

# Aula 17 - BGP, Roteamento Broadcast e Multicast

Diego Passos

Universidade Federal Fluminense

Redes de Computadores

Material adaptado a partir dos slides  
originais de J.F Kurose and K.W. Ross.

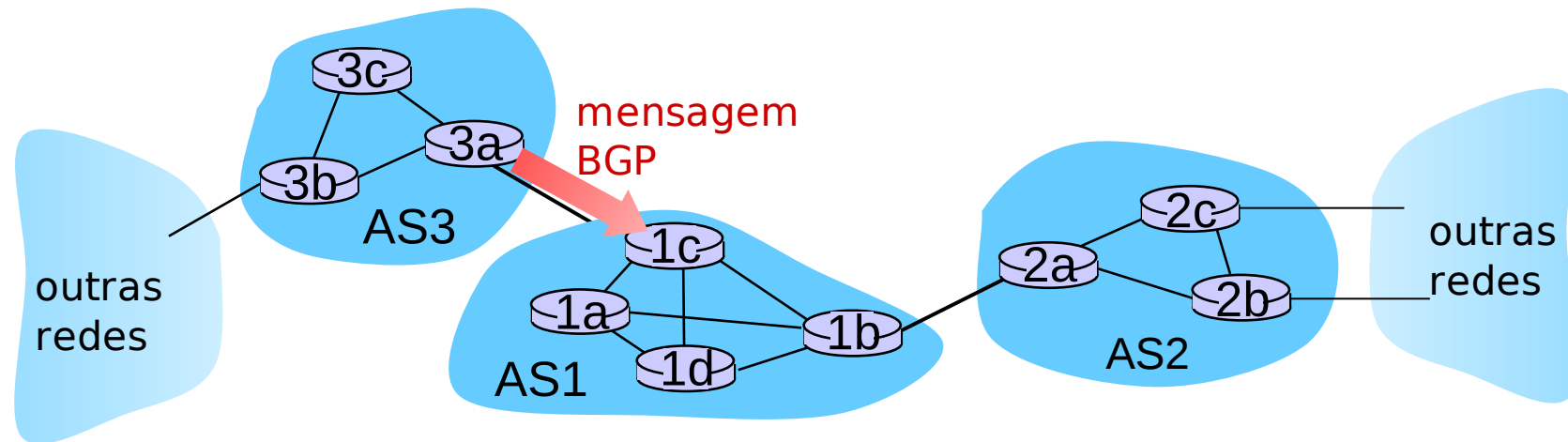
# Roteamento Inter-AS

# Roteamento Inter-AS na Internet: BGP

- **BGP (Border Gateway Protocol):** o padrão *de facto* para roteamento inter-domínios.
  - “A cola que mantém a Internet junta”.
- BGP provê a cada AS uma maneira de:
  - **eBGP:** obter informações sobre sub-redes alcançáveis de ASs vizinhos.
  - **iBGP:** propagar informações de sub-redes externas alcançáveis para todos os roteadores do AS.
  - Determinar “boas” rotas para outras redes com base nas informações das sub-redes alcançáveis e políticas diversas.
- Permite a uma sub-rede anunciar sua existência para o resto da Internet: **“estou aqui!”**

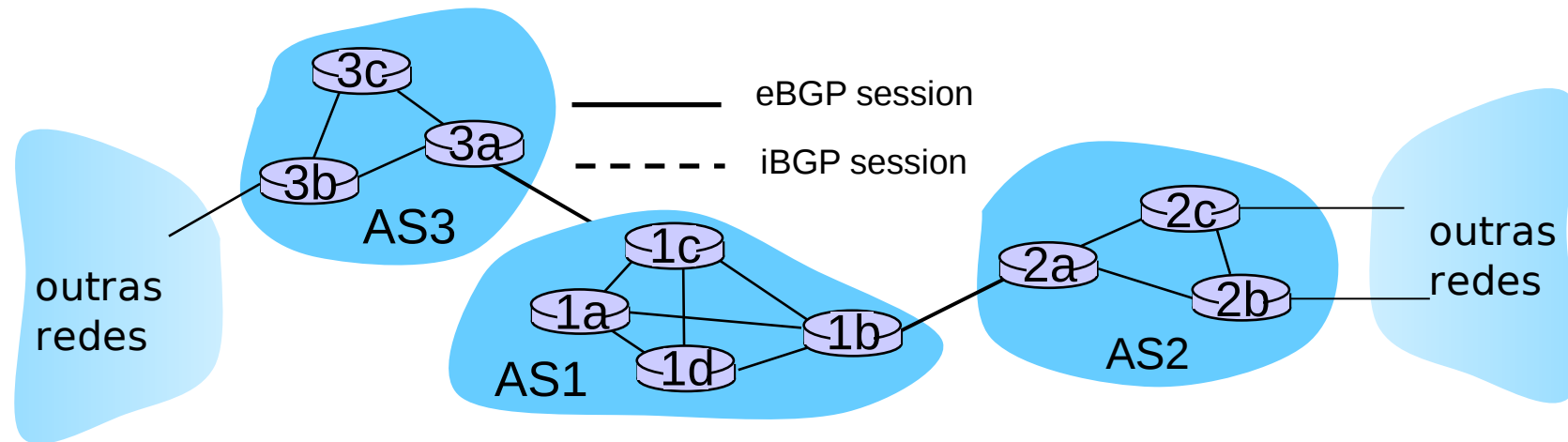
# BGP: Conceitos Básicos

- **Sessão BGP:** dois roteadores BGP (“peers”) trocam mensagens BGP.
  - Anunciam **caminhos** para vários prefixos de sub-redes diferentes (protocolo baseado em “vetor de caminhos”).
  - Trocadas sobre conexões TCP semi-permanentes.
- Quando o AS3 anuncia um prefixo para o AS1:
  - AS3 **se compromete** a encaminhar datagramas em direção àquele prefixo.
  - AS3 pode agregar prefixos nos seus anúncios.



# BGP: Distribuindo Informação de Caminhos

- Usando a sessão eBGP entre 3a e 1c, AS3 envia prefixos alcançáveis para o AS1.
  - 1c pode, então, usar o iBGP para distribuir novas informações de prefixos conhecidos para todos os roteadores no AS1.
  - 1b pode, então, re-anunciar a informação de prefixos alcançáveis para o AS2 através da sessão eBGP entre 1b e 2a.
- Quanto roteador aprende um novo prefixo, ele cria uma entrada para o prefixo na sua tabela de roteamento.



# Atributos de Caminhos e Rotas BGP

- Prefixos anunciados incluem atributos BGP.
  - prefixo + atributos = “rota”.
- Dois atributos importantes:
  - **AS-PATH**: contém lista de ASs pelos quais o anúncio do prefixo passou: e.g., AS 67, AS 17.
  - **NEXT-HOP**: indica roteador do AS de próximo salto (que pode estar a vários saltos do AS atual).
- Roteador *gateway* que recebe anúncios utiliza **política de importação** para aceitá-los ou não.
  - e.g., nunca encaminhe para o AS x.
  - Roteamento baseado em **políticas**.

