

# Aula 21 - Roteamento *Broadcast* e *Multicast*

Diego Passos

Universidade Federal Fluminense

Redes de Computadores I

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

# Revisão da Última Aula...

- **Roteamento Intra-AS:**

- Critério: desempenho.
- Principais protocolos: OSPF, RIP, IGRP.
- Também conhecido como IGP.

- **RIP:**

- Vetor de Distâncias.
- Entradas para sub-redes.
- Métrica: número de saltos.
- Distância máxima: 16.
- Envenenamento reverso.
- Roda sobre UDP.

- **OSPF:**

- Estado de Enlace.
- Mais moderno que o RIP.
- Considera segurança, múltiplos caminhos, hierarquia, ...
- Roda diretamente sobre IP.

- **Roteamento Inter-AS:**

- Orientado a políticas.
- Principal protocolo: BGP.

- **BGP:**

- “Vetor de Caminhos”.
- Sessões BGP: TCP.
- Anúncios: **compromisso de rotear**.
- **eBGP** vs. **iBGP**.

- **BGP: rotas.**

- Possuem atributos.
- *e.g.*, AS-PATH.
- *e.g.*, NEXT-HOP.

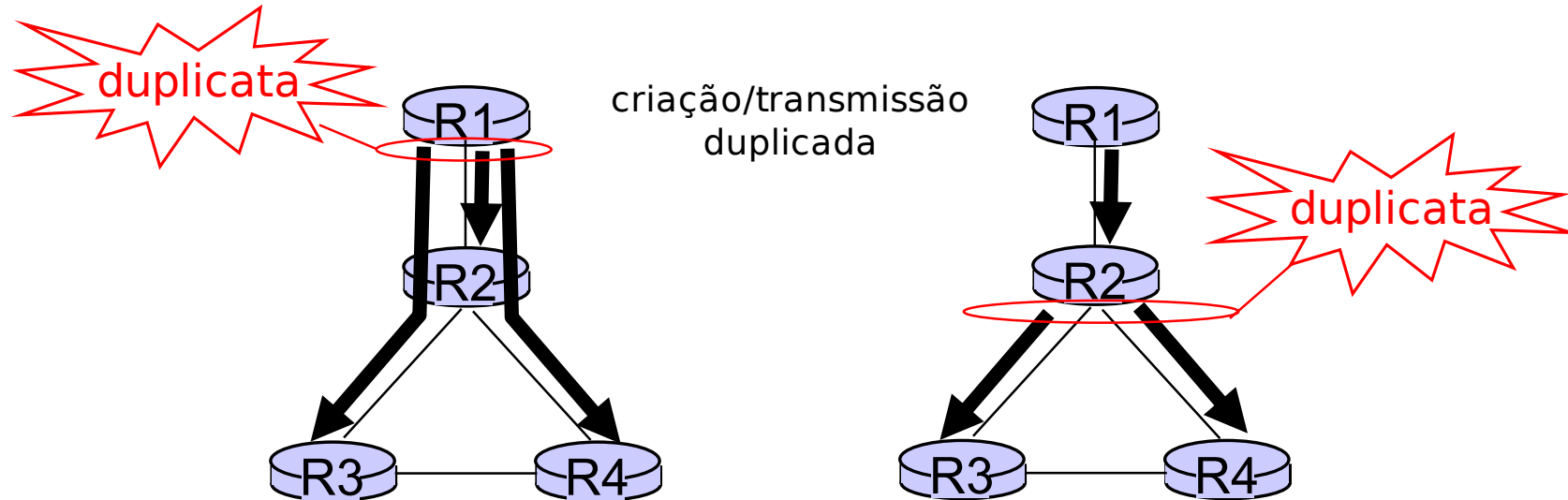
- **BGP: seleção de rotas.**

- **Baseada em políticas.**
- AS-PATH mais curto.
- NEXT-HOP mais próximo: **roteamento batata-quente**.
- Outros critérios.

