How did IP Multicast get so complicated?

Mark Handley
ACIRI
mjh@aciri.org

Overview

- **IP Multicast Service Model**
  - Multicast Addresses
- **DVMRP (1988-1993)**
  - Broadcast and Prune
- **PIM-DM (~1993)**
  - DVMRP for "real" routers
- **CBT and PIM-SM (~1993)**
  - A different way of looking at the problem
- **Core Discovery**
  - We merely moved the problem
  - HPIM or change the service model?
  - PIM-SM v2
- **MSDP**
  - A necessary short-term kludge

Overview (cont)

- **Multicast Address Allocation**
  - SAP/SDP (1993-)
  - Scaling limits
- **Adding Hierarchy to Address Allocation**
  - AAP and MASC
- **MADCAP: client API to address allocation**

- **BGMP: Core Discovery Revisited**

- **But it's so complicated!!!**
  - Express
  - Simple Multicast

IP Multicast

- **Dynamically constructs efficient distribution trees from senders to receivers.**

- **Service Model:**
  - Receivers announce their interest
  - Senders just send
  - Routers conspire to deliver data from senders to receivers

Class D addresses

- **Traditionally, IP Addresses were divided into classes:**
  - Class A for networks with millions of hosts
  - Class B for networks with thousands of hosts
  - Class C for networks with tens to hundreds of hosts

- **Class D is multicast group addresses**
  - A multicast sender just sends to a class D multicast address
  - Multicast receivers express an interest in a class D multicast address
  - If they choose the same address, the network delivers traffic from the senders to the receivers.
  - Class D addresses are in the range 224.0.0.0 to 239.255.255.255.

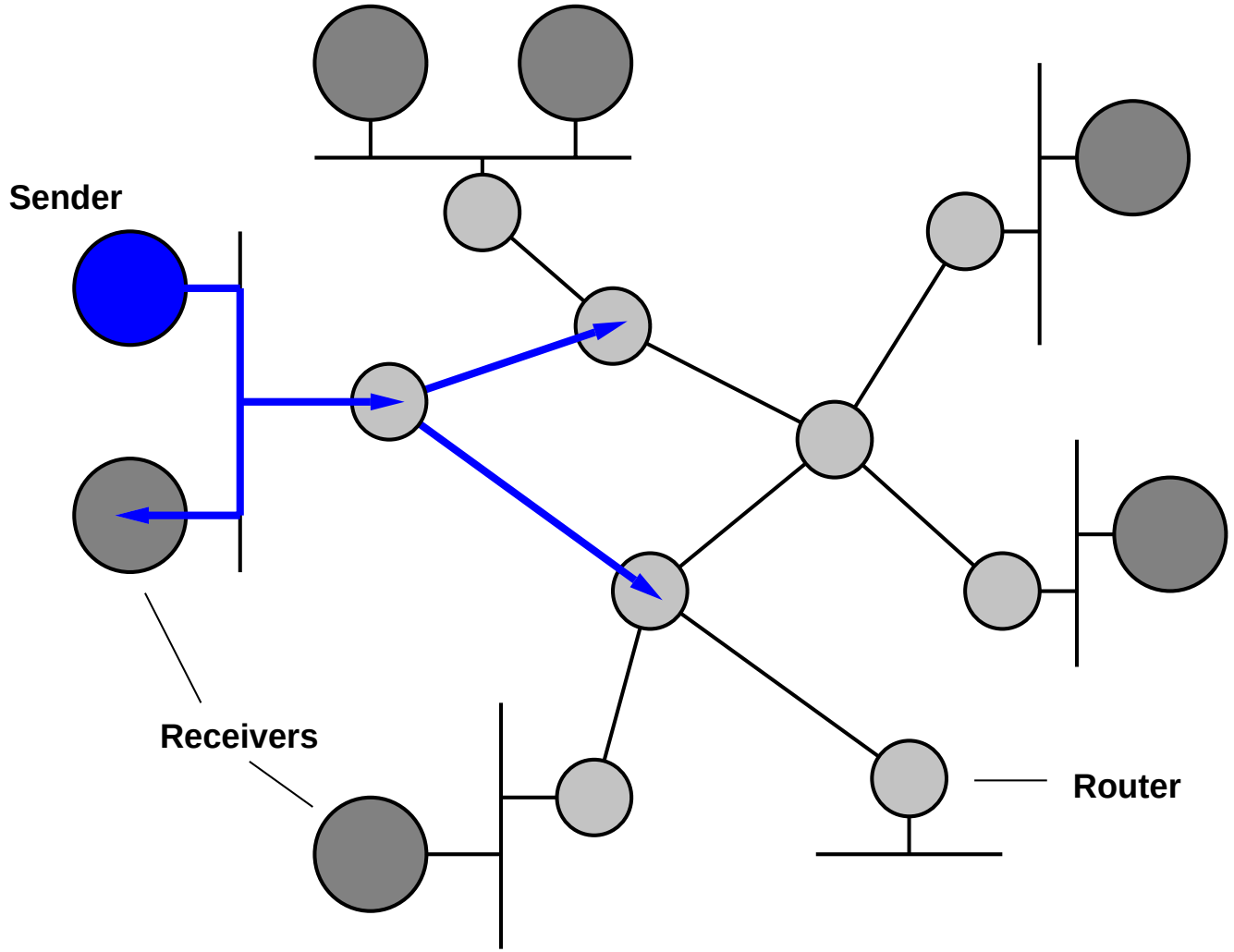
Why is this so cool?