# BGP: Seleção de Rota

- Roteador pode aprender múltiplas rotas para um AS de destino. Seleção é baseada em:
  1. Valor do atributo *Local Preference*: decisão baseada em política.
  2. AS-PATH mais curto.
  3. Roteador NEXT-HOP mais próximo: roteamento batata-quente.
  4. Critérios adicionais.

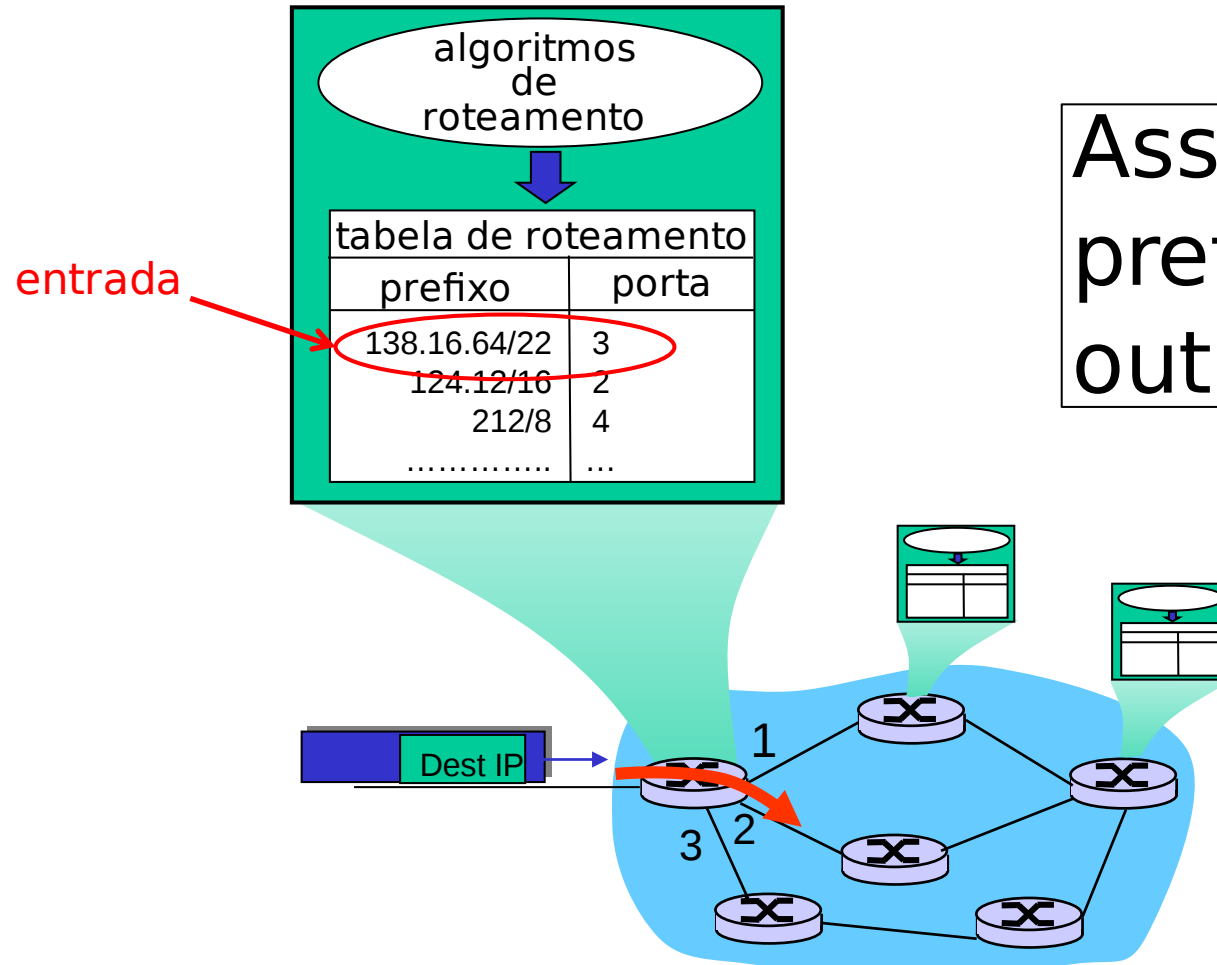
# Mensagens BGP

- Mensagens BGP trocadas entre *peers* sobre conexão TCP.
- Mensagens BGP:
  - **OPEN**: abre conexão TCP para o *peer* e autentica transmissor.
  - **UPDATE**: anuncia novo caminho (ou anula anúncio antigo).
  - **KEEPALIVE**: mantém a conexão aberta em caso de inatividade prolongada; também serve de ACK para mensagem OPEN.
  - **NOTIFICATION**: reporta erros na mensagem anterior; também usada para fechar conexão.



- **Como uma entrada é colocada na tabela de roteamento de um roteador?**
  - Resposta é complicada!
  - Junta Roteamento Hierárquico, BGP e OSPF.
  - Provê boa visão geral do funcionamento do BGP!

