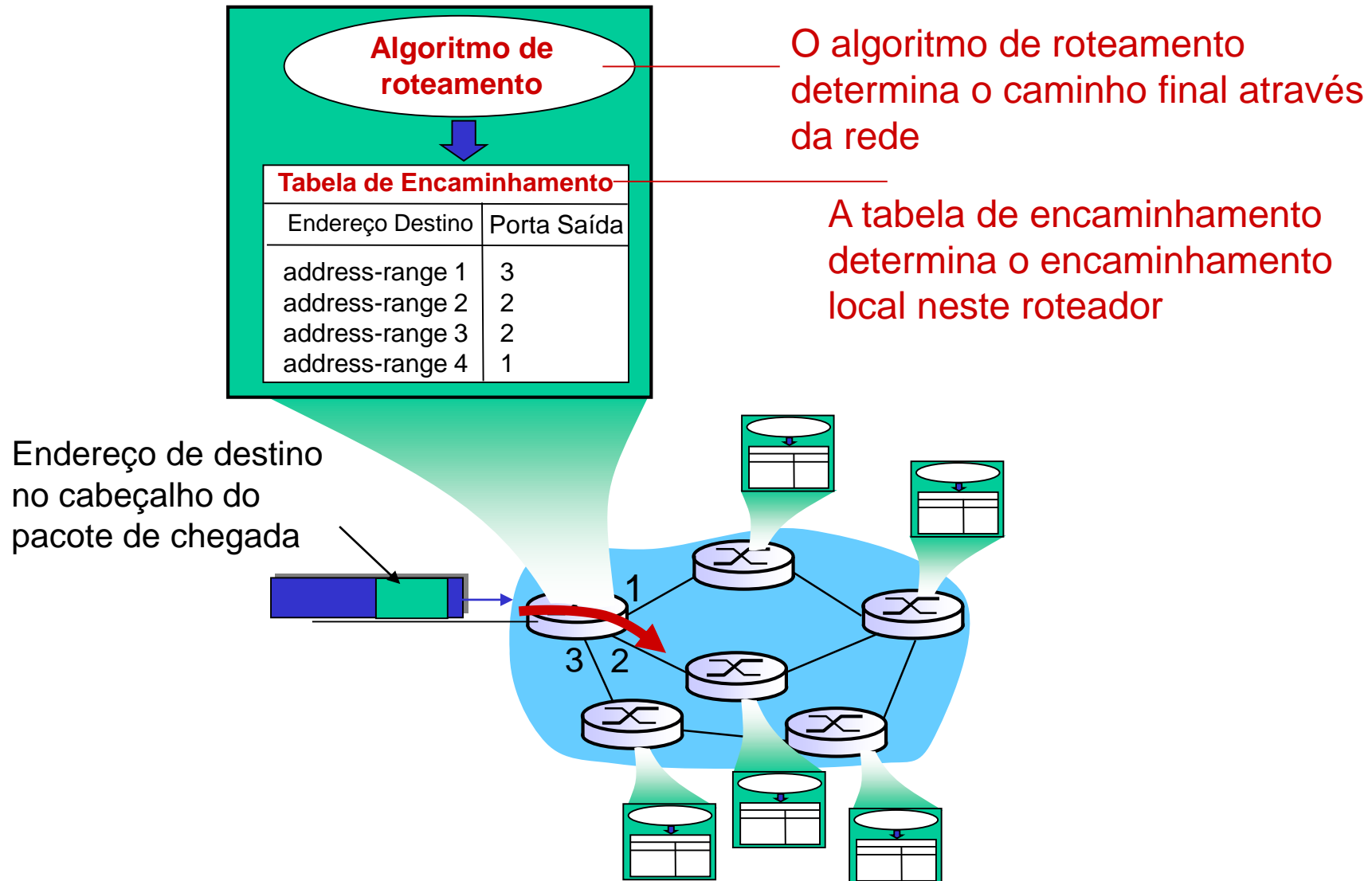
Algoritmos de Roteamento

- Link state
- Distance vector
- Roteamento Hierárquico

Roteamento na Internet

- RIP
- OSPF
- BGP

Roteamento vs. Encaminhamento



Roteamento estático

- ❖ Um administrador pode configurar as rotas de forma estática para as redes destino
- ❖ Informação necessária:
 - Endereço de destino
 - Endereço da rede que pode ser alcançada
 - Ponteiro para o destino (rota)
 - Diretamente conectada
 - Próximo roteador (next-hop router)
- ❖ Sintaxe de uma rota:
 - ip route <end destino> <máscara> <next-hop>
 - ip route <end destino> <máscara> <interface>