- **Efficient one-to-many distribution.**

- **Multicast address is a logical address.**
  - It binds together senders and receivers, which need have no prior knowledge of each other or the network topology.
  - In the extreme, this can lead to a whole new way of viewing the design of applications (peer-to-peer rather than client/server).
  - Resource discovery made easy.

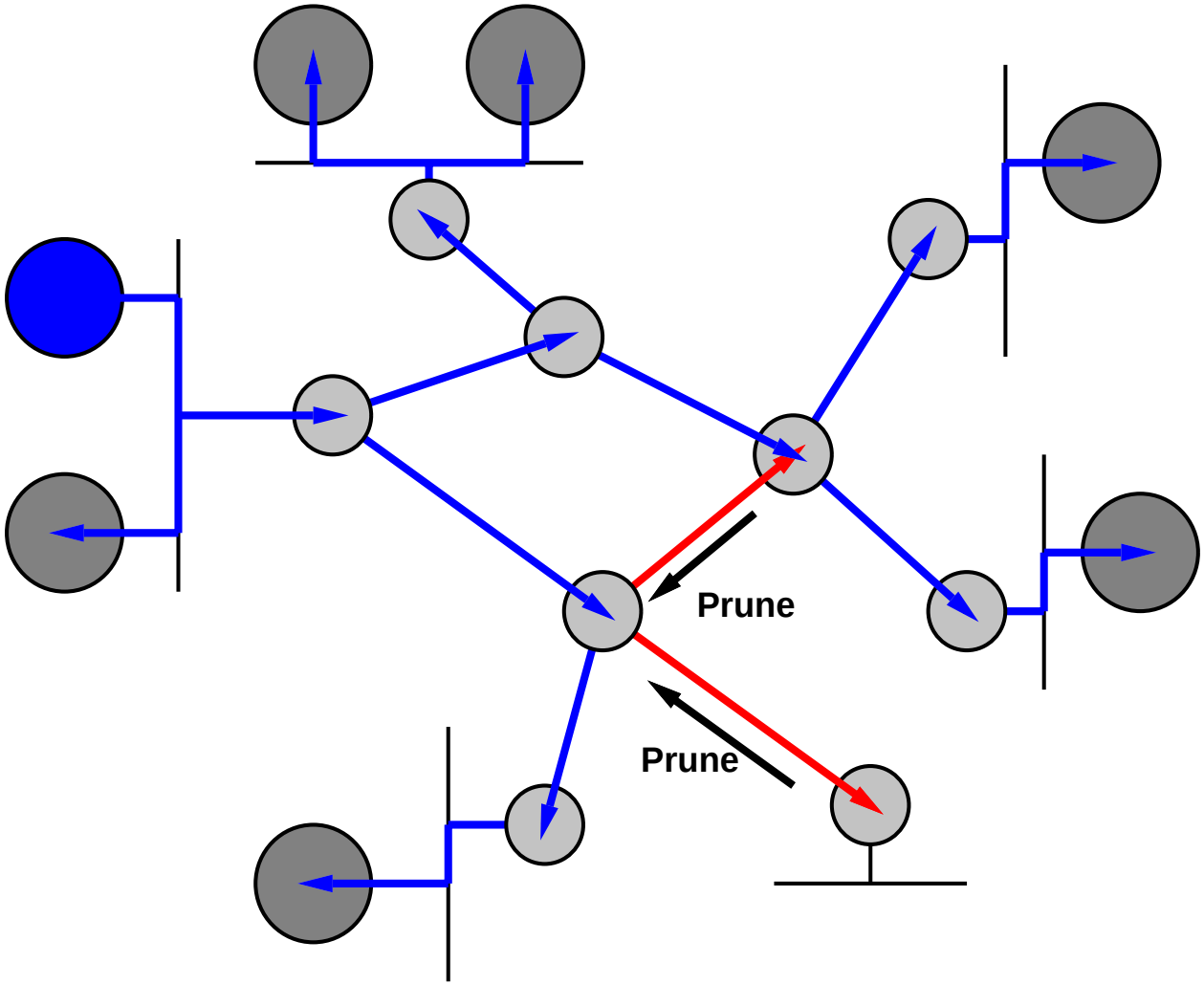
DVMRP (Steve Deering, ~1988)