# Roteamento *Broadcast*

# Roteamento Broadcast

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



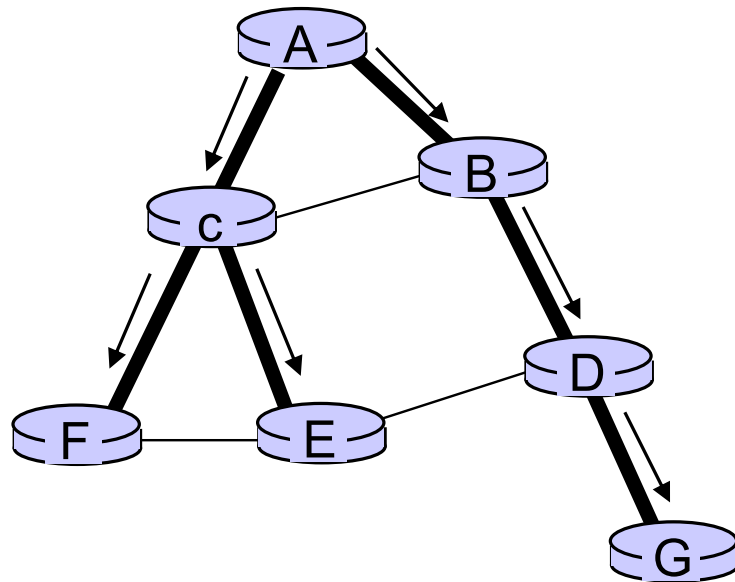
- Replicação na origem: como a origem determina o endereço dos destinatários?

# Replicação na Rede

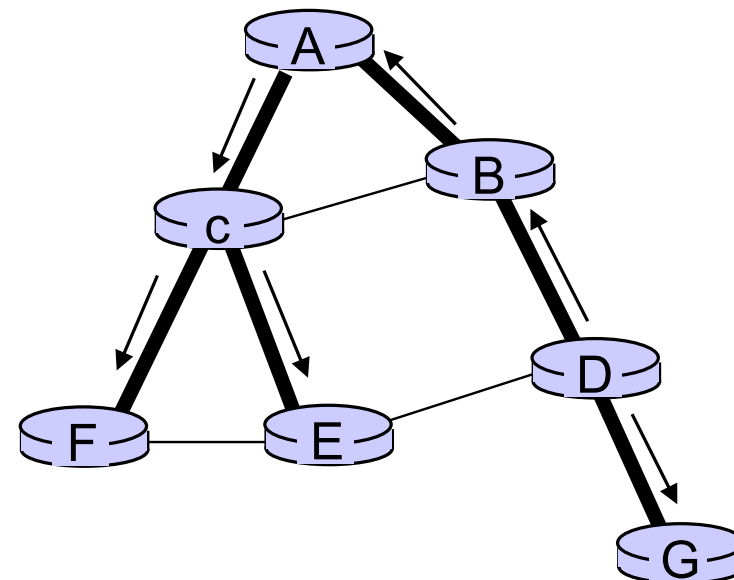
- **Inundação:** quando nó recebe pacote em *broadcast*, envia cópias para todos os seus vizinhos.
  - Problemas: ciclos e **tempestade de broadcast**.
- **Inundação Controlada:** nó só envia cópias se está é a primeira recepção deste pacote.
  - Nós mantêm lista dos IDs dos pacotes já replicados.
  - 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.