Roteamento estático

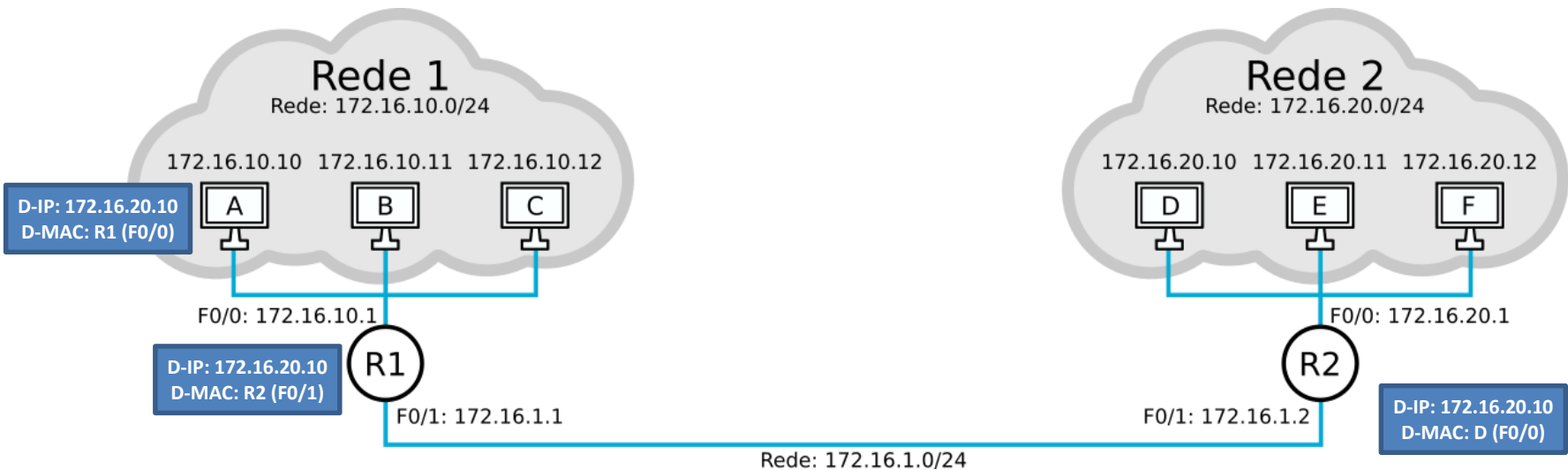


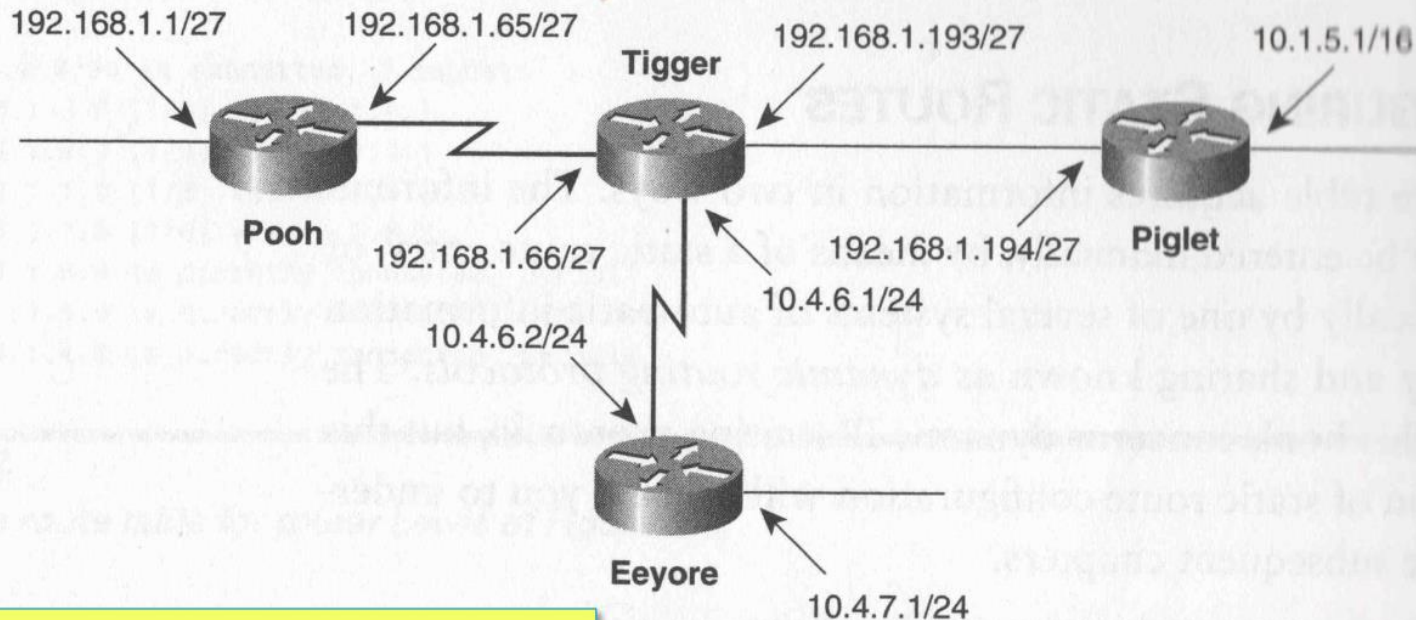
Tabela R1

Tipo	Destino	Próximo Salto
Connected	172.16.10.0/24	F0/0
Connected	172.16.1.0/24	F0/1
Static	172.16.20.0/24	172.16.1.2

Tabela R2

Tipo	Destino	Próximo Salto
Connected	172.16.20.0/24	F0/0
Connected	172.16.1.0/24	F0/1
Static	172.16.10.0/24	172.16.1.1