# Como a Entrada é Inserida na Tabela de Roteamento? (I)



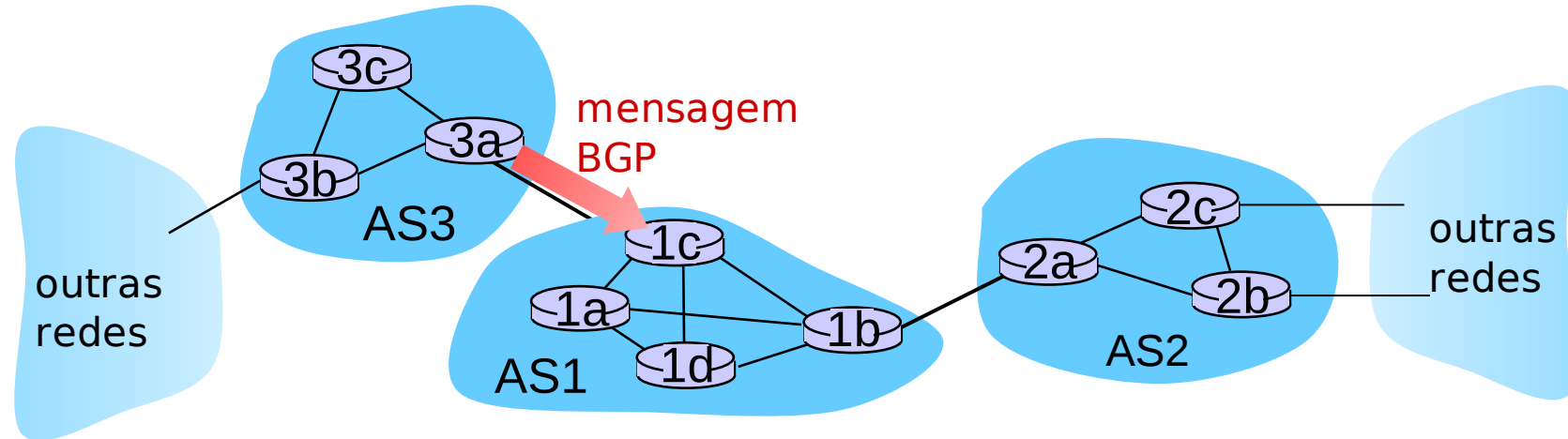
Assuma que o prefixo é de outro AS

# Como a Entrada é Inserida na Tabela de Roteamento? (II)

- **Visão de alto nível:**

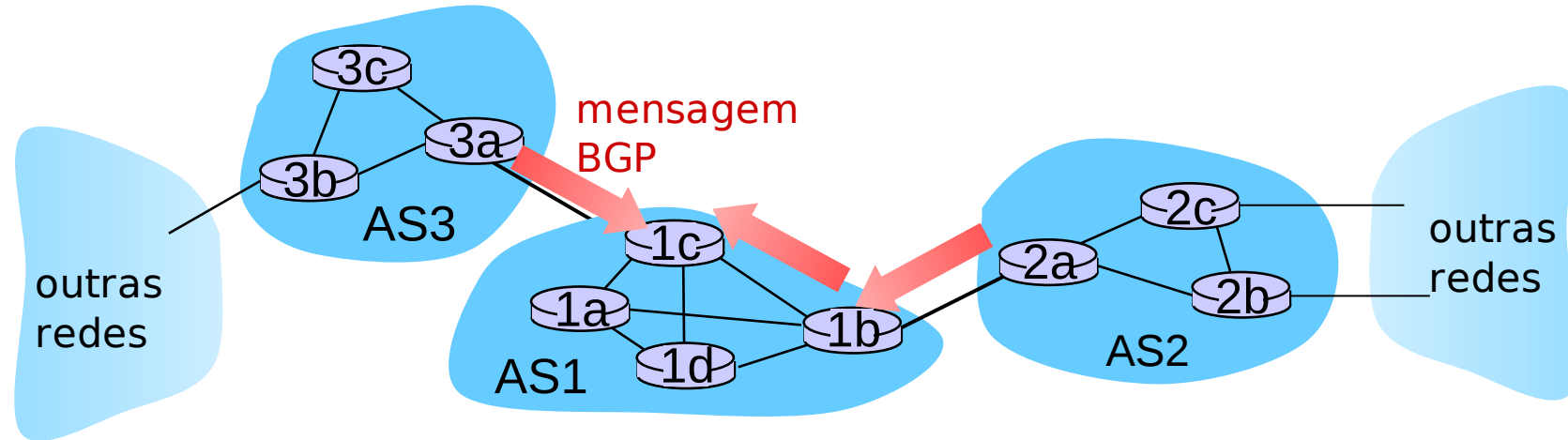
1. Roteador passa a conhecer o prefixo.
2. Roteador determina a porta de saída para alcançar o prefixo.
3. Roteador insere mapeamento porta-prefixo na tabela de roteamento.

# Roteador Passa a Conhecer o Prefixo



- Mensagem BGP contém “rotas”.
- “Rota” é um prefixo, mais seus atributos: AS-PATH, NEXT-HOP, ...
- Exemplo de rota BGP:
  - Prefixo: 138.16.64/22.
  - AS-PATH: AS3 AS131.
  - NEXT-HOP: 201.44.13.125.

# Roteador Pode Receber Múltiplas Rotas



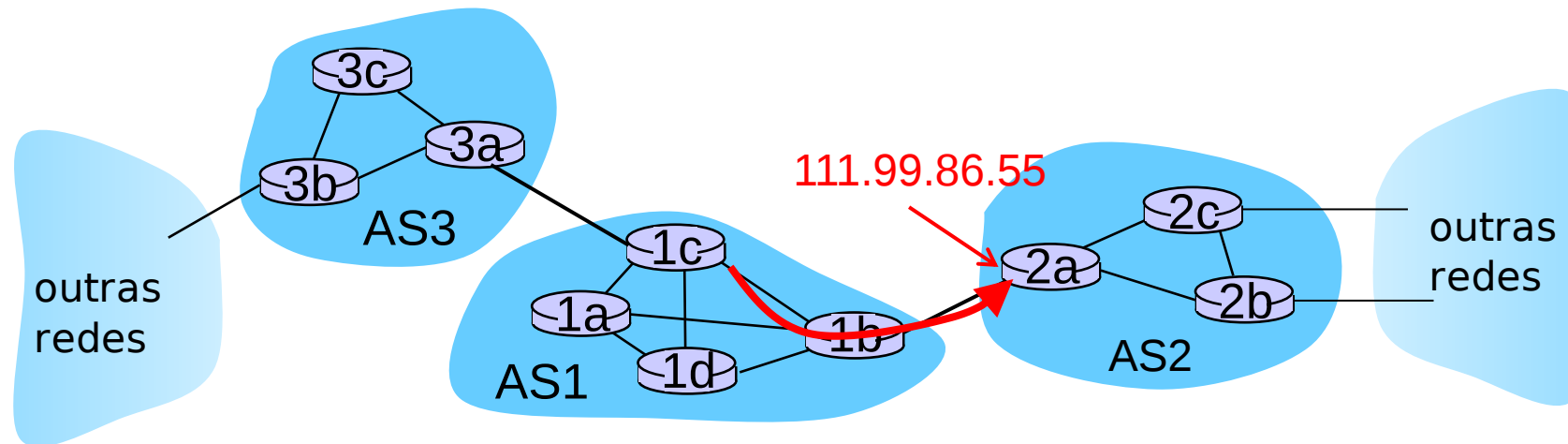
- Roteador pode receber múltiplos anúncios de rotas para **um mesmo prefixo**.
- Precisa selecionar um.

# Seleção da Melhor Rota BGP para o Prefixo

- Roteador seleciona rota com base no AS-PATH mais curto.
- Por exemplo:
  - AS2 AS17 para alcançar 138.16.64/22 (**selecionada**).
  - AS3 AS131 AS201 para alcançar 138.16.64/22.
- E se ocorrer um empate? Voltaremos a este ponto em breve!

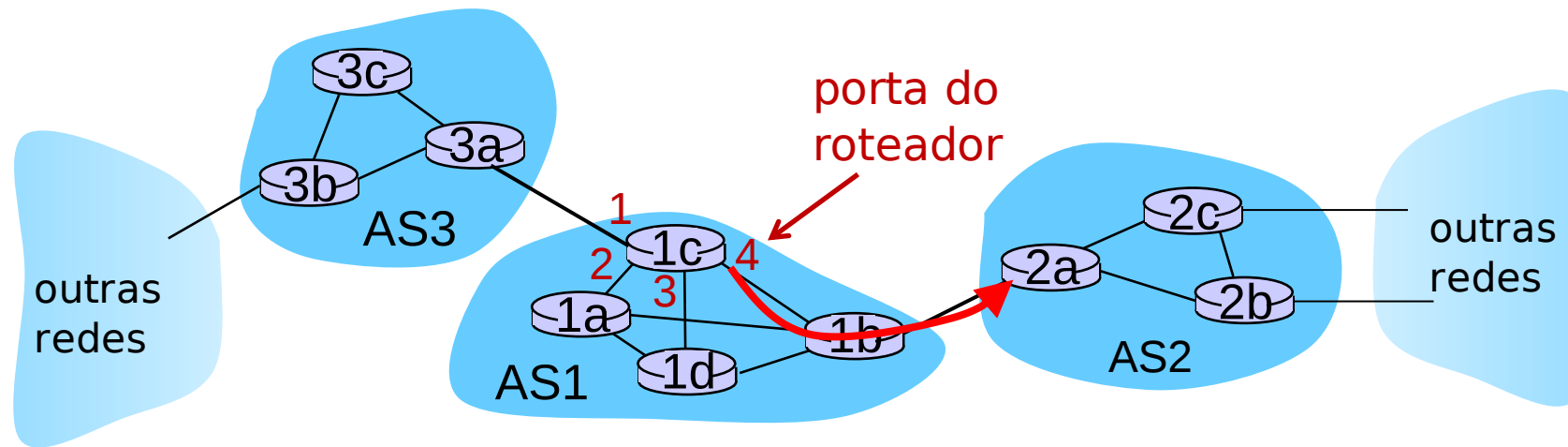
# Encontrar Melhor Rota Interna para a Rota BGP

