

Redes de Computadores II

EEL 879

Parte V

Roteamento Multicast na Internet

Luís Henrique M. K. Costa

luish@gta.ufrj.br

Universidade Federal do Rio de Janeiro - PEE/COPPE
P.O. Box 68504 - CEP 21945-970 - Rio de Janeiro - RJ
Brasil - <http://www.gta.ufrj.br>

Introdução

- **Comunicação de grupo (aplicações multi-destinatárias)**
 - Vídeo-conferência
 - Ensino a distância
 - Jogos distribuídos
 - TV na Internet, ...
- **A mesma informação deve ser enviada a múltiplos receptores**

Como enviar a N receptores?

- **Opções: diferentes tipos de transmissão**

- **Unicast**

- Transmissão ponto-a-ponto
- 1 emissor, 1 receptor

- **Multicast**

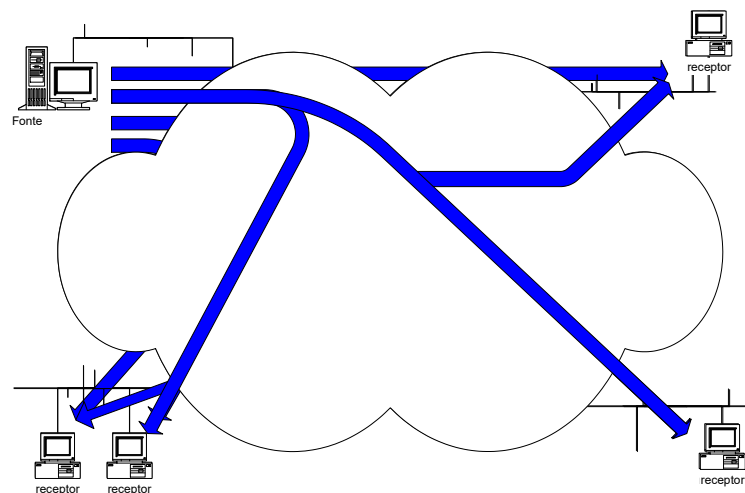
- Transmissão ponto-a-multiponto
- 1 emissor, N receptores

- **Broadcast**

- Envio a todos os nós da rede

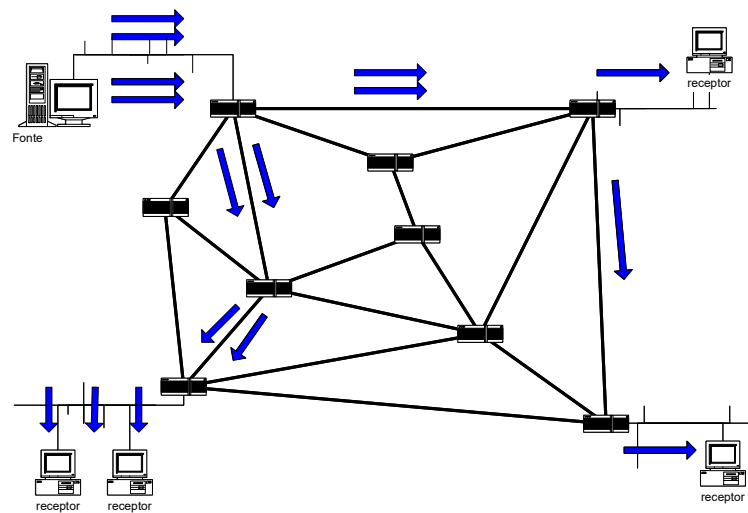
GTA/UFRJ

Unicast x Multicast



GTA/UFRJ

Unicast x Multicast



GTA/UFRJ

Utilização do Multicast

○ Vantagens

- Produz menos pacotes
 - Utilização eficiente da banda passante da rede
 - Menor processamento em estações e roteadores

GTA/UFRJ

Utilização do Multicast

○ Problemas

- Como identificar o grupo?
 - Lista dos receptores
 - Overhead de cabeçalho limita o tamanho do grupo
 - Endereço de grupo
 - Identidade e número dos receptores desconhecidos
- Como realizar a distribuição dos pacotes?
 - Endereçamento e roteamento (encaminhamento dos pacotes) são **diretamente** relacionados

GTA/UFRJ

Endereçamento na Internet

○ endereço IP = inteiro de 32 bits

- escrito na forma de 4 números decimais separados por pontos: **146.164.69.2**
- o mapeamento de nomes em endereços IP e vice-versa é feito pelo Sistema de Nome de Domínio (DNS)
- atribuído a cada interface de rede de uma máquina
 - identifica a conexão de uma estação na rede

○ endereçamento IP

- topológico (ou **hierárquico**: utiliza **prefixos**)
 - a posição de uma máquina determina seu endereço
 - torna eficaz as operações de roteamento

GTA/UFRJ

Endereçamento IP

- Inicialmente, baseado em 3 classes de endereços: A, B e C
- Endereço IP decomposto em
 - identificador de rede (*netid*) ou prefixo de rede
 - identifica uma rede
 - identificador da estação dentro desta rede (*hostid*)
 - identifica uma máquina numa rede

GTA/UFRJ

Classes A, B e C



- 0 – 127 - 128 prefixos
- $(2^{24} - 2)$ estações em cada rede



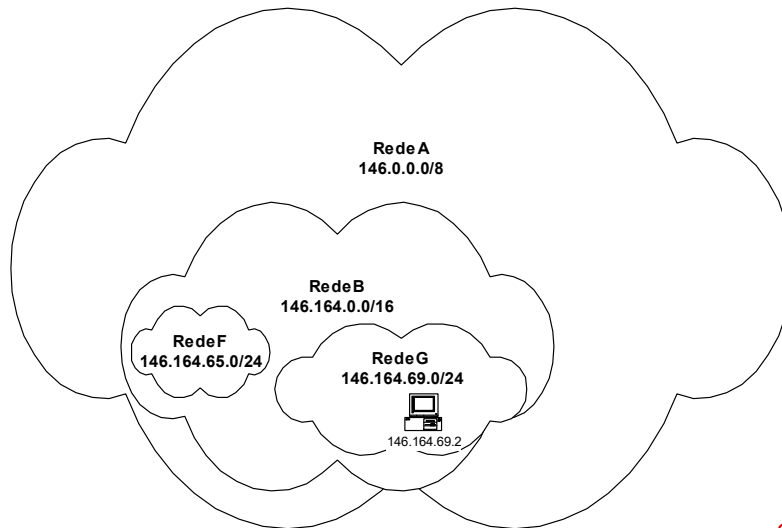
- 128 – 191 - 2^{14} prefixos de classe B
- $(2^{16} - 2)$ estações em cada rede