Roteamento estático - agregação



Rotas para o roteador Piglet

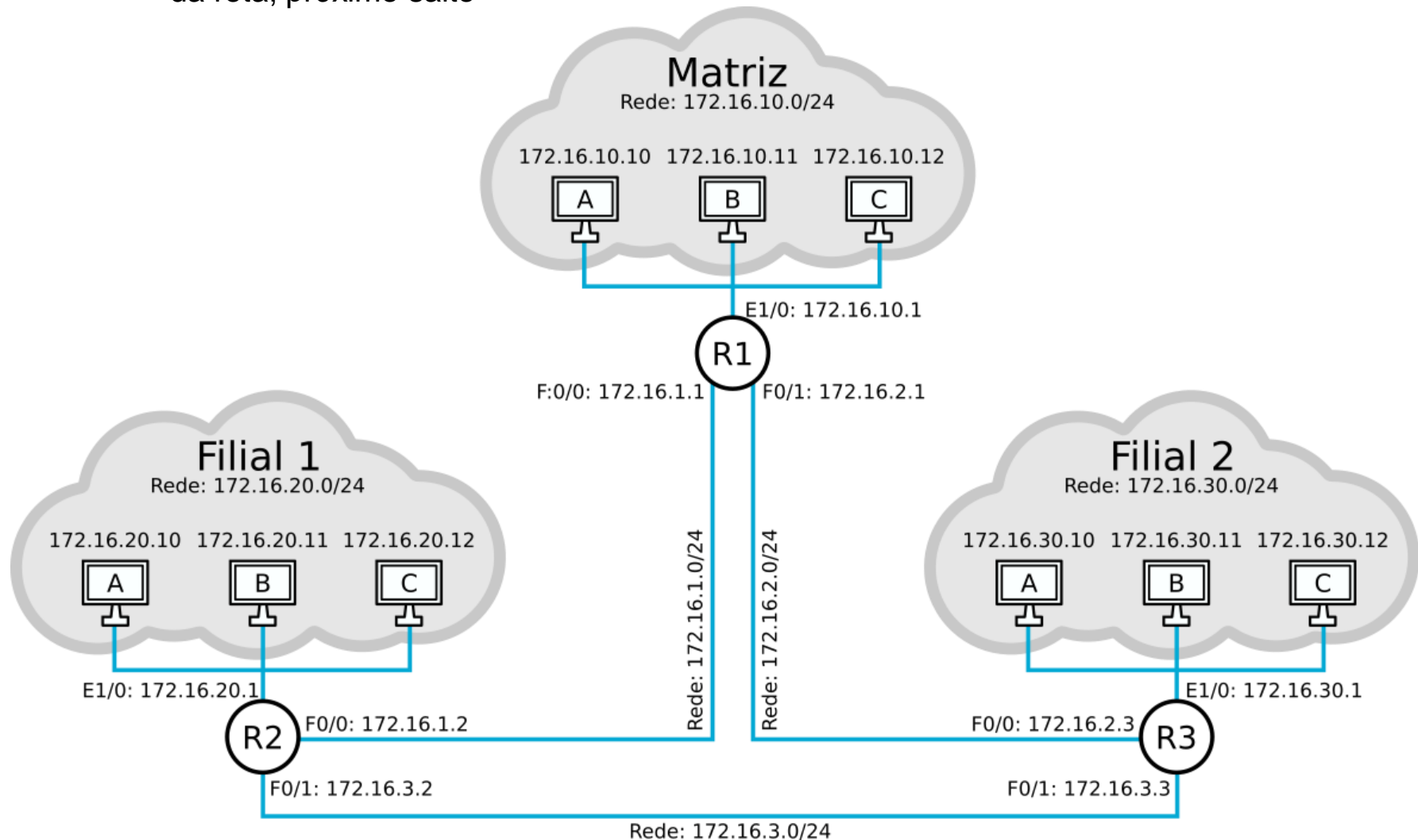
$10.4.6.0/24 \Rightarrow 192.168.1.193$
 $10.4.7.0/24 \Rightarrow 192.168.1.193$
 $192.168.1.64/27 \Rightarrow 192.168.1.193$
 $192.168.1.0/27 \Rightarrow 192.168.1.193$

Rotas sumarizadas do roteador Piglet

$10.4.0.0/16 \Rightarrow 192.168.1.193$
 $192.168.1.0/24 \Rightarrow 192.168.1.193$

Exercício

- ❖ Mostre as tabelas de roteamento de cada roteador da rede abaixo.
 - As tabelas devem conter: tipo de rota (static ou connected), endereço de rede da rota, próximo salto



Agenda

Roteamento

- Estático
- Dinâmico

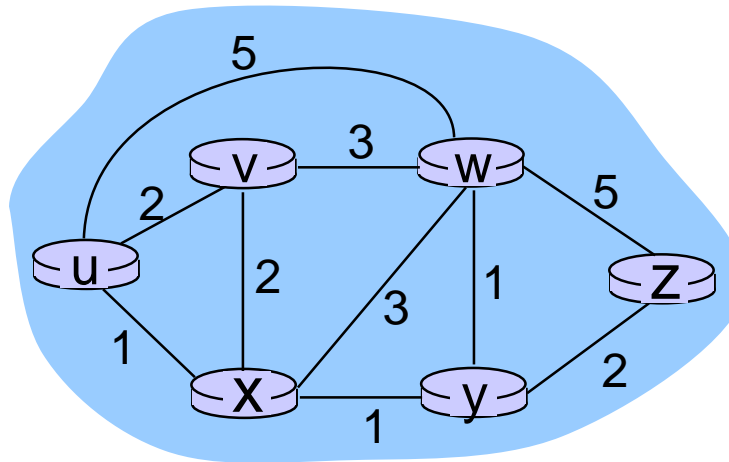
Algoritmos de Roteamento

- Link state
- Distance vector
- Roteamento Hierárquico

Roteamento na Internet

- RIP
- OSPF
- BGP

Abstração em Grafo



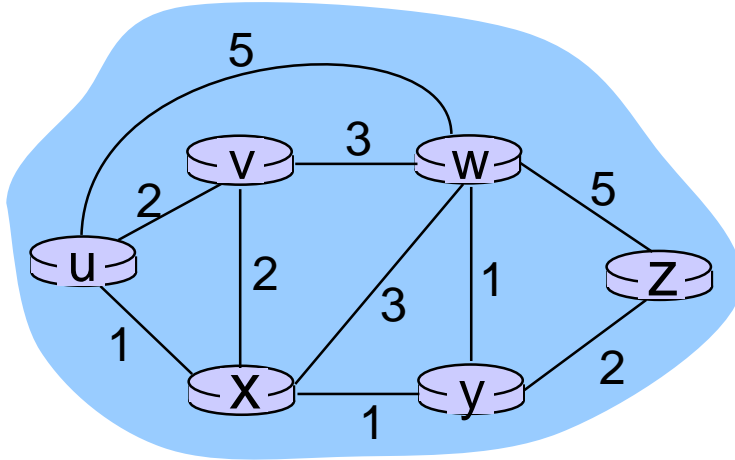
Grafo: $G = (N, E)$

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

E = conjunto de links = $\{ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) \}$

Abstração de redes como grafo é útil em outros contextos, por exemplo, P2P, onde N é conjunto de pares e E é conjunto de conexões TCP

Abstração em Grafo: Custo



$c(x,x')$ = custo do link (x,x')
e.g., $c(w,z) = 5$

O custo sempre pode ser 1, ou inversamente relacionado à largura de banda, ou inversamente relacionado ao congestionamento

Custo de um 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-chave: qual é o caminho de menor custo entre u e z ?