- Use o atributo NEXT-HOP da rota selecionada.
  - É o endereço IP da interface do roteamento que inicia o AS PATH.
- Exemplo:
  - AS-PATH: AS2 AS17; NEXT-HOP: 111.99.86.55.
- Roteador usa OSPF para encontrar caminho mais curto de 1c para 111.99.86.55.



# Roteador Identifica Porta para Rota

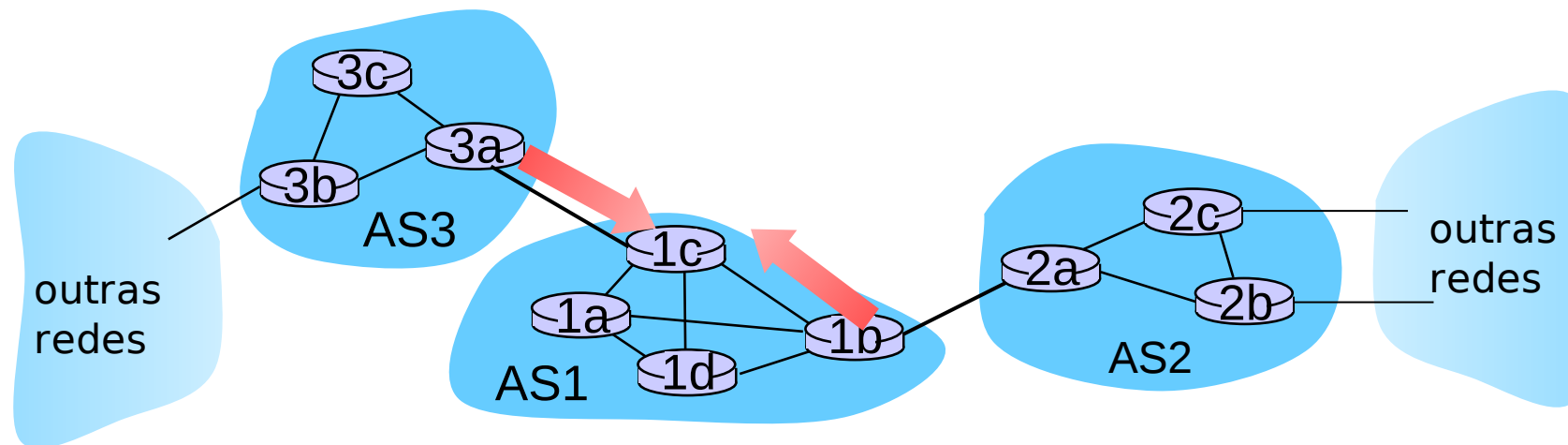
- Identifica porta no caminho mais curto apontado pelo OSPF.
- Adiciona entrada mapeamento o prefixo à porta na sua tabela de roteamento.
  - **(183.16.64/22, porta 4).**





# Roteamento Batata-Quente

- Suponha que haja duas ou mais melhores rotas BGP.
- Então escolha aquela com o NEXT-HOP mais próximo.
  - **Use o OSPF para determinar qual gateway está mais próximo.**
  - **Pergunta:** de 1c, escolher AS3 AS131 ou AS2 AS17?
  - **Resposta:** rota AS3 AS 131, já que NEXT-HOP é mais próximo.



# Como uma Entrada é Inserida na Tabela de Roteamento?

- Sumário:

1. Roteador passa a conhecer o prefixo.

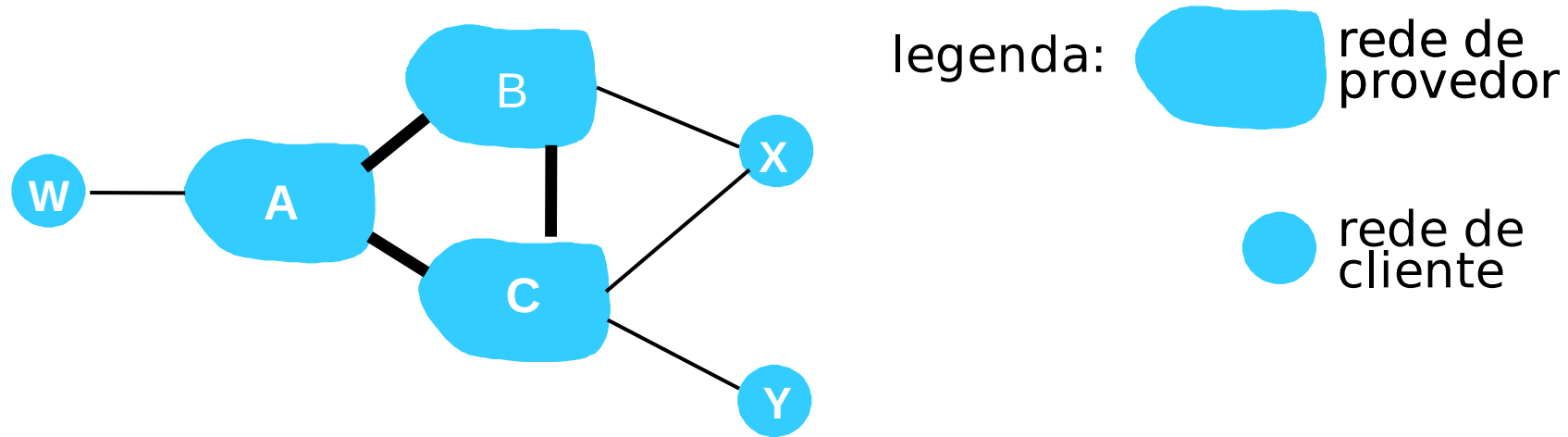
- Através de anúncios BGP de outros roteadores.

2. Determina porta de saída para o prefixo.

- Usa seleção de rotas BGP para encontrar a melhor rota BGP.
- Usa OSPF para encontrar melhor rota interna ao AS que leva o pacote até o início da melhor rota BGP.
- Roteador identifica porta de saída para esta rota.