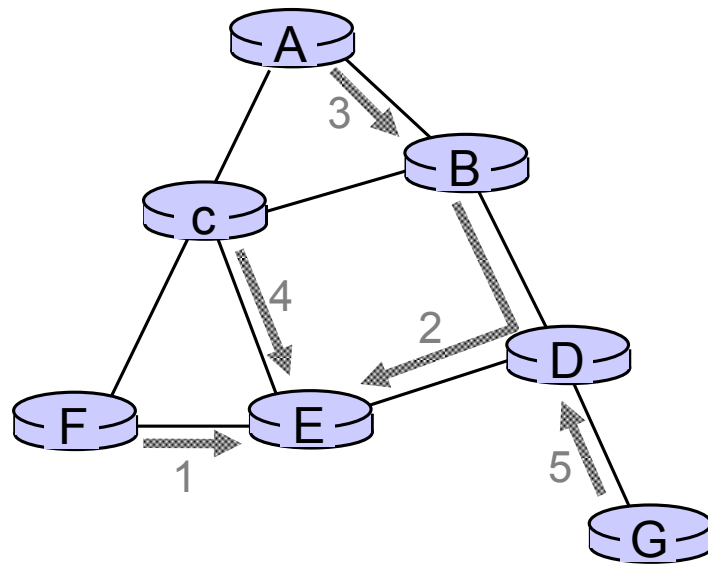
(a) broadcast iniciado em A



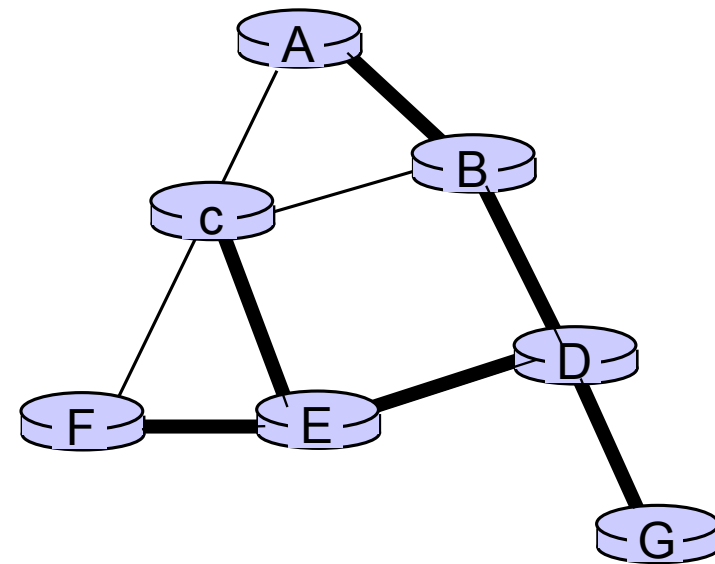
(b) broadcast iniciado em D

# 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)



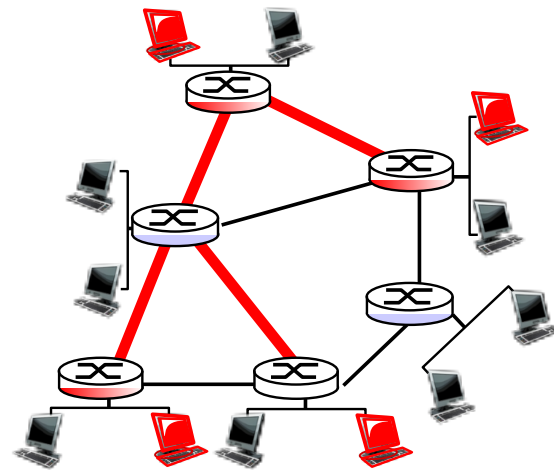
(b) árvore geradora construída

# Roteamento *Multicast*

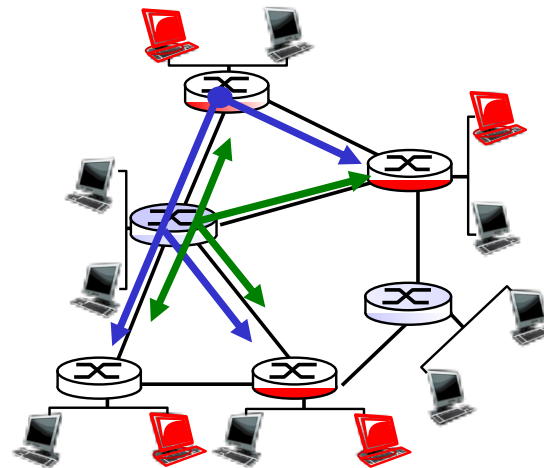