Algoritmo de roteamento: algoritmo que localiza esse caminho de menor custo.

Classificação de Algoritmos de Roteamento

Visão Global:

- ❖ *Todos os roteadores possuem topologia completa, informações de custo do link*
- ❖ *Algoritmos “link state”*

Visão Descentralizado:

- ❖ *O roteador conhece os vizinhos fisicamente conectados, relaciona os custos com os vizinhos*
- ❖ *Processo iterativo de computação, intercâmbio de informações com os vizinhos*
- ❖ *Algoritmos de “vetor de distância”*

Atualização Estática:

- ❖ *Rotas são alteradas lentamente ao longo do tempo*

Atualização Dinâmica:

- ❖ *Rotas mudam rapidamente*
 - *Atualizações periódicas*
 - *Em resposta a mudanças de estado dos enlaces (e.g., custos)*

Agenda

Roteamento

- Estático
- Dinâmico

Algoritmos de Roteamento

- Link state
- Distance vector
- Roteamento Hierárquico

Roteamento na Internet

- RIP
- OSPF
- BGP

Algoritmo de Link-State

Algoritmo de Dijkstra

- ❖ Topologia de rede e custos conhecidos por todos os nós
- ❖ Realizado através da "transmissão do estado do link"
- ❖ Todos os nós têm a mesma informação
- ❖ Calcula os caminhos do menor custo de um nó (fonte) para todos os outros nós
- ❖ Fornece tabela de encaminhamento para esse nó
- ❖ Iterativo: após k iterações, descobre o caminho do menor custo para destinos k

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) = ∞
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'
```

Notação:

- ❖ $c(x,y)$: custo do link de x para y; $= \infty$ se não forem vizinhos
- ❖ $D(v)$: valor atual do custo de ir da origem até o destino v
- ❖ $p(v)$: nó predecessor no caminho até v
- ❖ N' : conjunto de nós para os quais já se conhece o menor caminho

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	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxvyz					

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

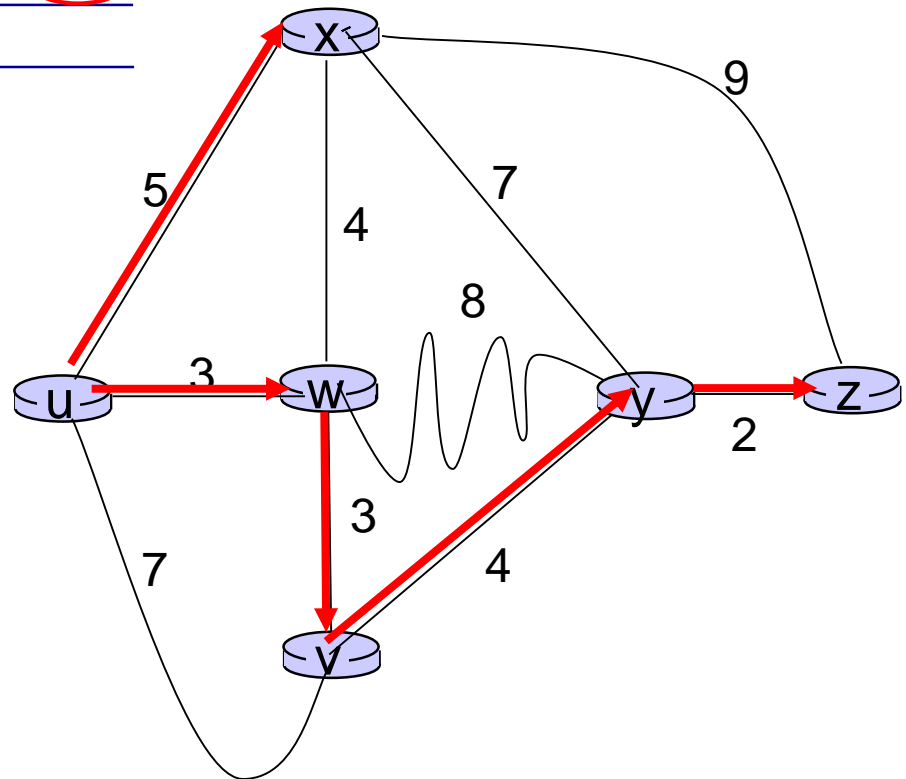
```