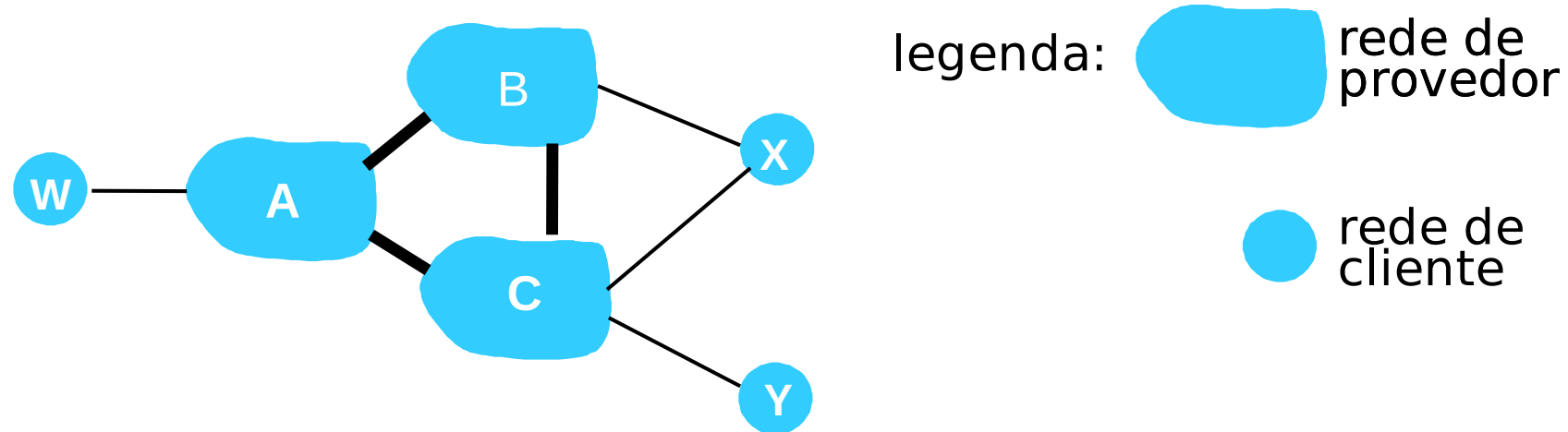
3. Insere o mapeamento (prefixo, porta) na tabela de roteamento.

# BGP: Políticas de Roteamento (I)



- A, B e C são redes de provedores.
- X, W, e Y são consumidores (clientes dos provedores).
- X é **dual-homed**: se conecta à Internet por dois provedores diferentes.
  - X não quer rotear pacotes de B para C.
  - ...logo, X não irá anunciar para B uma rota para C.

# BGP: Políticas de Roteamento (II)



- A anuncia caminho AW para B.
- B anuncia caminho BAW para X.
- B deve anunciar caminho BAW para C?
  - Claro que não! B não lucra roteando CBAW, já que nem W nem C são clientes de B.
  - B quer forçar C a rotear para W por A.
  - B quer rotear **apenas** tráfego de e para seus clientes!

# Por que Roteamentos Diferentes Intra- e Inter-AS?

- **Políticas:**

- Inter-AS: administrador quer controle sobre como seu tráfego é roteado, quem roteia pela sua rede.
- Intra-AS: único administrador, não são necessárias decisões políticas.

- **Escala:**

- Roteamento hierárquico reduz tamanho das tabelas, reduz tráfego de atualização de rotas.

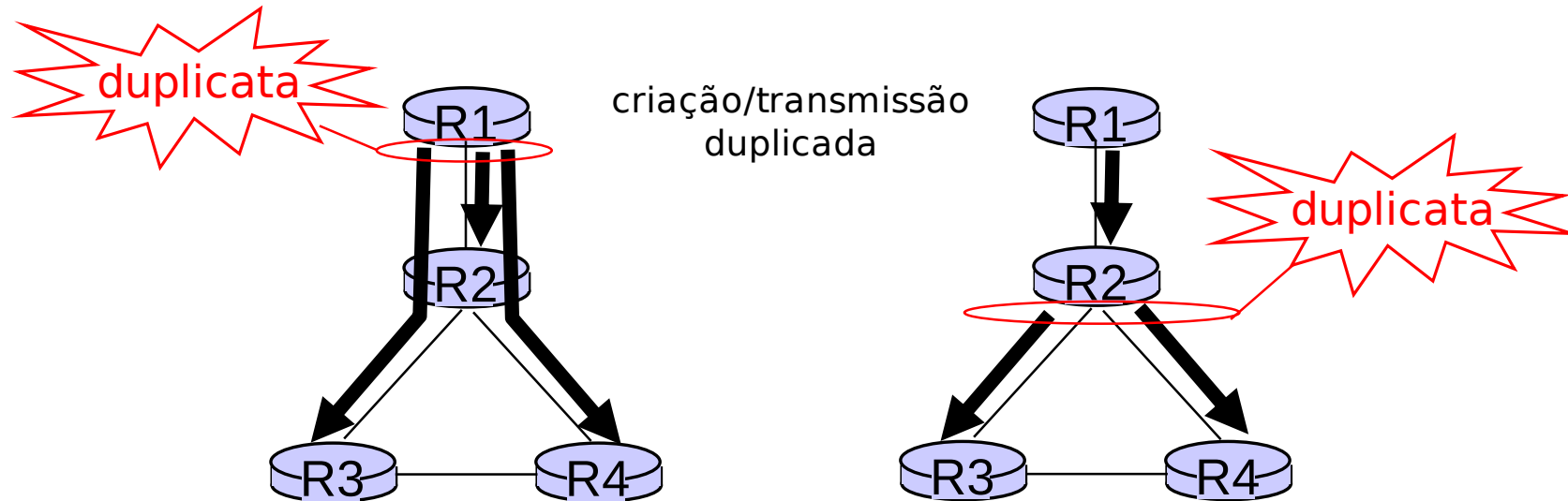
- **Desempenho:**

- Intra-AS: foco no desempenho.
- Inter-AS: políticas podem dominar decisões.

# Roteamento *Broadcast*

# Roteamento Broadcast

- Entrega pacotes da origem a todos os outros nós.
- Replicação na origem é ineficiente:



- Replicação na origem: como a origem determina o endereço dos destinatários?

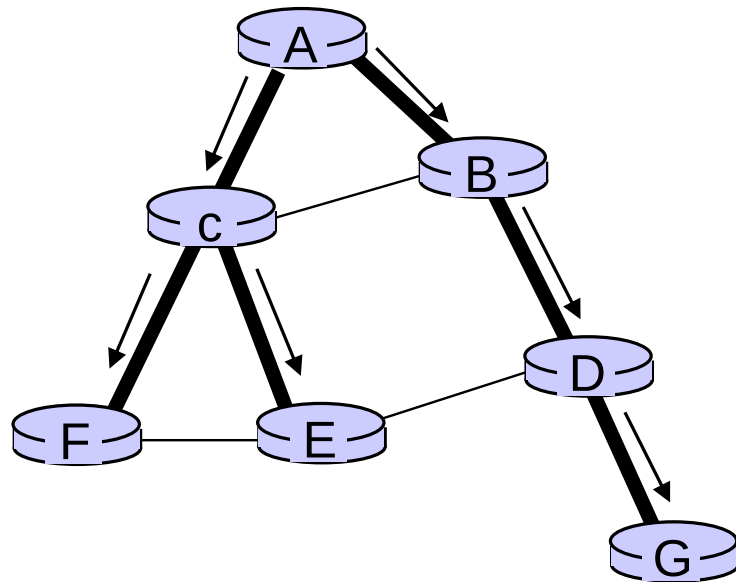
# Replicação na Rede

- **Inundação:** quando nó recebe pacote em *broadcast*, envia cópias para todos os seus vizinhos.
  - Problemas: ciclos e **tempestade de broadcast**.
- **Inundação Controlada:** nó só envia cópias se está é a primeira recepção deste pacote.
  - Nós mantêm lista dos IDs dos pacotes já replicados.
  - Outra alternativa é o *Reverse Path Forwarding* (RPF): apenas replicar pacotes que chegaram pelo enlace de próximo salto do caminho entre o nó corrente e **a origem**.
- **Árvore Geradora (*Spanning Tree*):**
  - Nós nunca recebem pacotes redundantes.