# Roteamento Multicast: Definição do Problema

- **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



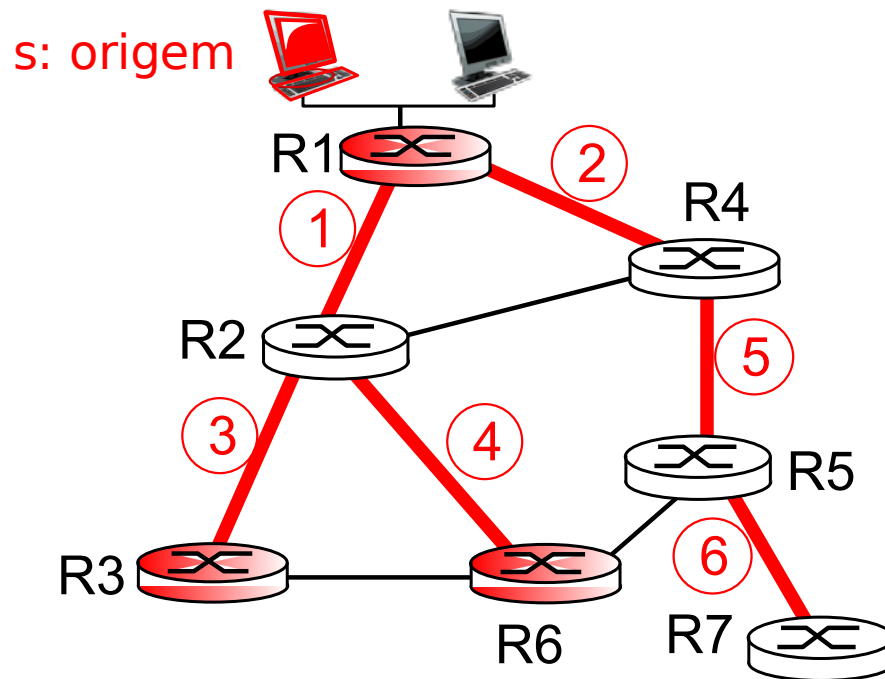
roteador sem membro do grupo

# 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

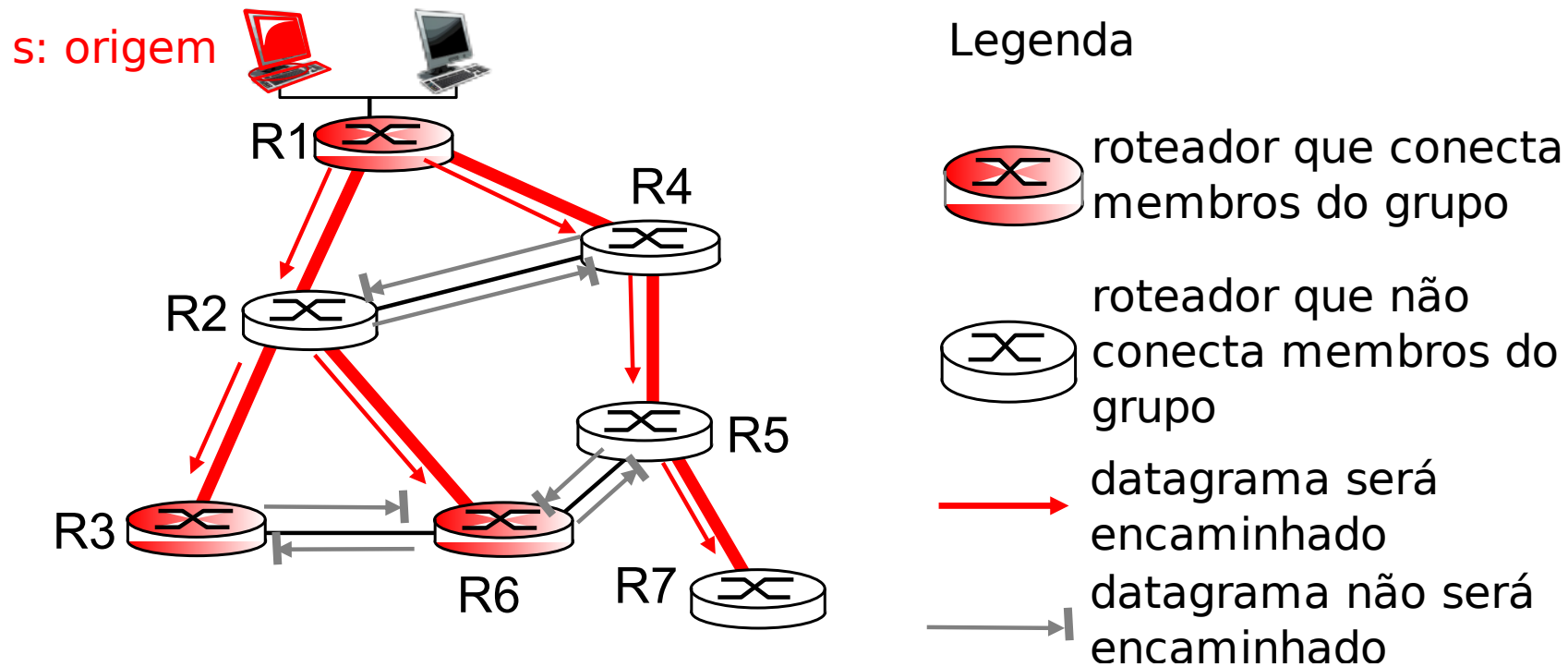
- Se baseia no conhecimento dos roteadores sobre os caminhos *unicast* mais curtos para a origem.
- Cada roteador aplica o seguinte algoritmo para realizar o encaminhamento:

**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.

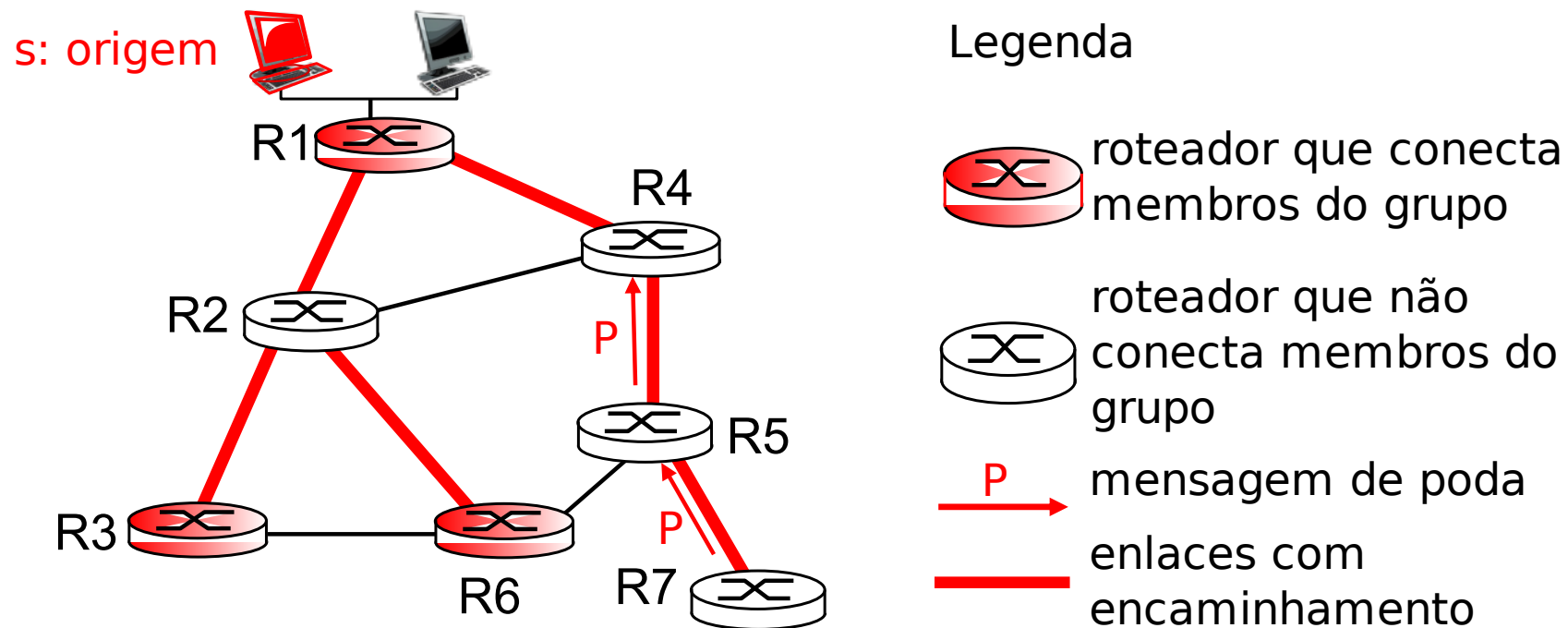
# Encaminhamento baseado no Caminho Reverso: Exemplo