- 192 – 223 - 2^{21} prefixos de classe C
- $(2^8 - 2)$ estações em cada rede

GTA/UFRJ

Endereçamento Hierárquico



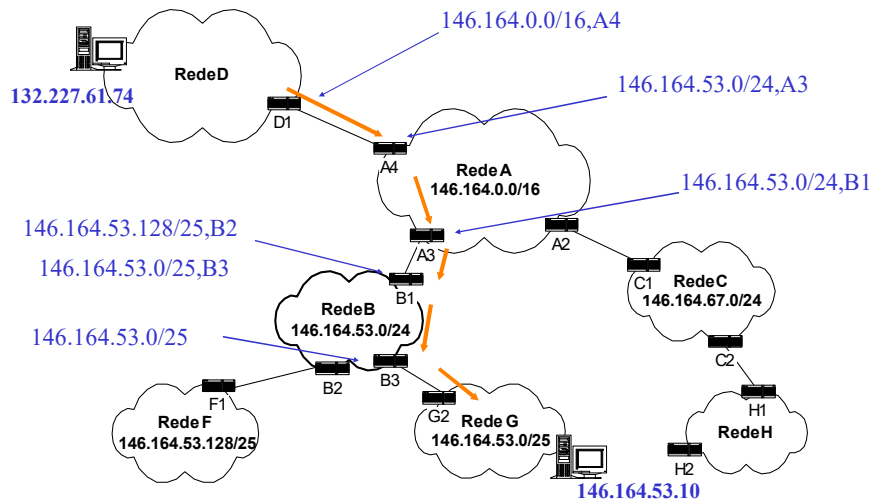
GTA/UFRJ

Endereçamento Hierárquico

- **Hoje em dia**
 - CIDR (*Classless Inter-domain routing*)
 - Prefixos podem ter comprimento arbitrário
 - Redes podem ter tamanho arbitrário
- **Rota:**
 - Dado um destino, qual o próximo salto? (qual a “porta de saída”?)
- **Roteamento:**
 - Escolher a melhor rota
- **Roteamento Hierárquico:**
 - Agregação de rotas

GTA/UFRJ

Agregação de Rotas



GTA/UFRJ

Problema do Multicast

- Dado o endereçamento, como realizar a distribuição dos pacotes?
 - Endereço unicast
 - Identifica e localiza uma estação
 - Endereço de grupo
 - Hierarquia impossível, receptores espalhados em toda a rede

GTA/UFRJ

Modelo de Serviço IP Multicast

○ Identificação

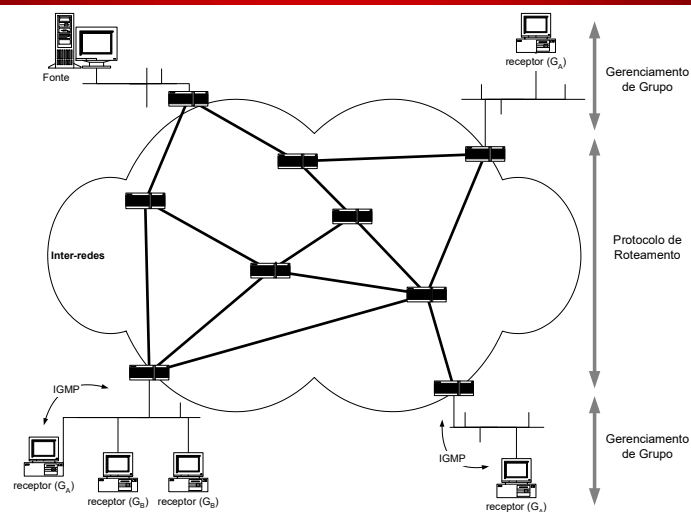
- Endereço de grupo

○ Distribuição dos dados

- Gerenciamento de grupo
 - Entrada / saída do grupo
 - “quero escutar o grupo” / “quero parar de escutar o grupo”
 - Entre a estação e seu roteador local
- Protocolos de roteamento
 - Distribuição dos dados entre as redes
 - Como fazer os pacotes chegarem ao meu roteador local?

GTA/UFRJ

Modelo de Serviço



GTA/UFRJ

Modelo de Serviço

- **Grupo**
 - **Identificado por um endereço de grupo**
 - **Conversação N x M, aberta**
 - Qualquer estação pode participar
 - Uma estação pode pertencer a vários grupos
 - Uma fonte pode enviar dados ao grupo, tendo se inscrito neste ou não
 - **O grupo é dinâmico**, uma estação pode entrar e sair a qualquer instante
 - **O número e a identidade** dos participantes do grupo são desconhecidos

GTA/UFRJ

[illegible]

Modelo de Serviço

- **O grupo é identificado por um endereço IP Multicast**
 - Endereço IP Classe D
- **Criação do grupo**
 - Escolha de um **endereço multicast** e envio de dados para o grupo
- **Destruição do grupo**
 - Parada do envio de dados

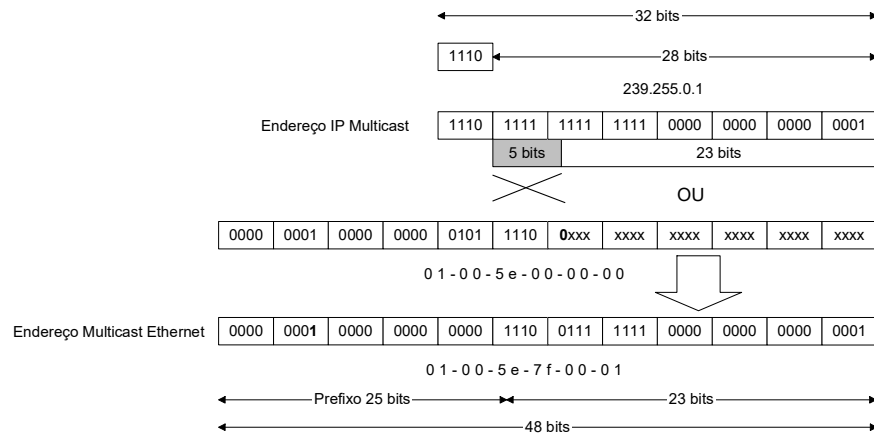
GTA/UFRJ

Conexão a um Grupo Multicast

- **A aplicação sinaliza à camada rede interesse no grupo G**
 - `socket`
- **Se não havia outra aplicação conectada a G**
 - Relatório IGMP é enviado na rede local
 - Camadas inferiores podem ser igualmente programadas
 - Ex. Ethernet

GTA/UFRJ

Multicast Ethernet



- 28 bits IP são mapeados em 23 bits Ethernet
- 32 endereço IP multicast = 1 endereço multicast Ethernet

GTA/UFRJ

Por que apenas 23 bits?

- No início da década de 90, Steve Deering desejava que o IEEE alocasse 16 OUIs (*Organizational Unique Identifier*) para os endereços multicast Ethernet.
- Cada OUI equivale a 24 bits de espaço de endereçamento
 - 16 OUIs consecutivos = 28 bits
- Na época, 1 OUI = US\$ 1.000,00
- Jon Postel (chefe de Deering na época) comprou apenas 1 OUI, e liberou apenas a metade do espaço para as pesquisas de Deering...

GTA/UFRJ

Gerenciamento de Grupo

- **Quem quer ouvir que grupos?**

- “estação de rádio”

- **IGMP (*Internet Group Management Protocol*)**

- Detecção de estações interessadas em grupos multicast
- Existem 4 versões do IGMP

- **Escopo local**

- diálogo entre a estação e o primeiro roteador
- criação da árvore de distribuição independente do IGMP

GTA/UFRJ

Funcionamento do IGMP

- **Parte estação**

- Conexão ao grupo (`join(G)`)
 - Receptor envia mensagem `report(G)`
- Envio de mensagens `report` em resposta às mensagens `query`
 - “Estes são os grupos de interesse desta estação”

- **Parte roteador**

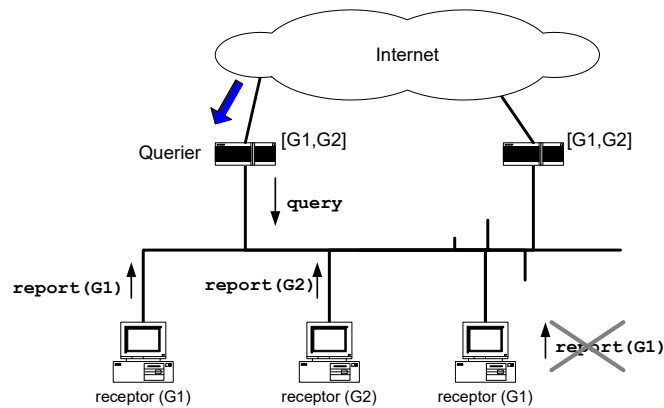
- Envio periódico de mensagens `query`
 - “Que grupos são escutados na rede?”