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'

```

Note:

- ❖ Constrói-se a árvore do caminho mais curto rastreando os nós predecessores
- ❖ Podem acontecer empates que podem ser desfeitos arbitrariamente



Agenda

Roteamento

- Estático
- Dinâmico

Algoritmos de Roteamento

- Link state
- Distance vector
- Roteamento Hierárquico

Roteamento na Internet

- RIP
- OSPF
- BGP

Algoritmo Distance Vector

Bellman-Ford (programação dinâmica)

Seja

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

Então

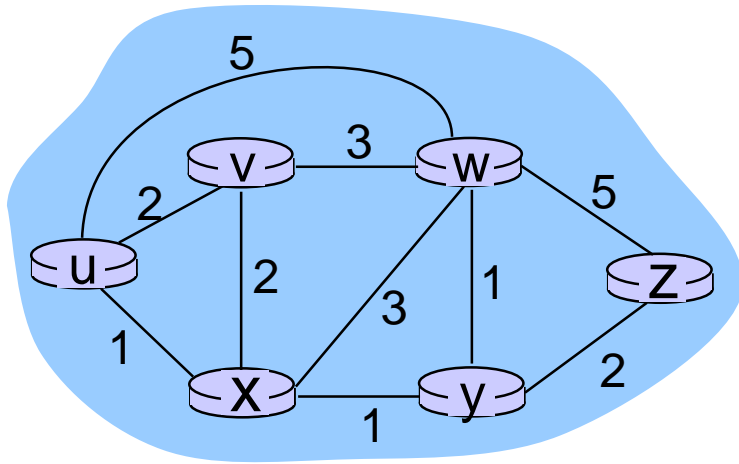
$$d_x(y) = \min_v \{ c(x,v) + d_v(y) \}$$

Custo do vizinho v para o destino y

Custo do menor caminho até v

\min entre todos os vizinhos v de x

Bellman-Ford: Exemplo



Sabendo que

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

B-F diz que:

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

O nó que alcança o mínimo é o próximo salto no caminho mais curto, usado na tabela de encaminhamento

Algoritmo Distance Vector

Ideia central

- ❖ De tempos em tempos, cada nó envia sua própria estimativa de vetor de distância para vizinhos
- ❖ Quando x recebe nova estimativa DV de um vizinho, atualiza sua própria DV usando a equação B-F

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

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

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

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

**node x
table**

		cost to		
		x	y	z
from	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

**node y
table**

		cost to		
		x	y	z
from	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

**node z
table**

		cost to		
		x	y	z
from	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	7	1	0

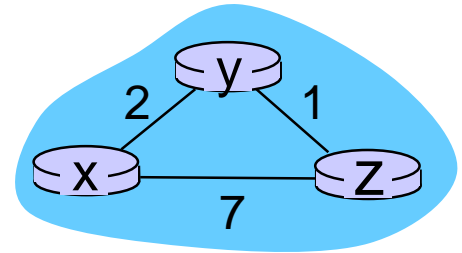
		cost to		
		x	y	z
from	x	0	2	7
	y	2	0	1
	z	7	1	0

		cost to		
		x	y	z
from	x	0	2	7
	y	2	0	1
	z	3	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	3	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	3	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	3	1	0



time

Agenda

Roteamento

- Estático
- Dinâmico

Algoritmos de Roteamento

- Link state
- Distance vector
- Roteamento Hierárquico

Roteamento na Internet

- RIP
- OSPF
- BGP

Roteamento Hierárquico

Escala: 600 milhões de destinos:

- ❖ Não é possível armazenar todos os destinos em tabelas de roteamento!
- ❖ As trocas de informações de roteamento iriam inundar os links!

Autonomia administrativa