- Resultado   uma  rvore espec fica para n  de origem.
  -  rvore de caminhos mais curtos reversa.
  - Pode n o ser uma boa escolha se enlaces forem muito assim tricos.

# 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.

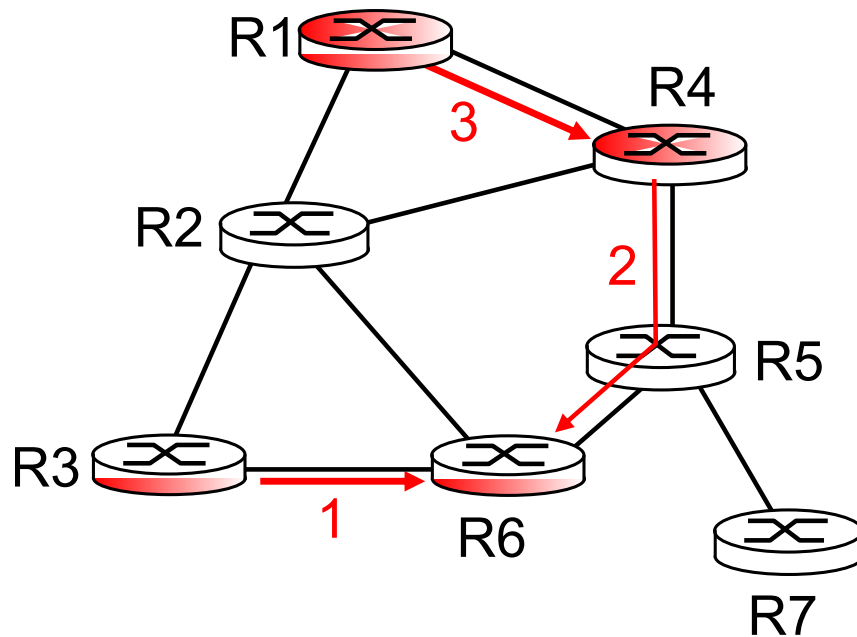
# Árvores Baseadas em Nó Central

- 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

- **Modos de Comunicação:**

- *Unicast*: com um nó de destino específico.
- *Broadcast*: com todos os nós da rede.
- *Multicast*: com um sub-conjunto (**grupo**) de nós da rede.

- **Roteamento Broadcast:**

- Evitar: replicação na fonte, tempestade de *broadcast*.
- Métodos: inundação controlada, árvore geradora (*spanning tree*).

- **Roteamento Multicast:**

- Objetivo: encontrar árvore(s) conectando nós do grupo.
- Abordagens: uma árvore por fonte *vs.* múltiplas árvores.
- Técnicas: árvore de caminhos mínimos, encaminhamento por caminho reverso, árvore de Steiner, árvores de nó central.
- Mecanismo auxiliar: mensagens de poda.

# (Breve) Revisão do Capítulo 4 (I)

- **Camada de rede:**

- Comunicação fim a fim **entre hosts**.
- Encapsula segmentos em **datagramas**.
- Roda em todos os nós.

- **Funções chave:**

- Encaminhamento: mover datagrama da entrada para saída.
- Roteamento: encontrar **rota** fim-a-fim.

- **Tabela de roteamento:**

- Montada pelo roteamento.
- Usada pelo encaminhamento.

- **Modelos de Serviço:**

- O que a rede promete.
- e.g., banda mínima, atraso máximo, entrega ordenada.
- Modelo da Internet: **melhor esforço**.