- **Parte estação**

- Mecanismo de supressão de mensagens `report`

GTA/UFRJ

Funcionamento do IGMP



GTA/UFRJ

IGMPv2

- **Introduz o mecanismo de *fast-leave***
 - Diminuição da latência de desconexão
- **Desconexão**
 - Receptor envia mensagem IGMP **leave (G)**
- **Regras de processamento para evitar a desconexão de outras estações**
 - Ex. roteador deve enviar **query (G)** para detectar se existem ouvintes remanescentes

GTA/UFRJ

IGMPv3

- **Filtragem de fontes**
- **A estação anuncia o interesse no grupo G ,**
 - “apenas nos dados enviados por determinadas fontes”, ou
 - “nos dados enviados por todas, exceto determinadas fontes”
- **Interface**
 - `IPMulticastListen (socket, interface, mcast-address, filter mode, source-list)`
 - `filter-mode` **pode ser** `INCLUDE` **ou** `EXCLUDE`

GTA/UFRJ

Exemplo no IGMPv3

- **Recepção do que apenas as fontes S1 e S2 enviam a G**
 - `IPMulticastListen (sock, iface, G, INCLUDE, {S1,S2})`
- **Recepção de tudo que é enviado a G, exceto por S2 e S3**
 - `IPMulticastListen (sock, iface, G, EXCLUDE, {S2,S3})`
- **Estado no roteador**
 - `(G, EXCLUDE{S3})`

GTA/UFRJ

Roteamento Multicast

- **Problema de Roteamento Multicast**
- **$G = (V, E)$**
 - V conjunto de vértices
 - E conjunto de enlaces
- **M sub-conjunto de V**
 - inclui fontes e receptores do grupo multicast
- **Problema: construir uma, ou várias, topologias de interconexão, árvores, que incluem todos os nós em M**
 - árvore por fonte (*source-based tree*)
 - árvore compartilhada (*shared tree*)

GTA/UFRJ

Primeiras Soluções

- **Árvores de cobertura (*spanning trees*)**
- **Algoritmo de inundação**
- **Árvores RPF (*Reverse Path Forwarding*)**
- **Árvores centradas**

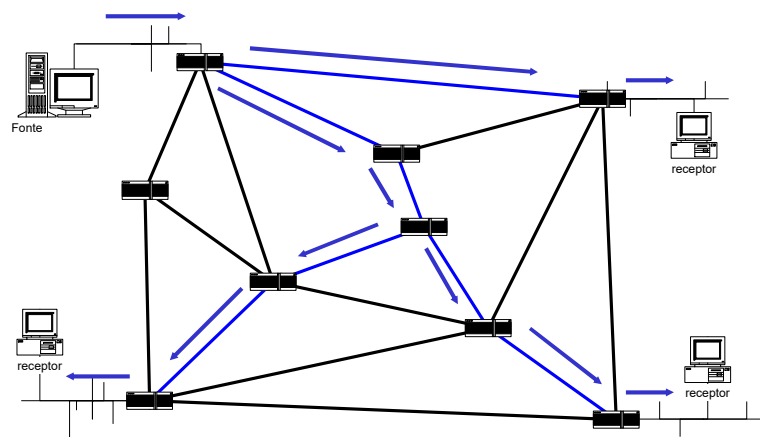
GTA/UFRJ

Árvores de Cobertura

- Sub-grafo contendo todos os nós em M , sem ciclos
- Pode-se adicionar objetivo de custo mínimo
 - Associa-se um custo, c_{uv} , a cada enlace (u,v)
- Se $c_{uv} = 1 \ \forall u,v$, árvore de Steiner
 - Problema NP-completo

GTA/UFRJ

Árvores de Cobertura



GTA/UFRJ

Inundação

○ Ao receber o pacote

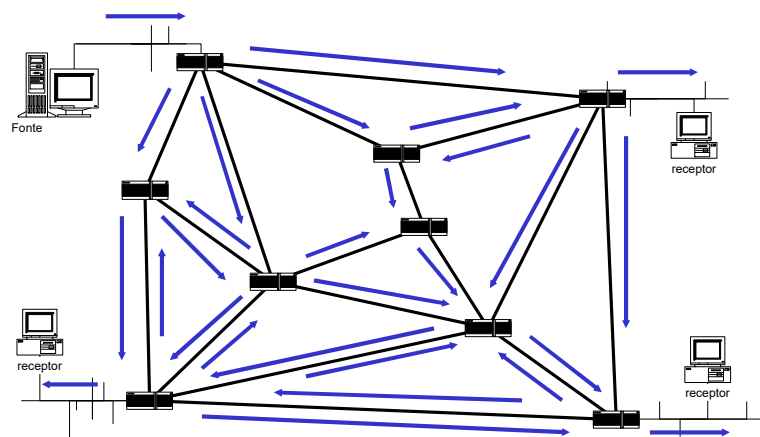
- Esta é a primeira vez que foi recebido?
 - Se sim, re-envio em todas as interfaces de saída
 - Se não, descarte

○ Problema

- Como identificar o primeiro envio de um pacote
 - Armazenar identificação
 - Carregar lista dos nós atravessados
- Consumo de memória e banda passante

GTA/UFRJ

Inundação



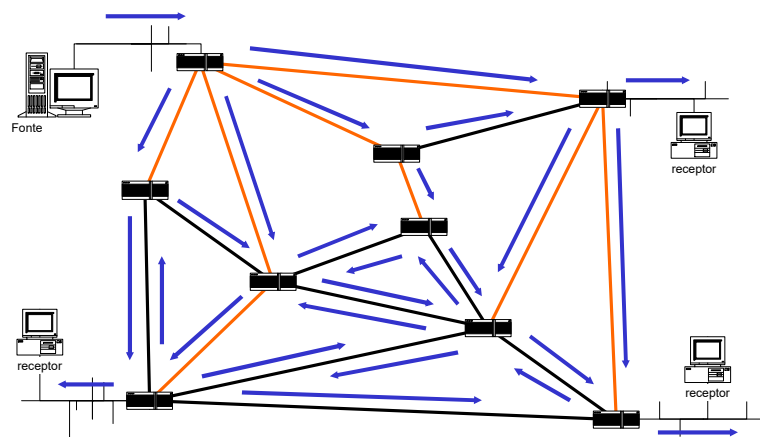
GTA/UFRJ

Árvores RPF

- Hipótese: um roteador **R** conhece o caminho mais curto para ir à fonte, **S**
- *Reverse Path Forwarding check (RPF check)*
- **Reverse Path Broadcasting**
 - O roteador **R** recebe um pacote da fonte **S**
O pacote chegou pela interface utilizada por **R** para ir à **S**? (RPF check)
Se sim, enviar o pacote por todas as interfaces de saída.
Se não, descartar o pacote.