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

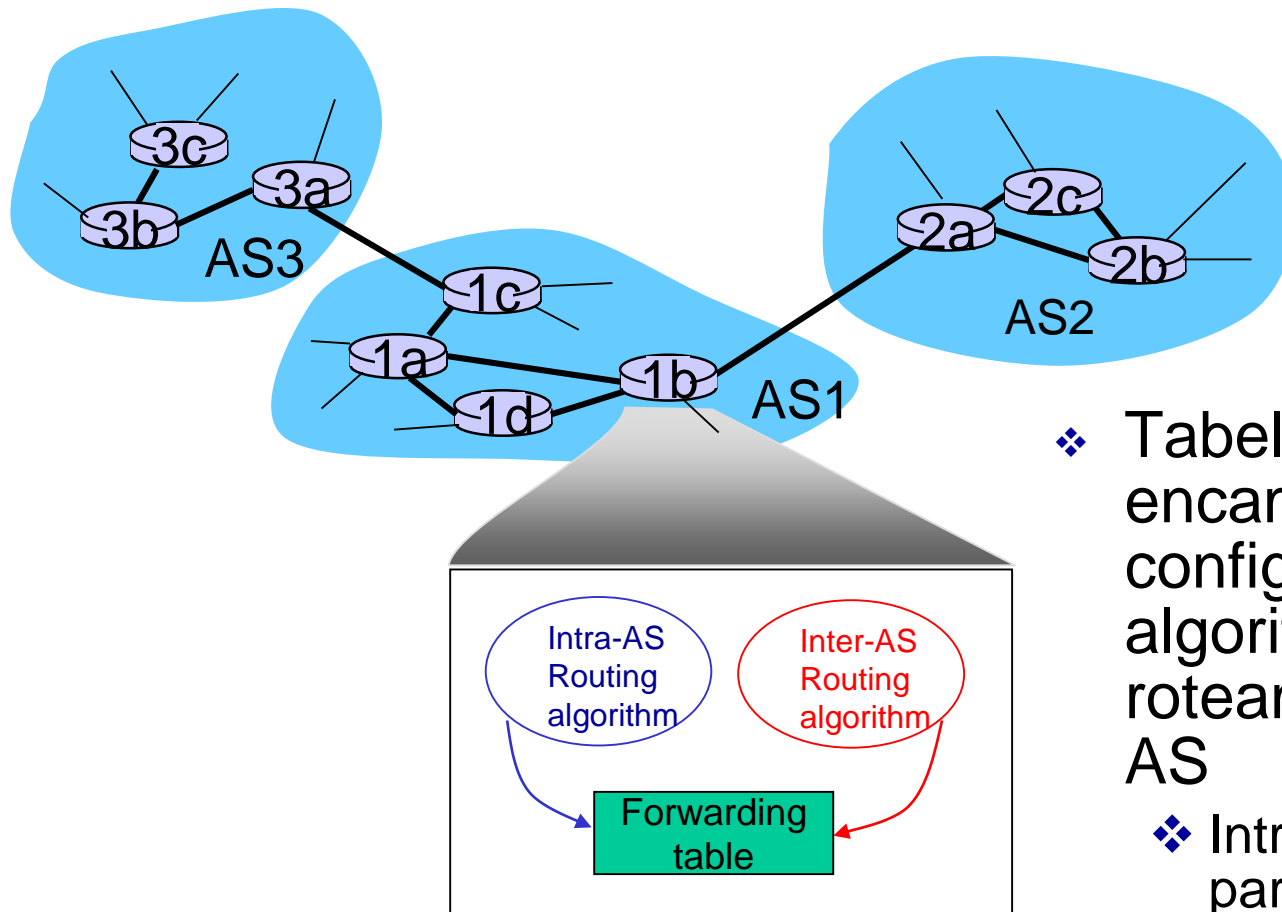
Roteamento Hierárquico

- ❖ Agrega roteadores em regiões, "**sistemas autônomos**" (AS)
- ❖ Roteadores no mesmo AS executam o mesmo protocolo de roteamento
 - Protocolo de roteamento "**intra-AS**"
 - Roteadores em diferentes AS podem executar diferentes protocolo de roteamento intra-AS

Gateway Router:

- ❖ Na borda dos ASs existem os roteadores que fazem a “transição” entre domínios diferentes
- ❖ Tem pelo menos um link direto para um roteador em outro AS

Sistemas Autônomos Interconectados



- ❖ Tabela de encaminhamento configurada por algoritmo de roteamento intra e inter-AS
 - ❖ Intra-AS define entradas para destinos internos
 - ❖ Inter-AS e intra-AS configuram entradas para destinos externos

Agenda

Roteamento

- Estático
- Dinâmico

Algoritmos de Roteamento

- Link state
- Distance vector
- Roteamento Hierárquico

Roteamento na Internet

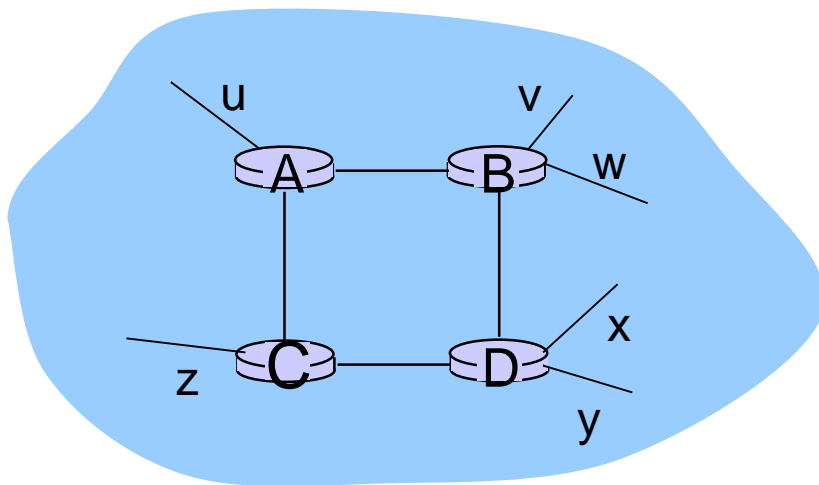
- RIP
- OSPF
- BGP

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 Routing Protocol (Proprietário Cisco)

RIP (Routing Information Protocol)

- ❖ Incluído em sistemas BSD-UNIX em 1982
- ❖ Algoritmo de distance vector
 - Métrica de distância: # hops (max = 15 hops), todos os links de custo 1
 - Os vetores distância são trocados com os vizinhos a cada 30 segundos (**advertisement**)
 - Cada advertisement lista até 25 **subnets**



Olhando a partir do roteador A
para todos os destinos (**subnets**):

<u>subnet</u>	<u>hops</u>
u	1
v	2
w	2
x	3
y	3
z	2

RIP: falhas de enlaces

- ❖ Se nenhum advertisement foi ouvido depois de 180 segundos vizinho/link declarado morto
 - Rotas via vizinho invalidadas
 - Novas mensagens *advertisement* enviadas aos vizinhos
 - Os vizinhos, por sua vez, enviam novos anúncios
 - (se as tabelas forem alteradas)
 - Informações de falha se propagam para toda a rede

OSPF (Open Shortest Path First)

❖ Histórico

- 1989 – RFC #1131 “The OSPF Specification”
 - Inspirado no IS-IS que é padronizado pela ANSI
- 1998 – RFC #2328 “OSPF Version 2”
 - Primeira proposta em 1991 (RFC # 1247)
 - <https://tools.ietf.org/html/rfc2328>
- 2008 – RFC #5340 “OSPF for IPv6”
 - Primeira proposta em 1999 (RFC # 2740)
 - <https://tools.ietf.org/html/rfc5340>

OSPF (Open Shortest Path First)

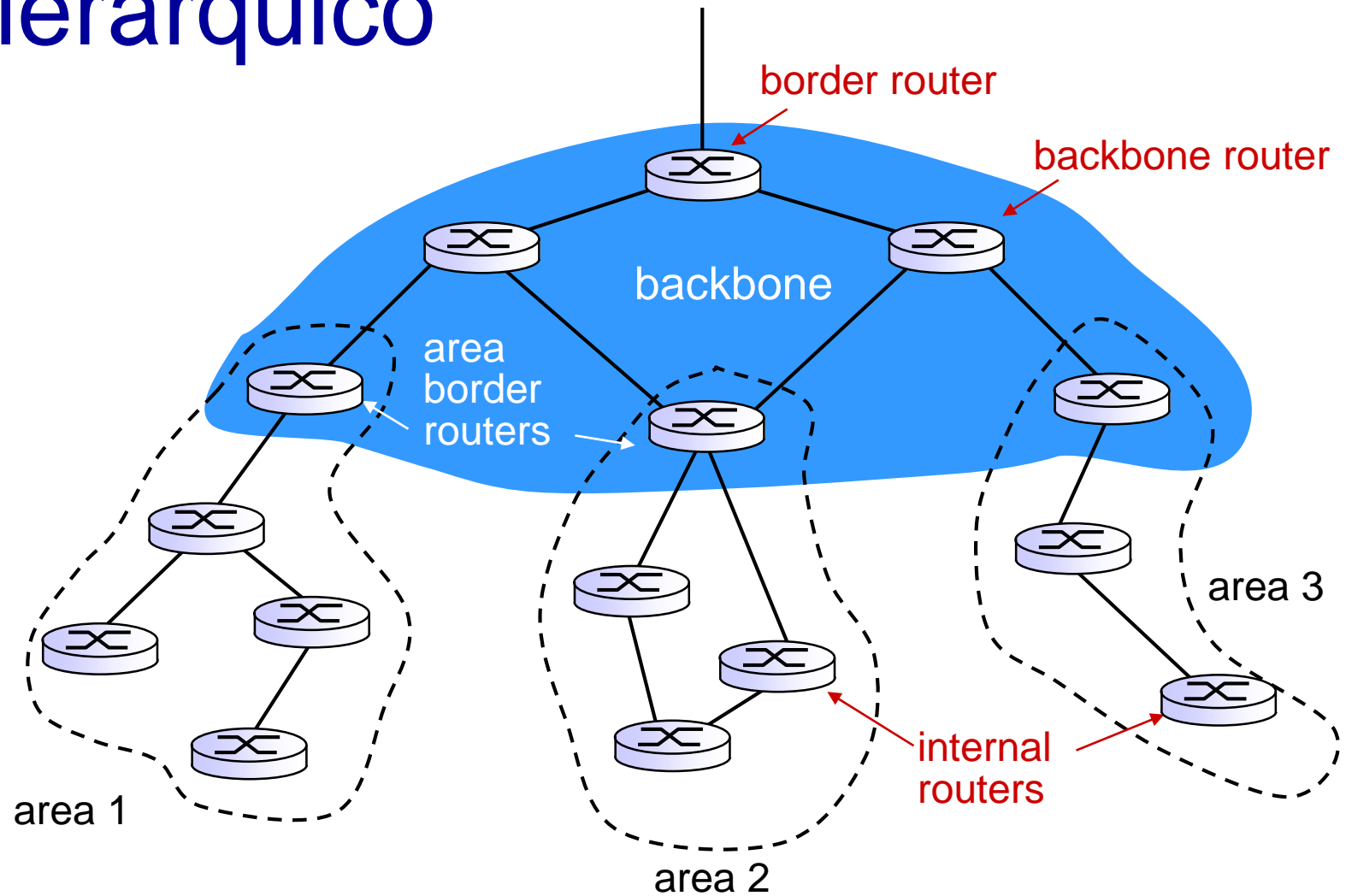
- ❖ “open”: disponível publicamente
- ❖ Usa o algoritmo de *link state*
 - Disseminação de pacotes do estado dos enlaces