- **Redes de Circuitos Virtuais:**

- Serviço orientado a conexão.
- Pacotes carregam identificador do VC.
  - E não endereço de destino.
- Recursos muitas vezes reservados.
- Tabela de roteamento associa:
  - Interface, # do VC de entrada.
  - Interface, # do VC de saída.

- **Redes de datagramas:**

- Paradigma usado na Internet.
- Encaminhamento baseado no **endereço de destino**.
- **Agregação de endereço**.
- **Casamento por prefixo mais longo**.

# (Breve) Revisão do Capítulo 4 (II)

- **Roteadores: arquitetura.**

- Plano de controle: protocolos de roteamento.
- Plano de encaminhamento: portas, malha de comutação.

- **Portas de entrada:**

- Terminação do enlace físico.
- Camada de enlace.
- Decisões de comutação, **fila**.
- *Head-of-line Blocking*.

- **Malha de comutação:**

- Transporta pacotes de entradas para saída.
- **Memória, barramento, rede de interconexão.**
- Taxa de comutação.

- **Portas de saída:**

- **Fila:** atrasos, perdas.

- **Buffers: tamanho.**

- Objetivo: absorver variações temporárias.
- *Buffer* excessivo → atrasos altos, mascara congestionamento.

- **Políticas de enfileiramento:**

- Descarte e escalonamento.
- Impacto em QoS, desempenho do TCP.

- **Políticas de Escalonamento:**

- FIFO, *Priority Scheduling*, RR.
- Esfomeação vs. Justiça vs. Simplicidade.

- **Políticas de Descarte:**

- *Drop-tail*, *Drop-head*, RED.
- Sincronização.
- Detecção avançada de congestionamento.

# (Breve) Revisão do Capítulo 4 (III)

- **Protocolos da Camada de Rede:**

- Vários contribuem.
- Protocolos de roteamento.
- IP.
- ICMP.

- **Protocolo IP:**

- Define convenções.
- Formato de datagrama.
- Endereçamento.

- **Datagrama IP:**

- Checksum apenas do cabeçalho.
- Campo de opções, tamanho variável.
- TTL (*time-to-live*).

- **Fragmentação:**

- Quebrar datagramas grandes.
- Adequa a limitações de cada enlace.
- **Remontados apenas no destinatário.**

- **Endereçamento IP:**

- 32 bits.
- **Associados a interfaces.**
- **Prefixo identifica a sub-rede.**
- CIDR, máscara de sub-rede.

- **DHCP:**

- Protocolo de auto-configuração.
- Atribuição dinâmica de endereços IP.
- **Roteador de primeiro salto.**
- E mais configurações.
- Cliente-servidor.
- Roda sobre UDP.
- Mensagens em **broadcast**.

- **Endereçamento hierárquico:**

- Sub-redes são divididas.
- Novas sub-redes menores.
- Simplifica anúncio de rotas.

# (Breve) Revisão do Capítulo 4 (IV)

- **NAT:**

- Tradução de endereços.
- Rede local *vs.* rede externa.
- Endereços privados *vs.* públicos, roteáveis.
- Pacote sai: IP e porta de origem alterados.
- Pacote entra: IP e porta de destino são alterados.
- Tabela NAT: armazena mapeamentos.

- NAT: Motivação.

- **Escassez de IPs.**
- Independência dos endereços do ISP.
- Segurança.

- NAT *Traversal*:

- Conexão de fora para dentro do NAT.
- Entradas estáticas na tabela.
- Protocolo IGD.
- *Relaying* (aplicação).

- **ICMP:**

- Gerência do IP.
- Informações, condições de erro.
- Diversas tipos de mensagens.
- Suporte a algumas ferramentas usuais.

- **IPv6: Motivações.**

- Mais endereços.
- Menor *overhead* de processamento.

- **IPv6: diferenças.**

- Cabeçalho fixo.
- Fragmentação não permitida.
- Melhor suporte a QoS.
- ICMPv6.

- **IPv6: Transição.**

- Gradual, coexistência com IPv4.
- Solução: tunelamento.