GTA/UFRJ

Reverse Path Broadcasting



GTA/UFRJ

Reverse Path Forwarding

○ Hipótese

- um roteador **R** sabe se seu vizinho o utiliza como caminho para a fonte, **S**

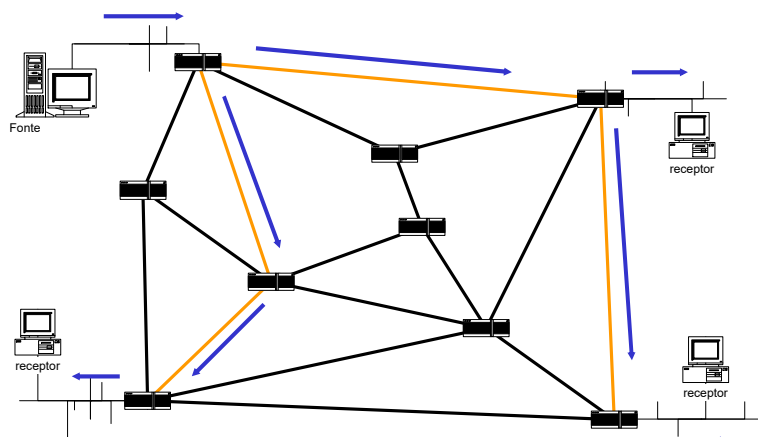
○ Como obter esta informação

- trivial, se protocolo de estado do enlace
- se protocolo de vetor-distância
 - mensagem adicional para alertar o roteador “pai”, ou
 - mensagem de poda para eliminar a rota reversamente

○ Informação por (fonte, grupo)

GTA/UFRJ

Árvore RPF



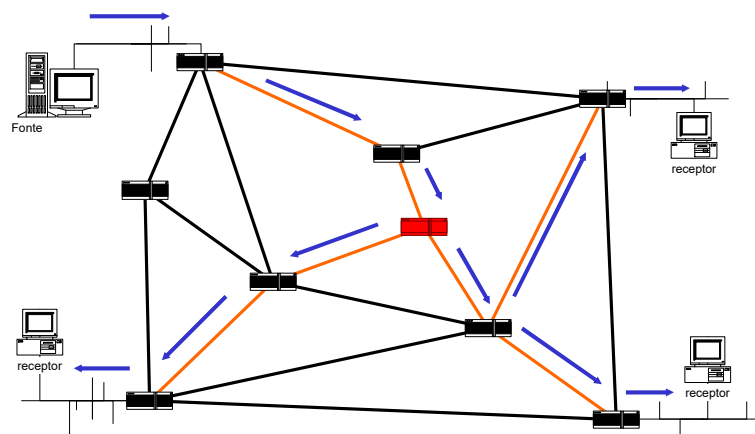
GTA/UFRJ

Árvores Centradas

- **Construída a partir de um nó central (core)**
- **Compartilhada por diversas fontes**
 - diversas fontes utilizam o mesmo *core*
 - “pedidos de conexão” são enviados ao *core*

GTA/UFRJ

Árvores Centradas



GTA/UFRJ

Roteamento Multicast Intra-domínio

- **DVMRP (*Distance Vector Multicast Routing Protocol*)**
 - Primeiro protocolo utilizado no MBone
- **MOSPF (*Multicast Open Shortest Path First*)**
- **CBT (*Core Based Trees*)**
- **PIM (*Protocol Independent Multicast*)**
 - PIM-DM (PIM *Dense-Mode*)
 - PIM-SM (PIM *Sparse Mode*)
 - PIM-SSM (PIM Source Specific Multicast)

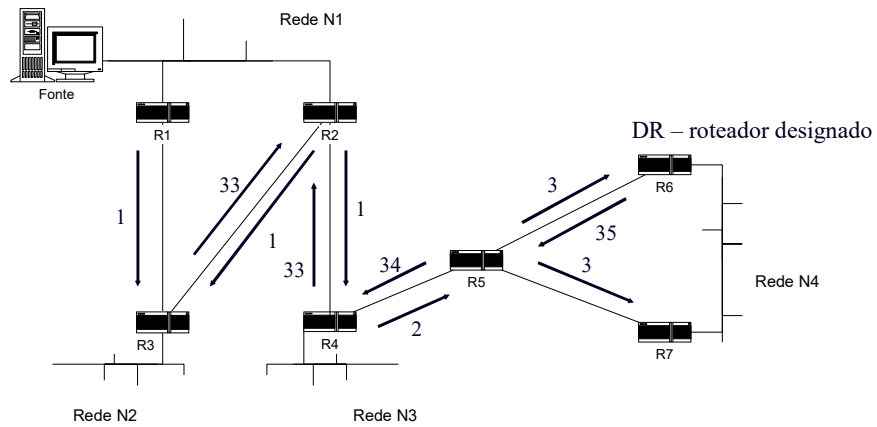
GTA/UFRJ

DVMRP

- **Utiliza vetores de distância**
 - Semelhante ao RIP (*Route Information Protocol*)
 - Constrói rotas **unicast** para cada fonte multicast
 - *Poison-reverse* especial utilizado para marcar interfaces filhas
- **Distribuição de dados**
 - Inundação e poda (*flood-and-prune*)
 - Teste RPF baseado em sua tabela de roteamento unicast
- **A inundação é periódica**
 - Descoberta de fontes ativas

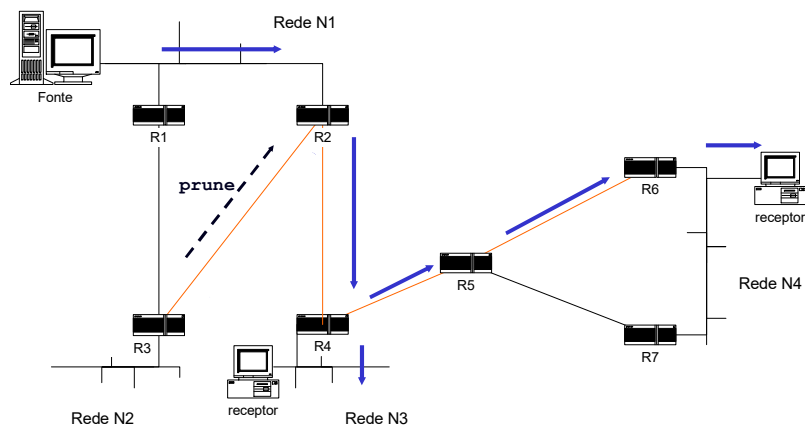
GTA/UFRJ

Funcionamento do DVMRP



GTA/UFRJ

Envio de Dados no DVMRP



GTA/UFRJ

DVMRP

- Algoritmo simples
- Protocolo de roteamento unicast *próprio*
- Inundação periódica da rede com *dados*
- Vetores-de-distância
 - Convergência lenta, como no RIP

GTA/UFRJ

MOSPF

- Extensão do OSPF (*Open Shortest Path First*)
 - roteadores trocam mensagens de estado-do-enlace
 - LSA – *Link State Advertisement*
 - Cada nó possui a topologia atualizada da rede
 - Algoritmo de Dijkstra – caminhos mais curtos
- Novo tipo de LSA anuncia receptores multicast
- A árvore de distribuição é uma SPT (*Shortest-Path Tree*)
 - união dos caminhos mais curtos entre fonte e cada receptor

GTA/UFRJ

MOSPF

○ Estrutura hierárquica

- Áreas OSPF (roteamento intra-área e inter-área)