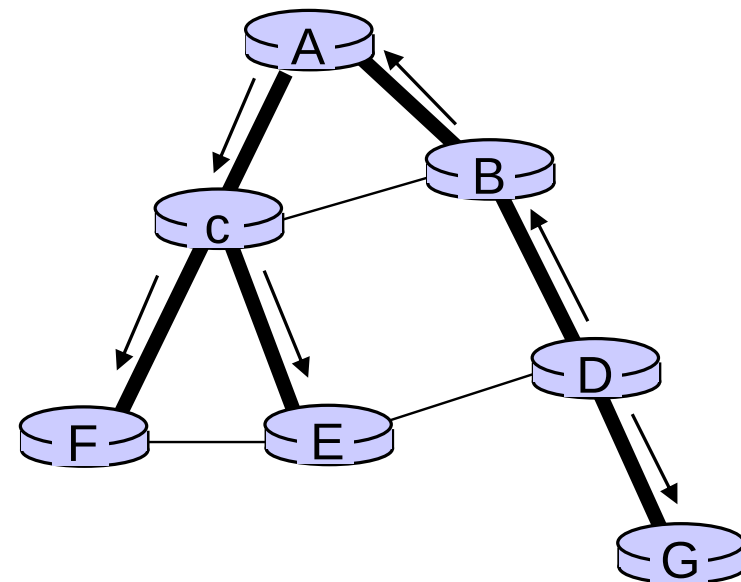


# Árvores Geradoras

- Primeiramente, construa uma árvore geradora.
  - Grafo acíclico conectando todos os nós.
- Os nós, então, encaminham/criam cópias dos pacotes apenas nos enlaces da árvore geradora.



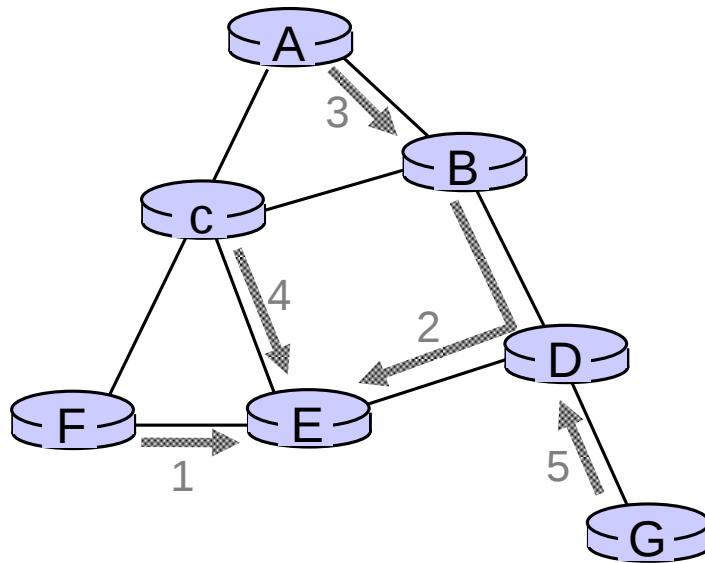
(a) broadcast iniciado em A



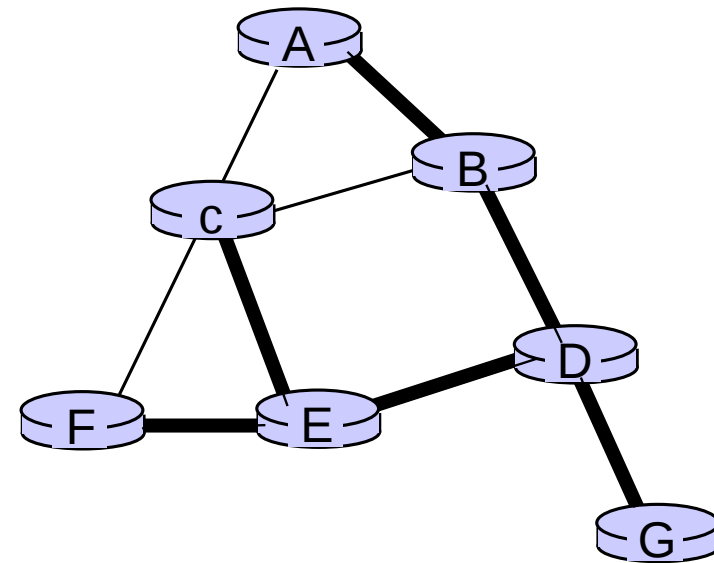
(b) broadcast iniciado em D

# Árvores Geradoras: Criação Distribuída

- Elege-se um nó central.
  - e.g., origem do tráfego *broadcast*.
- Cada nó envia uma mensagem *join* em *unicast* para o nó central.
  - Mensagem encaminhada normalmente até que chega a um nó que já pertence à árvore geradora.



(a) construção passo a passo da árvore geradora (centro: E)

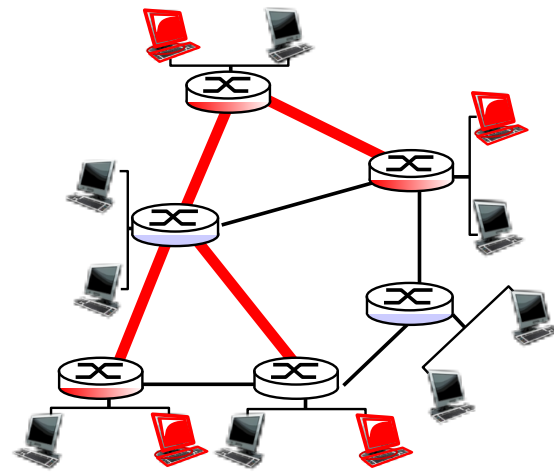


(b) árvore geradora construída

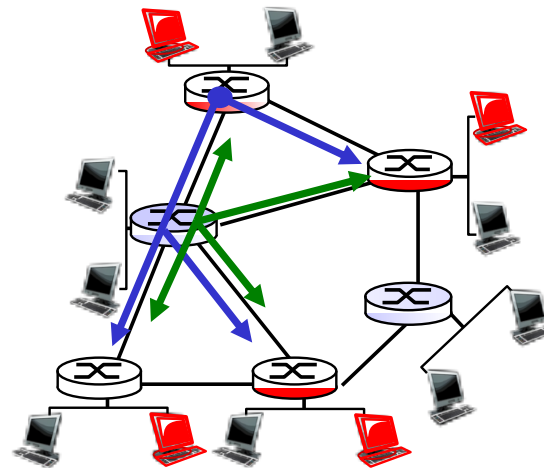
# Roteamento *Multicast*

# Roteamento Multicast: Definição do Problema

- **Objetivo:** encontrar uma árvore (ou árvores) conectando roteadores **membros de um grupo multicast**.
- Mais definições:
  - **Árvore:** nem todos os enlaces da rede são usados.
  - **Árvore Compartilhada:** uma mesma árvore é usada por todos os membros do grupo.
  - **Enraizada na Fonte:** árvores diferentes para cada transmissor do grupo.