# (Breve) Revisão do Capítulo 4 (V)

- **Roteamento:**
  - Encontrar caminhos fim-a-fim.
  - **Construir tabela de roteamento.**
    - Consultada no encaminhamento.
- **Grafos:** usados como abstração para representar a rede.
  - Roteadores são nós.
  - Enlaces são arestas.
  - Podem ter pesos: **medida de qualidade do enlace.**
    - Relacionado a banda, atraso, congestionamento, ...
- **Classificações:**
  - Estado de Enlace vs. Vetor de Distâncias.
  - Dinâmico vs. Estático.
- **Roteamento baseado em Estado de Enlaces:**
  - Algoritmo de Dijkstra.

# (Breve) Revisão do Capítulo 4 (VI)

- **Roteamento baseado em Vetor de Distâncias:**

- Ideia: melhor caminho até destino composto por enlace até vizinho e melhor caminho do vizinho até destino.
- Nós anunciam suas **estimativas** de custo até cada destino.
- Ao **receber** novas estimativas, nó **atualiza** suas próprias.
- Processo iterativo, **converge para melhores rotas**.
- Algoritmo **distribuído**: nós precisam conhecer apenas vizinhança.

- **Contagem ao infinito:**

- Potencial problema, ocorre em caso de grandes pioras nos custos dos enlaces.
- Solução (parcial): **envenenamento reverso**.

- **Roteamento Hierárquico:**

- Dois níveis: dentro e fora de **Sistemas Autônomos**.
  - Intra-AS e Inter-AS.
- Tabela de roteamento construída por **colaboração dos dois processos**.
- Reduz escopo, complexidade do roteamento.
- **Nem sempre é globalmente ótimo!**

- **Roteamento batata-quente:**

- Tirar datagrama do AS o mais rápido possível.



# (Breve) Revisão do Capítulo 4 (VII)

- **Roteamento Intra-AS:**

- Critério: desempenho.
- Principais protocolos: OSPF, RIP, IGRP.
- Também conhecido como IGP.

- **RIP:**

- Vetor de Distâncias.
- Entradas para sub-redes.
- Métrica: número de saltos.
- Distância máxima: 16.
- Envenenamento reverso.
- Roda sobre UDP.

- **OSPF:**

- Estado de Enlace.
- Mais moderno que o RIP.
- Considera segurança, múltiplos caminhos, hierarquia, ...
- Roda diretamente sobre IP.

- **Roteamento Inter-AS:**

- Orientado a políticas.
- Principal protocolo: BGP.

- **BGP:**

- “Vetor de Caminhos”.
- Sessões BGP: TCP.
- Anúncios: **compromisso de rotear**.
- **eBGP** vs. **iBGP**.

- **BGP: rotas.**

- Possuem atributos.
- *e.g.*, AS-PATH.
- *e.g.*, NEXT-HOP.

- **BGP: seleção de rotas.**

- **Baseada em políticas.**
- AS-PATH mais curto.
- NEXT-HOP mais próximo: **roteamento batata-quente**.
- Outros critérios.

# (Breve) Revisão do Capítulo 4 (VIII)

- **Modos de Comunicação:**

- *Unicast*: com um nó de destino específico.
- *Broadcast*: com todos os nós da rede.
- *Multicast*: com um sub-conjunto (**grupo**) de nós da rede.

- **Roteamento Broadcast:**

- Evitar: replicação na fonte, tempestade de *broadcast*.
- Métodos: inundação controlada, árvore geradora (*spanning tree*).

- **Roteamento Multicast:**

- Objetivo: encontrar árvore(s) conectando nós do grupo.
- Abordagens: uma árvore por fonte *vs.* múltiplas árvores.
- Técnicas: árvore de caminhos mínimos, encaminhamento por caminho reverso, árvore de Steiner, árvores de nó central.
- Mecanismo auxiliar: mensagens de poda.

# Próxima Aula...

- Aulas de conteúdo acabaram!
- Daqui até o fim do período, haverá apenas avaliações, vistas e entregas de notas.