○ Intra-área

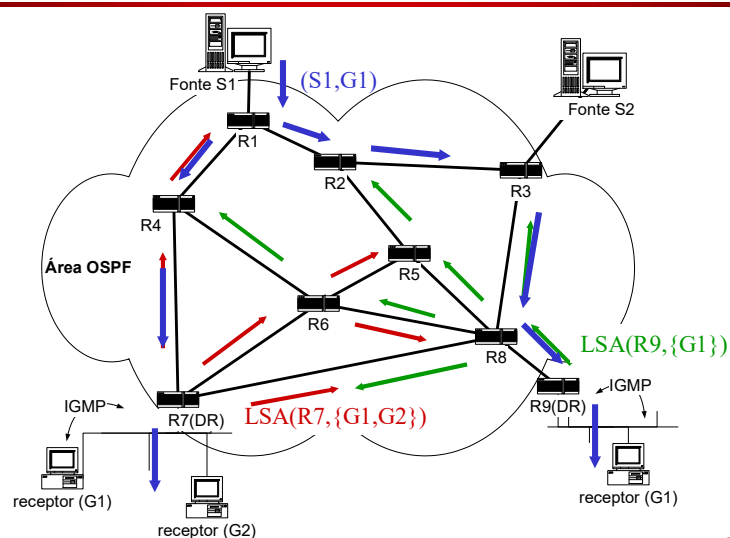
- IGMP – descoberta de receptores
- *Group Membership LSAs*
 - (roteador, grupo multicast, lista de interfaces)

○ Cálculo da SPT

- Disparado apenas após recepção do primeiro pacote de dados
- Diminui o custo computacional

GTA/UFRJ

MOSPF Intra-área



GTA/UFRJ

MOSPF Inter-área

○ Multicast Area Border Router (MABR)

- Envio de tráfego multicast
- Informação sobre os grupos multicast
- Conecta uma área OSPF à área 0 (área *backbone*)

○ Receptor coringa

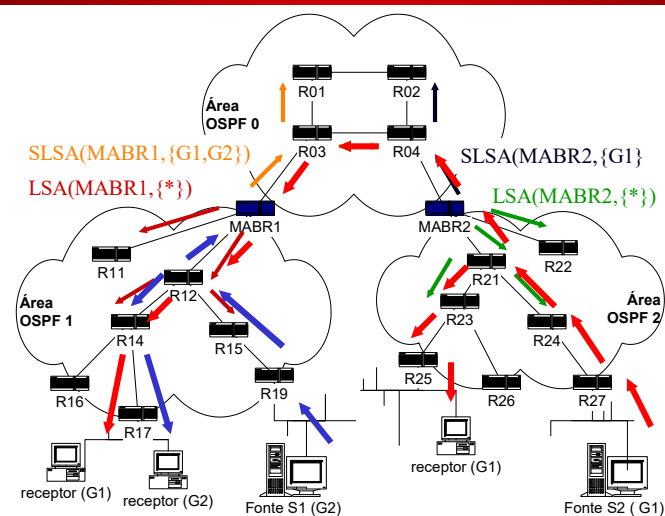
- LSA anuncia que o roteador possui receptores para *todos* os grupos
- Todos os MABRs em uma área são receptores coringa
 - Injetam LSAs coringa na área OSPF
 - Recebem todo o tráfego e o re-enviam na área 0 se necessário

○ LSA de Resumo de Grupos (*Summary Membership LSA*)

- Lista todos os grupos escutados em uma área
- São injetadas na área 0 pelos MABRs

GTA/UFRJ

MOSPF Inter-área



GTA/UFRJ

MOSPF Inter-área

- **Árvore SPT é construída na área 0**
- **A árvore completa (áreas comuns + área 0) não é SPT**
- **Pode haver envio desnecessário de tráfego ao MABR**
 - Receptor coringa

GTA/UFRJ

MOSPF

- **Protocolo de roteamento unicast deve ser OSPFv2**
- **Mensagens de estado-do-enlace**
 - evitam a inundação periódica de dados como no DVMRP
 - porém impedem o uso do OSPF em redes muito grandes
 - LSAs inundam toda a rede
- **DVMRP**
 - Dados são uma mensagem **implícita** sobre a localização dos receptores
- **MOSPF**
 - Mensagem **explícita** sobre onde existem receptores

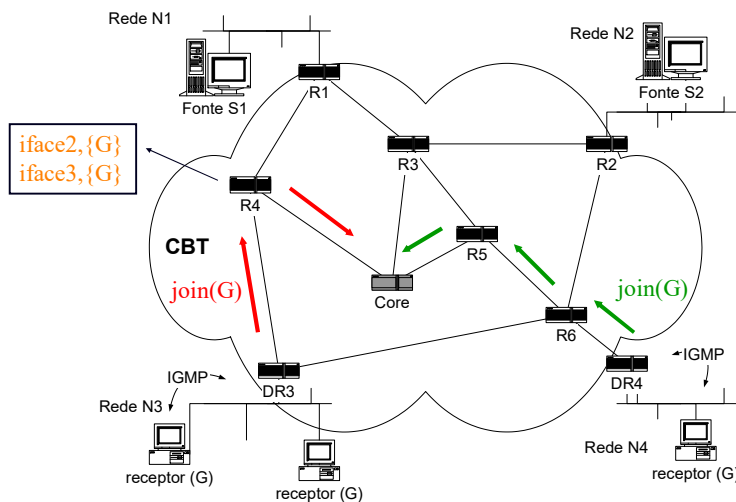
GTA/UFRJ

CBT

- **Utiliza árvores centradas**
 - Compartilhadas e bi-direcionais
- **Roteador central – core**
- **Construção da árvore**
 - Mensagens *join*
 - Enviadas pelos receptores na direção do **core**

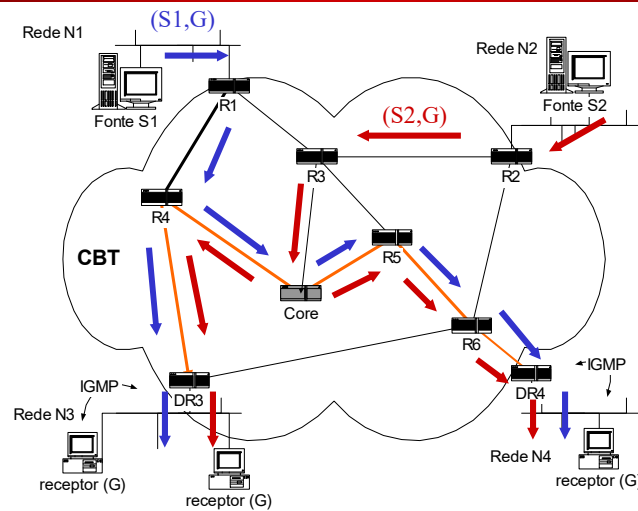
GTA/UFRJ

Construção da Árvore CBT



GTA/UFRJ

Envio de Dados no CBT



GTA/UFRJ

CBT

○ Escalabilidade

- Estado apenas nos roteadores na árvore de distribuição
 - Ao contrário de DVMRP e MOSPF
- Estado por (grupo), em vez de por (fonte, grupo)

○ Desvantagens

- Concentração de tráfego próximo ao core
- Rotas sub-ótimas entre a fonte e o receptor
 - Maiores atrasos

○ Localização do core é crítica

GTA/UFRJ

PIM

- **Protocol Independent Multicast (PIM)**
 - Independente do protocolo de roteamento unicast
- **Dense-Mode (PIM-DM)**
 - Receptores densamente distribuídos
 - Árvores por fonte
 - Inundação-e-poda (semelhante ao DVMRP)
- **Sparse-Mode (PIM-SM)**
 - Receptores esparsamente distribuídos na rede
 - Árvores compartilhadas (como o CBT)
 - Uni-direcionais