 - Topologia mapeada por cada nó
 - Computa as rotas usando Dijkstra
- ❖ As mensagens de advertisement do OSPF carregam uma entrada para cada vizinho
- ❖ *Advertisements* são propagados por todo o AS
 - Transporte sobre mensagens OSPF diretamente sobre IP (não usa TCP ou UDP)

OSPF “funcionalidades avançadas” (inexistentes no RIP)

- ❖ **Segurança**: todas as mensagens OSPF autenticadas (para evitar intrusões maliciosas)
- ❖ **Múltiplos caminhos** do mesmo custo permitidos (apenas um caminho no RIP)
- ❖ **Múltiplas métricas de custos** para diferentes ToS para cada link
 - Por exemplo, configurar o custo de um link de satélite como "baixo" para ToS *best-effort* e alto para ToS de tempo real)
- ❖ OSPF **hierárquico** em grandes domínios

OSPF

Hierárquico



Mensagens OSPF

❖ Hello

- Estabelece a relação de vizinhança (neighbors)
- Mantem essa relação ativa através do envio de mensagens periódicas

❖ Timers

- Redes com suporte a broadcast
 - Hello: 10s
 - Dead: 40s
- Redes sem suporte a broadcast
 - Hello: 30s
 - Dead: 120s

Mensagens OSPF

❖ Database Description

- Definir um mestre e um escravo
 - O roteador mestre será aquele que tiver o identificador mais alto
 - O roteador mestre fará o controle/incremento do DD Sequence Number
- ❖ Uma vez definidos os papéis dos roteadores são enviados **Link State Advertisements (LSA)**
- ❖ Esse cabeçalho contém um resumo das rotas/redes que o roteador conhece

Mensagens OSPF

❖ Link State Request

- Após as trocas de Database Descriptions, cada roteador poderá descobrir se a sua visão local da topologia pode estar desatualizada
- Ao enviar uma mensagem de Link State Request um roteador solicitará uma lista de LSAs de interesse (somente aqueles que precisam ser atualizados)

Inicialmente com as tabelas de roteamento locais vazias muitas requisições serão trocadas entre os roteadores. Depois que a rede “estabiliza”, somente haverá necessidade dessa troca de mensagens quando houverem atualizações de rotas ou estados de enlaces.

Mensagens OSPF

❖ Link State Update

- Esse tipo de mensagem carrega uma descrição mais completa de cada LSA anunciado por um roteador
- São enviadas em multicast para que todos os roteadores recebam as atualizações (mesmo que apenas um tenha solicitado)
 - Apenas quando são reenviadas se utiliza unicast

Mensagens OSPF

❖ **Link State Acknowledgment**

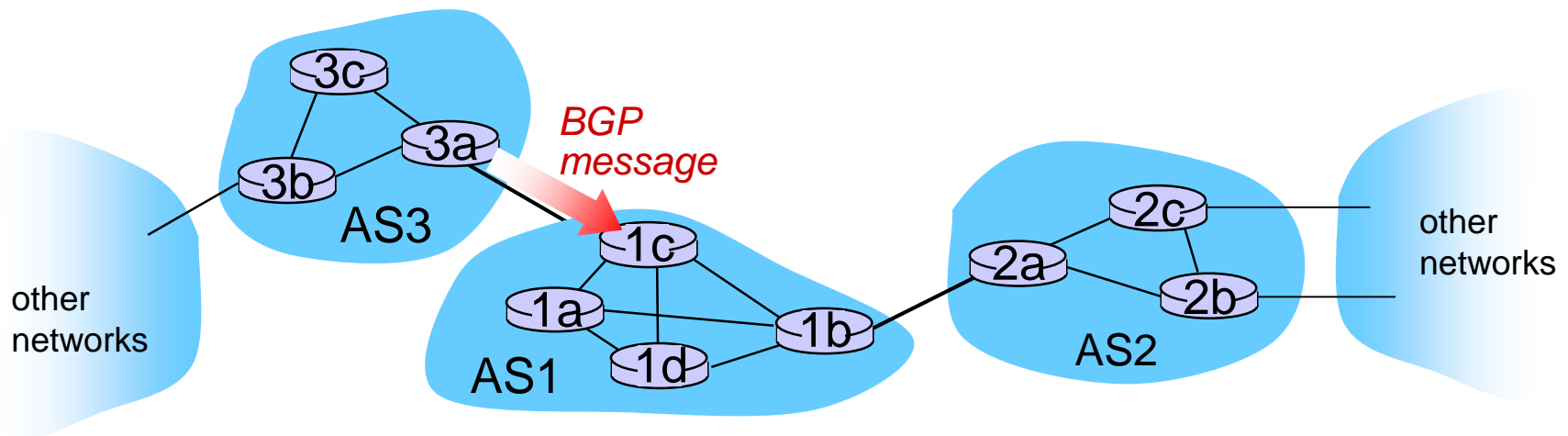