árvore compartilhada



árvores enraizadas na fonte

## legenda



membro do grupo



não membro



roteador com membro do grupo



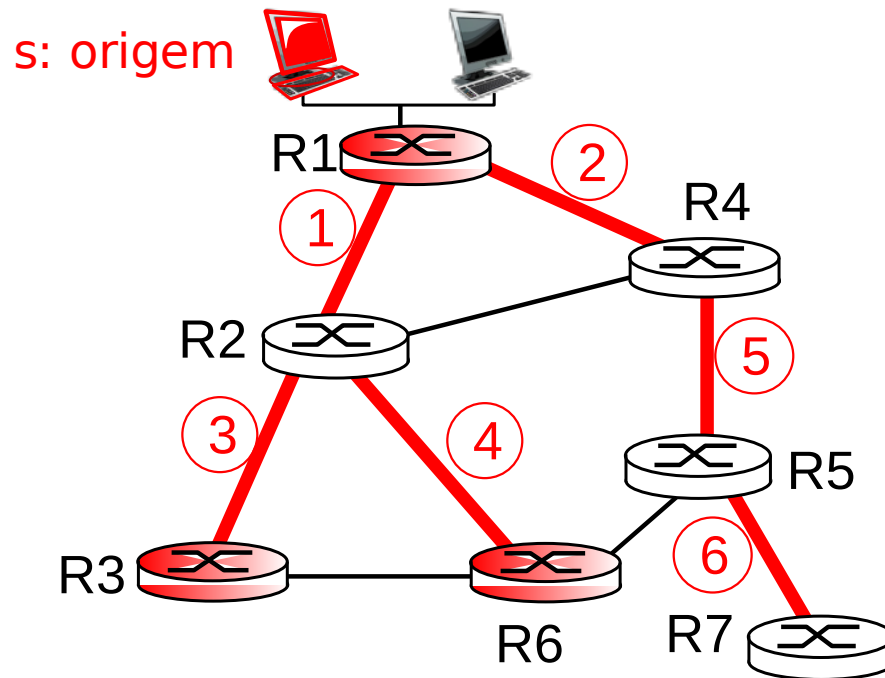
roteador sem membro do grupo

# Abordagens para Construção de Árvores Multicast


- **Árvores enraizadas na fonte:** uma árvore por fonte.
  - Árvore de caminhos mais curtos.
  - Encaminhamento baseado no caminho reverso.
- **Árvores compartilhadas:** grupo todo usa árvore única.
  - Árvore de custo mínimo (**Árvore de Steiner**).
  - Árvore baseadas em nó central.
- Veremos cada uma destas abordagens.


# Árvore de Caminhos mais Curtos


- Árvore de encaminhamento *multicast*: árvore composta pelos caminhos mais curtos da origem até cada destinatário.
  - Sub-produto do Algoritmo de Dijkstra.



## Legenda

 roteador que conecta membros do grupo

 roteador que não conecta membros do grupo

 enlace usado para encaminhamento, i indica quando foi adicionado

# Encaminhamento baseado no Caminho Reverso

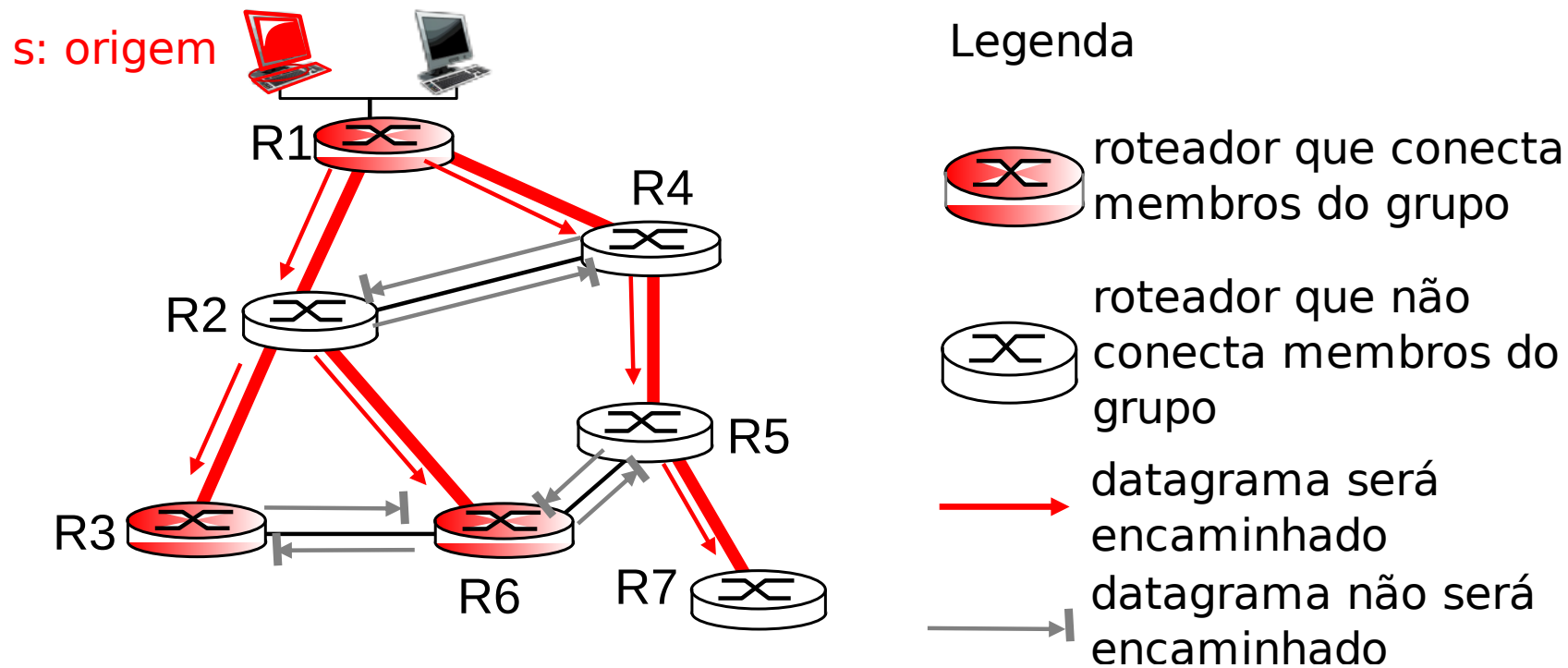
- Se baseia no conhecimento dos roteadores sobre os caminhos *unicast* mais curtos para a origem.
- Cada roteador aplica o seguinte algoritmo para realizar o encaminhamento:

**Se** (datagrama *multicast* foi recebido no enlace de próximo salto no caminho mais curto de volta à origem)

**Então** inunde datagrama para todos os enlaces de saída.

**Senão** ignore datagrama.

# Encaminhamento baseado no Caminho Reverso: Exemplo

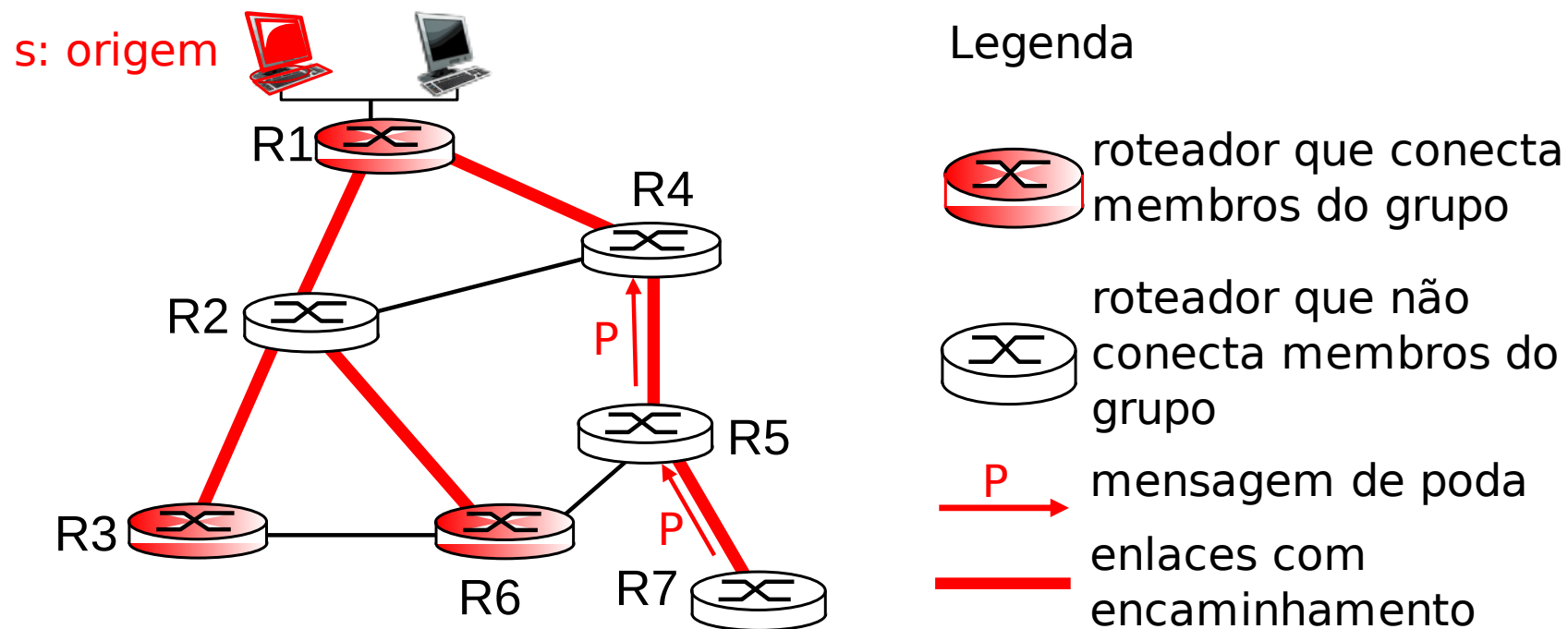


- Resultado é uma árvore específica para nó de origem.
  - Árvore de caminhos mais curtos reversa.
  - Pode não ser uma boa escolha se enlaces forem muito assimétricos.



# Encaminhamento baseado no Caminho Reverso: Poda

- Árvore de encaminhamento contém sub-árvores sem membros do grupo *multicast*.
  - Não é necessário encaminhar datagramas por estas sub-árvores.
  - Mensagens de “poda” enviadas em direção à origem por roteadores sem membros conectados.



# Árvore Compartilhada: Árvore de Steiner

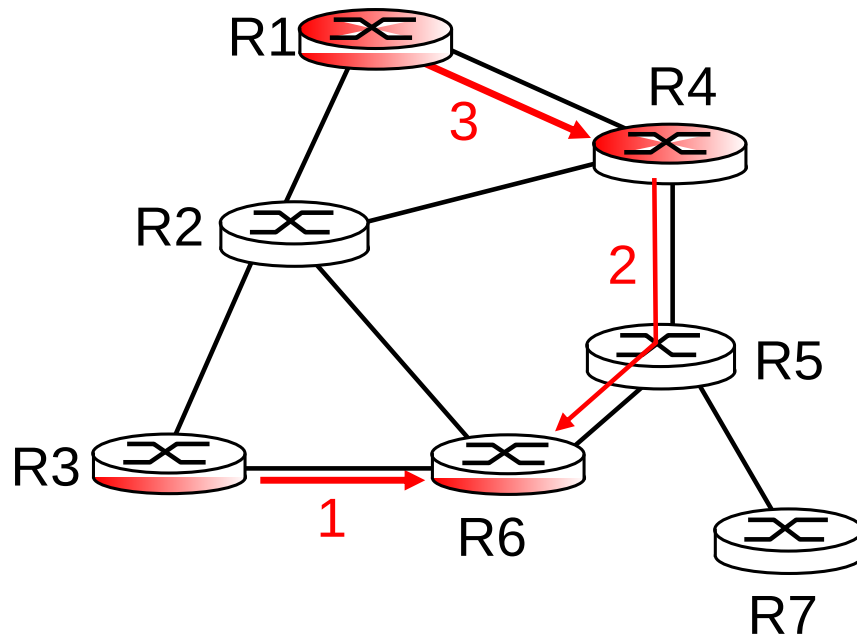
- **Árvore de Steiner:** árvore de custo mínimo conectando todos os roteadores que possuem membros do grupo.
- Problema é NP-Difícil.
  - *i.e.*, hoje não conhecemos algoritmos ótimos eficientes.
  - E é possível que não existam.
- Mas há excelentes heurísticas disponíveis.
- Mesmo assim, não é utilizado na prática:
  - Complexidade computacional.
  - Necessidade de informação sobre toda a rede.
  - Monolítico: re-execução é necessária sempre que um novo roteador entra/sai do grupo.

# Árvores Baseadas em Nó Central


- Uma única árvore de encaminhamento compartilhada por todos os nós.
- Um roteador se define como o “centro” da árvore.
- Para se juntar ao grupo:
  - Roteador de borda envia mensagem *unicast* do tipo *join* endereçada ao centro da árvore.
  - Mensagem é “processada” pelos roteadores intermediários e encaminhada em direção ao centro.
  - A mensagem de *join* ou chega ao centro ou chega a algum ramo já existente da árvore.
  - Caminho usado pela mensagem *join* se torna, então, um novo ramo da árvore conectando o novo roteador.


# Árvores Baseadas em Nó Central: Exemplo


- Suponha que R6 seja escolhido como centro.



## Legenda

 roteador que conecta membros do grupo

 roteador que não conecta membros do grupo

 ordem na qual as mensagens join foram geradas