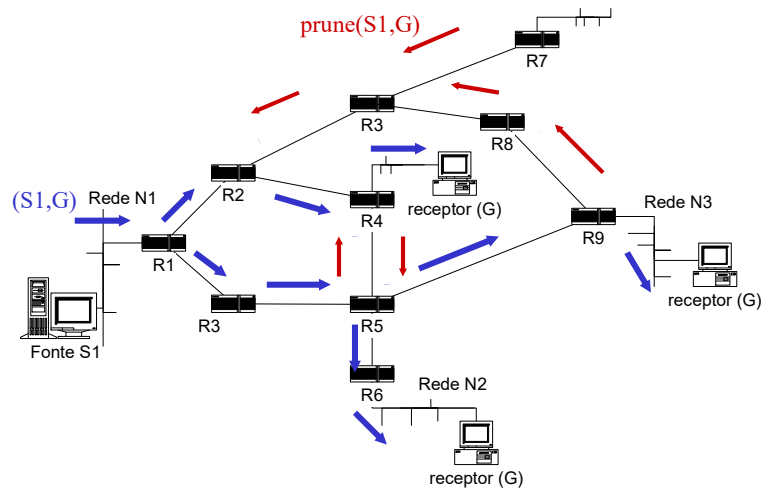
GTA/UFRJ

PIM-DM

- **Reverse Path Multicast**
 - Utiliza o teste RPF
 - Mas não constrói lista de interfaces filhas como o DVMRP
 - Tráfego enviado em todas as interfaces de saída
 - Duplicação de pacotes, todos os enlaces da rede são utilizados, mas
 - independência do roteamento unicast
 - evita base de dados com pais/filhos
 - Após a inundação inicial, mensagens de poda são enviadas
 - Por roteadores que não possuem receptores do grupo
 - Por roteadores que não possuem vizinhos interessados no grupo
 - Por roteadores que receberam tráfego por uma interface incorreta (RPF)

GTA/UFRJ

PIM-DM



GTA/UFRJ

PIM-DM

- **Árvore SPT reversa (RSPT)**
 - União dos caminhos mais curtos dos receptores até a fonte
- **Todos os roteadores da rede armazenam estado (fonte, grupo) para todas as fontes/grupos ativos**
- **Inundação periódica é necessária**
 - Descoberta de novos membros do grupo

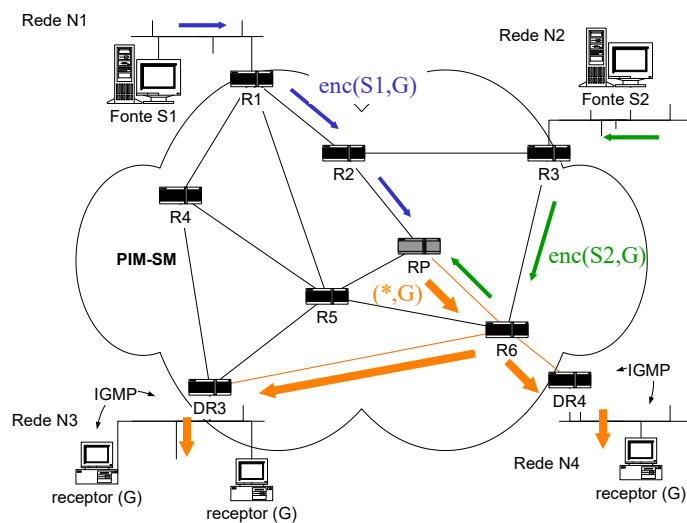
GTA/UFRJ

PIM-SM

- **Árvores de distribuição centradas ($(*,G)$, como o CBT)**
 - Nó central – roteador RP (*rendez-vous point*)
 - Uni-direcional
- **Construção da árvore**
 - Mensagens *join*
- **Mecanismo de mapeamento entre grupos e RPs**
- **Fontes se “registram” com o RP**
 - Dados são enviados ao RP (encapsulados em mensagens **PIM-register**)

GTA/UFRJ

Árvore Compartilhada no PIM-SM



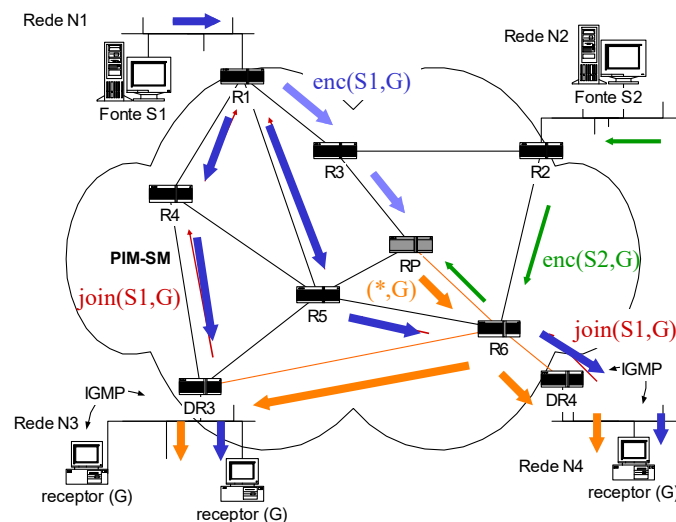
GTA/UFRJ

PIM-SM

- **Árvores por fonte (S,G)**
- **Troca realizada por configuração**
 - Taxa de envio de dados
- **Roteador local envia mensagens `join(S,G)`**
 - Mas não pára o envio de `join(*,G)`
 - Tráfego de outras fontes deve continuar
 - Envia mensagem de poda especial (**RP-bit-prune(S,G)**)
 - Evita a recepção de dados de **S** em duplicata

GTA/UFRJ

Árvore por Fonte no PIM-SM



GTA/UFRJ

PIM-SM

- **RP também pode enviar `join(S,G)`**
- **Possibilidade de árvores por fonte**
 - Diminui a importância da localização do RP
 - Reduz o atraso fonte-receptores

GTA/UFRJ

Problemas do Modelo de Serviço IP Multicast

- **Como limitar o alcance (ou escopo) do tráfego multicast**
 - Até onde vai o tráfego enviado por uma fonte?
 - (receptores **não** são conhecidos)
- **Como evitar a colisão de endereços**
 - Duas aplicações escolhem o mesmo endereço multicast

GTA/UFRJ

Alcance do Tráfego Multicast

- **Definição de Escopos: até onde o tráfego alcança**
- **Técnicas para limitar o alcance**
 - Por endereço
 - Utilizando o campo TTL
 - Administrativos

GTA/UFRJ

Escopo por Endereço

- **Faixa de endereços dinâmicos da Classe D**
 - 224.0.1.0 a 239.255.255.255
- **Sub-divisão**
 - 224.0.1.0 a 238.255.255.255
 - aplicações com escopo global
 - 239.0.0.0 a 239.255.255.255
 - aplicações com escopo limitado
 - 239.253.0.0/16 – local ao site
 - 239.192.0.0/14 – local à organização