- Por fim, esse tipo de mensagem confirma que LSAs enviado anteriormente foram recebidos nos roteadores destino
- Essa confirmação é importante pois as mensagens de Link State Update são enviadas em multicast, portanto uma ou mais tabelas de roteamento podem ter sido atualizadas

BGP (Border Gateway Protocol):

- ❖ O protocolo padrão *de facto* de roteamento entre domínios administrativos (**inter-AS**)
 - "Cola que mantém a Internet unida"
- ❖ BGP fornece cada AS um meio para:
 - **eBGP**: obtém informações de acessibilidade de subrede dos AS vizinhos
 - **iBGP**: propaga informações de acessibilidade para todos os roteadores internos AS
- ❖ Determina rotas "válidas" para outras redes com base em informações e **políticas de acessibilidade**

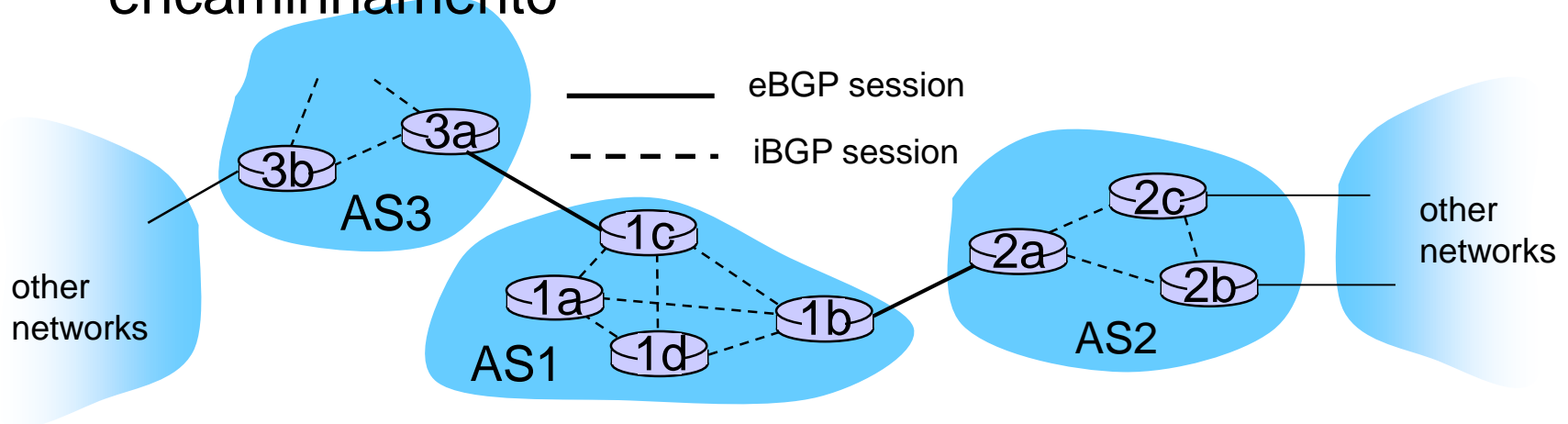
BGP por alto

- ❖ **Sessão BGP:** dois roteadores BGP ("peers") trocam mensagens:
 - ❖ Divulgam os caminhos para diferentes prefixos de rede de destino (protocolo "path vector")
 - ❖ Mensagens trocadas por conexões TCP
- ❖ Quando o AS3 anuncia um prefixo para AS1:
 - AS3 **promete** que irá encaminhar datagramas para esse prefixo
 - AS3 pode agregar prefixos anunciados



BGP por alto

- ❖ Usando a sessão **eBGP** entre **3a** e **1c**, o AS3 envia informações de acessibilidade de prefixo para AS1
 - ❖ **1c** pode usar o **iBGP** para distribuir novas informações de prefixo para todos os roteadores no AS1
 - ❖ **1b** pode então re-anunciar novas informações de acessibilidade para AS2 em uma sessão eBGP entre **1b** a **2a**
- ❖ Quando o roteador aprende sobre o novo prefixo, ele cria entrada para prefixo em sua tabela de encaminhamento



Rotas e atributos BGP

- ❖ O prefixo anunciado inclui atributos BGP
 - Prefixo + atributos = "rota"
- ❖ Dois atributos importantes:
 - **AS-PATH**: contém os ASs através dos quais o anúncio de prefixo passou: por exemplo, AS 67, AS 17
 - **NEXT-HOP**: indica o roteador AS interno específico para o próximo salto. (Podem ser vários links do AS atual para o próximo-salto-AS)
- ❖ Roteador de borda que recebe as rotas usa **políticas de importação** para aceitar/recusar
 - Por exemplo, nunca percorra AS x
 - Roteamento baseado em políticas

Seleção de Rotas em BGP

- ❖ O roteador pode aprender sobre mais de uma rota para AS destino, seleciona rota com base em:
 - ❖ Atributo de valor de preferência local: decisão de política
 - ❖ Caminho mais curto AS-PATH
 - ❖ Roteador NEXT-HOP mais próximo: hot potato routing
 - ❖ Critérios adicionais