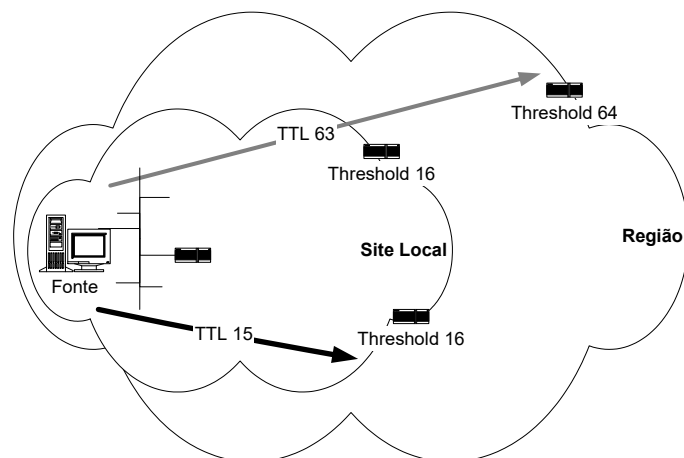
GTA/UFRJ

Escopo usando o TTL

- **TTL (*Time-to-live*)**
 - Campo decrementado de 1 a cada roteador atravessado
 - Pacote descartado quando TTL=0
- **Escopo usando o TTL**
 - Escolhe-se um valor de TTL inicial para os pacotes multicast
- **Limita-se a distância em número de saltos**
 - Pouca correlação entre numero de saltos e uma região
- **Limiar TTL (*TTL threshold*)**
 - Configurado nos roteadores de borda
 - Pacotes com TTL menor que o limiar de TTL são descartados

GTA/UFRJ

Escopo usando o TTL

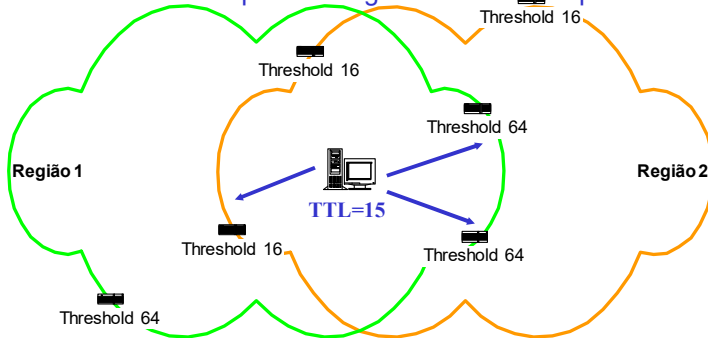


GTA/UFRJ

Escopos Administrativos

○ Roteadores não encaminham certas faixas de endereços

- Maior flexibilidade que por TTL
- Por TTL não se pode configurar zonas ~~sobrepostas~~

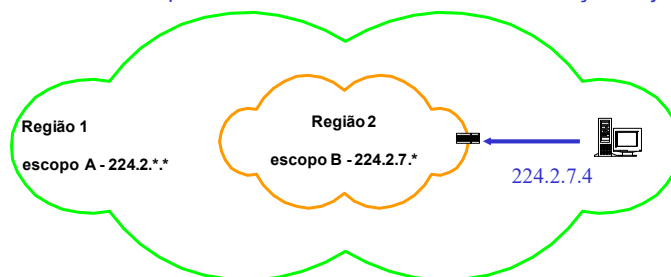


GTA/UFRJ

Escopos Administrativos

○ Desvantagens

- Alcance definido por **todas** as zonas às quais a fonte pertence
 - Como descobrir que zonas se aplicam?
- Zonas sobrepostas devem utilizar faixas de endereços disjuntas



- Erros de configuração
 - Zonas maiores ou menores que o necessário
 - Com o TTL, pode-se escolher um valor pouco maior que o necessário e garantir o funcionamento da aplicação

GTA/UFRJ

Roteamento Inter-domínio

- **Nem todos os roteadores são multicast**
- **Diferentes protocolos nos diferentes domínios**
- **Problemas com o PIM-SM**
 - Mecanismo escalável de mapeamento entre RPs e grupos
 - Inter-dependência entre provedores de serviço introduzida pelos RPs

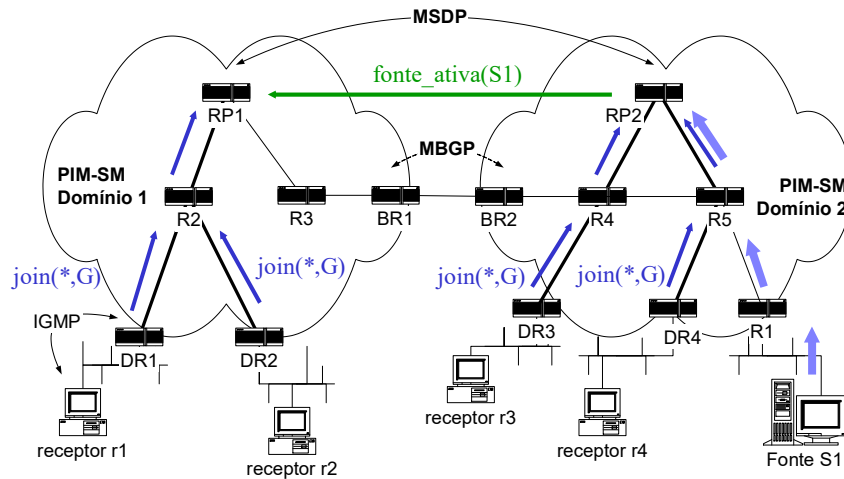
GTA/UFRJ

Arquitetura MBGP/MSDP

- **Solução de curto-prazo**
 - Interconexão de domínios PIM-SM
- **MBGP – *Multiprotocol Extensions for BGP-4***
 - Permite múltiplas tabelas de roteamento
 - Pode-se utilizar uma tabela unicast e uma tabela multicast
 - M-RIB (*Multicast – Route Information Base*)
- **MSDP – *Multicast Source Discovery Protocol***
 - Anúncio das fontes ativas, entre todos os RPs

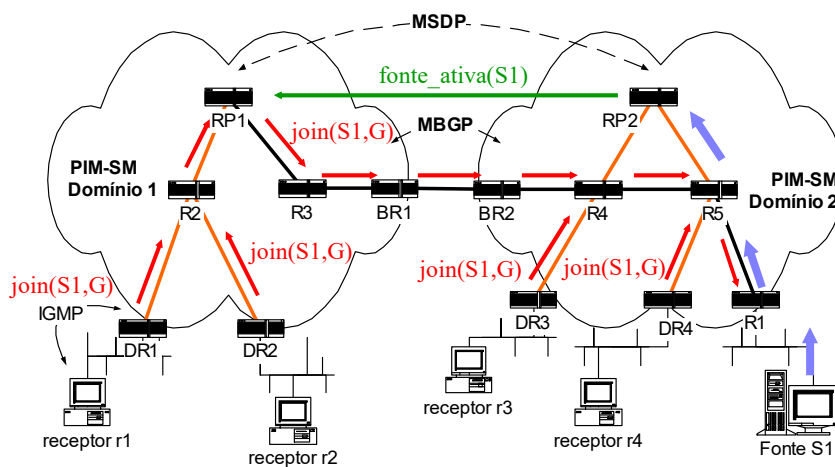
GTA/UFRJ

Árvores Intra-domínio no MBGP/MSDP



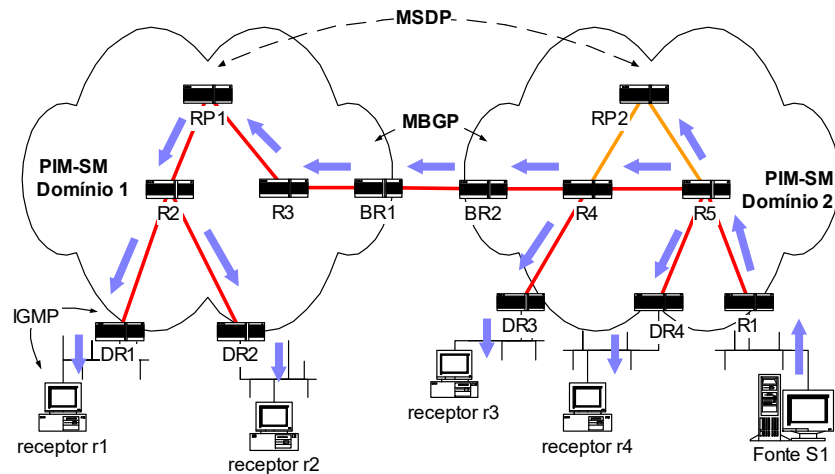
GTA/UFRJ

Árvore Inter-domínio no MBGP/MSDP



GTA/UFRJ

Envio de Dados no MBGP/MSDP



GTA/UFRJ

MBGP/MSDP

- Inter-dependência entre domínios evitada
- **Todos os domínios são notificados de todas as fontes ativas**
 - Problema de escalabilidade
- **Tráfego é encapsulado nas mensagens de “fonte-ativa”**
 - Evita perda dos primeiros dados
 - E de fontes em rajadas
 - Problema: dados são enviados a *todos* os RPs

GTA/UFRJ

Novas Propostas

○ Modelo de Serviço IP Multicast

- Endereço IP class-D = grupo de estações
 - qualquer estação pode se inscrever no grupo
 - e qualquer estação pode enviar dados para o grupo
- alocação de endereços multicast é problemática
- protocolos: IGMP + protocolos de roteamento

○ IP Multicast não foi implantado na Internet

- Redes de *backbone* superdimensionadas

○ Tentativas de simplificação da arquitetura

- Simple Multicast
- EXPRESS, PIM-SSM
- REUNITE, HBH

GTA/UFRJ

Protocolos Multicast

○ IGMP

- Gerenciamento de grupo (estações – roteadores designados)

○ Protocolos de roteamento

- Modo denso
 - DVMRP, PIM-DM
 - Inundação-e-poda, árvores por fonte
- Modo esparsos
 - PIM-SM
 - Join explícito, árvores compartilhadas, árvores por fonte

○ MBGP (*Multi-protocol BGP*)

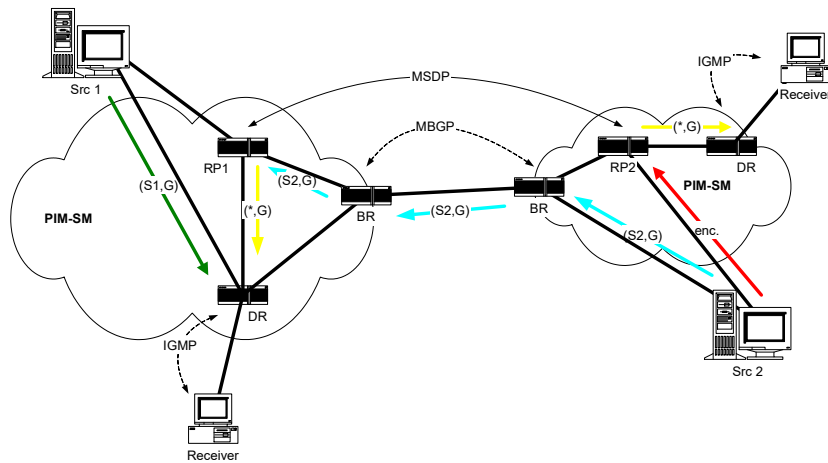
- Anúncio de rotas unicast e multicast

○ MSDP (*Multicast Source Discovery Protocol*)

- Anúncio de fontes ativas entre todos os RPs

GTA/UFRJ

Arquitetura Atual



GTA/UFRJ

Inconvenientes da Arquitetura IP Multicast Original

- **Modelo de serviço aberto**
- **Alocação de endereços**
- **PIM-SM**
 - é possível comutar da árvore compartilhada para árvore por fonte
 - nos roteadores Cisco
 - limiar de tráfego configurado para 1 pacote
 - RP, MSDP
 - servem apenas para a descoberta de fontes
 - Árvore por fonte é preferível em muitas aplicações
 - Mesmo para fontes conhecidas
 - Construção da árvore compartilhada no início da transmissão

GTA/UFRJ

EXPRESS

- **EXPlicitely REquested Single Source multicast**
- **Canal multicast**
 - 1 fonte para N receptores
 - ECMP protocol
 - controle do canal
 - coleta de informações sobre o canal
- **Canal**
 - (S,G) - S = endereço IP da fonte, G = endereço multicast classe D

GTA/UFRJ

Source Specific Multicast

- **SSM (Source-Specific Multicast)**
 - conversação 1 x N
 - *Subscribe channel* <S,G>
 - Fornece base para o controle de acesso
 - Apenas S pode enviar para (S,G), outras fontes são bloqueadas
 - Alocação de endereços multicast (G)
 - Problema local à fonte
 - Roteadores RP e o protocolo MSDP não são necessários

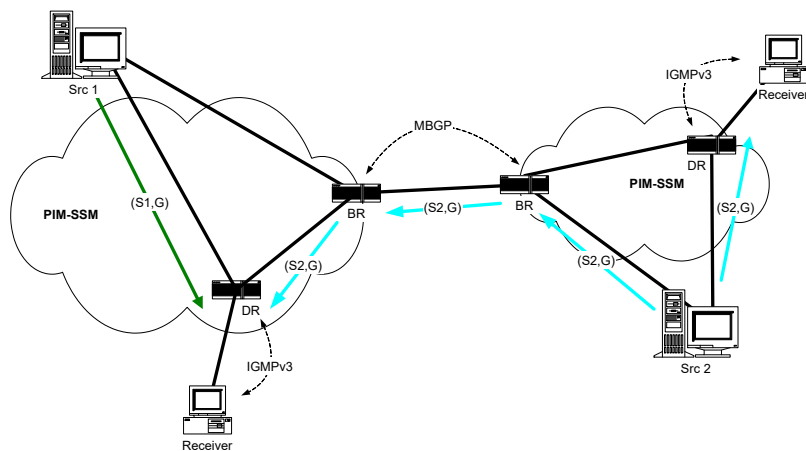
GTA/UFRJ

Componentes do Serviço SSM

- **Faixa de endereços exclusiva - 232/8 (IANA)**
- **Roteamento: PIM-SSM**
 - Versão modificada do PIM-SM
 - Pode implementar ambos os serviços (SM & SSM)
- **IGMPv3 (MLDv2 no IPv6)**
 - Suporta a filtragem de fontes
 - (INCLUDE, EXCLUDE)

GTA/UFRJ

Arquitetura SSM



GTA/UFRJ

Funcionamento do PIM-SSM

○ Regras do PIM-SSM

- somente *join*(S,G) é permitido na faixa 232/8
- *join*(*,G) e *join*(S,G) permitidos na faixa restante
- roteadores de borda (DR no PIM)
 - implementam *join*(S,G) imediato
- roteadores de núcleo
 - devem evitar as árvores compartilhadas em 232/8

GTA/UFRJ

Modificações no IGMPv3

○ Using IGMPv3 and MLDv2 For Source-Specific Multicast

<RFC4604.txt>

○ Estações

- Módulo IGMP não precisa ser modificado
- Aplicações devem conhecer a faixa de endereços SSM, e utilizar apenas uma API específica à fonte nesta faixa

○ Roteadores

- Na faixa de endereços SSM, apenas modo INCLUDE
 - IGMP reports (queries) são processados (produzidos) de acordo

GTA/UFRJ

Observações Finais

- **Arquitetura IP Multicast**
 - Continua complexa
 - Ainda possui problemas de escalabilidade
 - Estado armazenado nos roteadores
- **Faltam ferramentas de gerenciamento**
- **Modelo de tarifação complexo**

GTA/UFRJ