- **Basic Model is "Broadcast and Prune":**



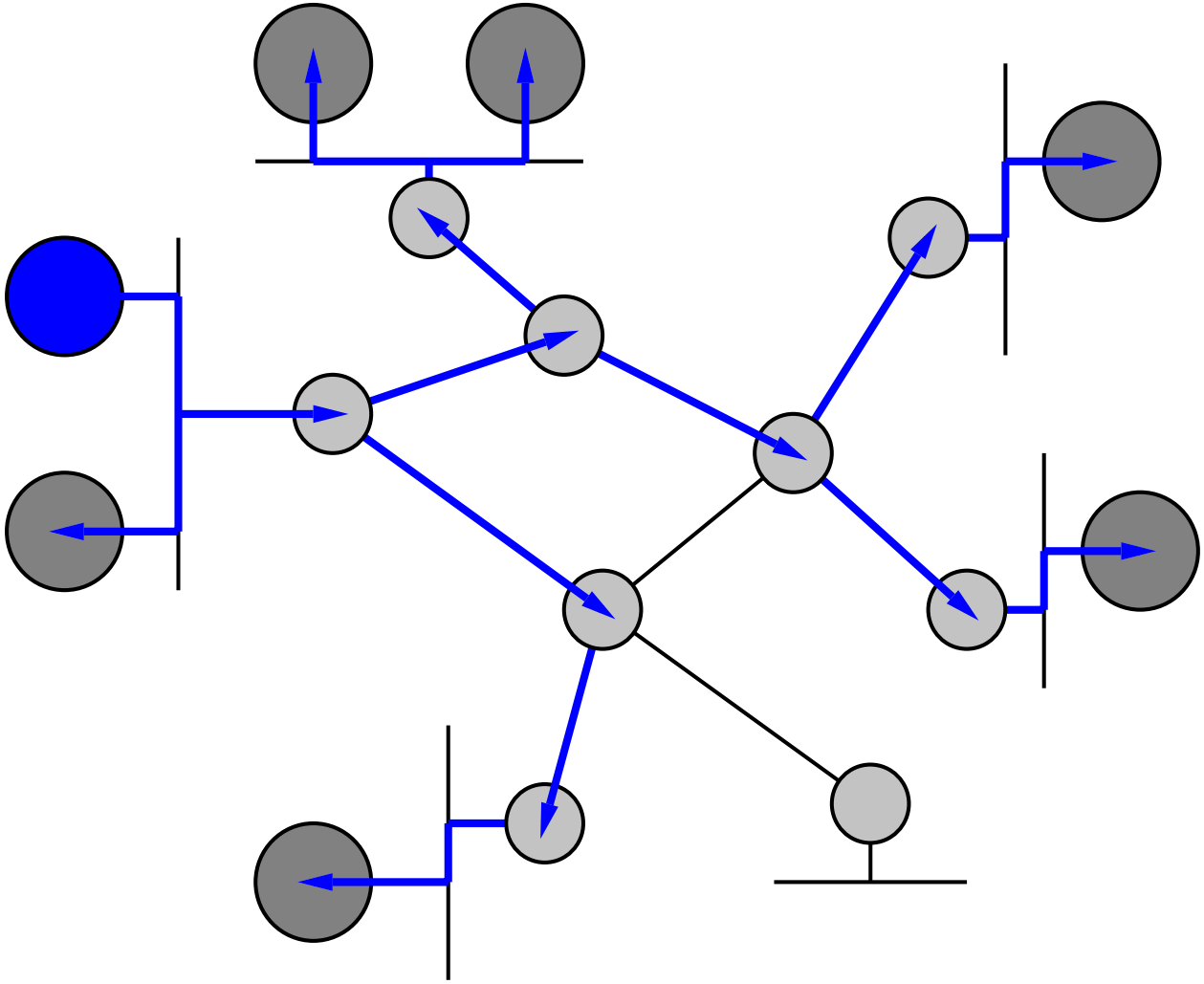
Broadcast and Prune

- **Branches not on the shortest path tree to a receiver are pruned off:**



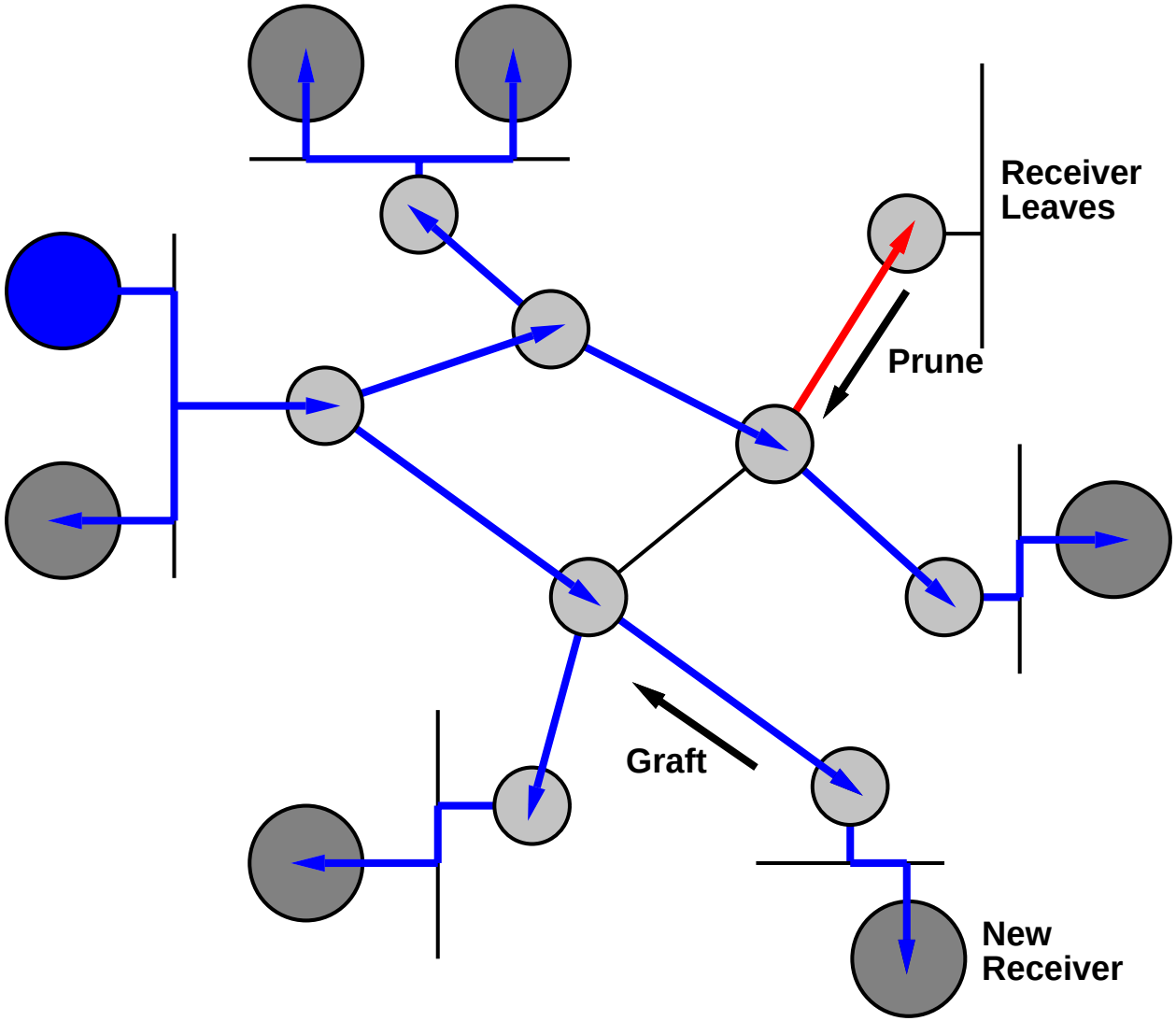
Shortest Path Tree

- **Resulting in a shortest path distribution tree rooted at the sender:**



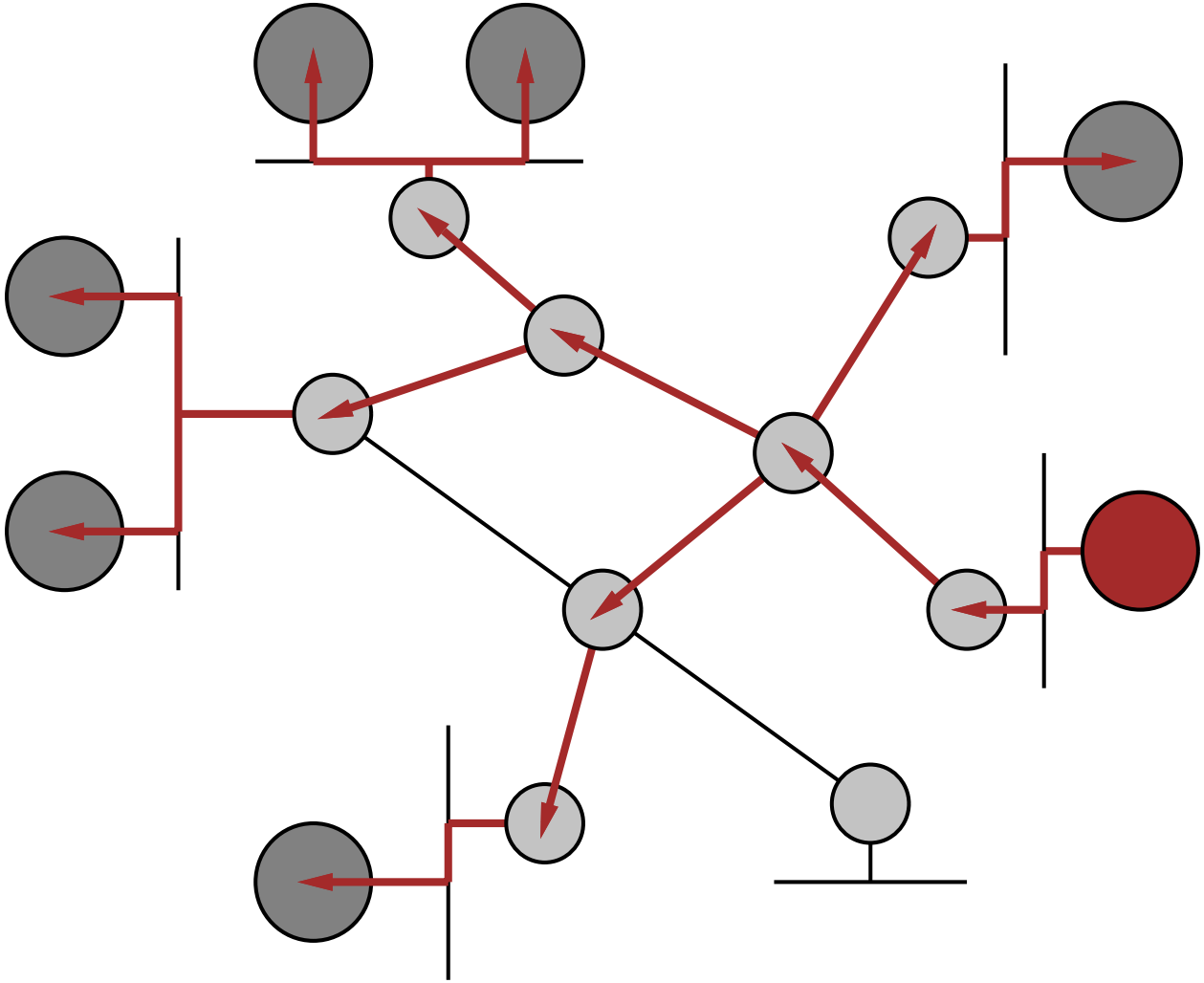
Local Tree Maintenance

- **Changes in membership are handled efficiently and locally by grafting and pruning:**



A Shortest Path Tree Per-Sender

- Distribution trees are per-{sender,group} and are triggered by data packets:



Dense-Mode PIM

- **PIM = Protocol Independent Multicast**
  - DVMRP builds it's own routing table.
  - PIM attempts to use the underlying unicast routing table.

- **Very similar to DVMRP**
  - Broadcast and Prune.

- **Also DVMRP pre-computes certain RPF information.**
  - This prevents packet duplication.
  - PIM didn't want to do this, so detects duplicates and then fixes them.

Surely there's a better way?

- **Dense mode protocols create per-(source,group) state in routers unrelated to where the traffic goes.**

- **Why not rendezvous at a meeting point instead of broadcasting the data?**
  - Basic idea from Paul Francis/Van Jacobson

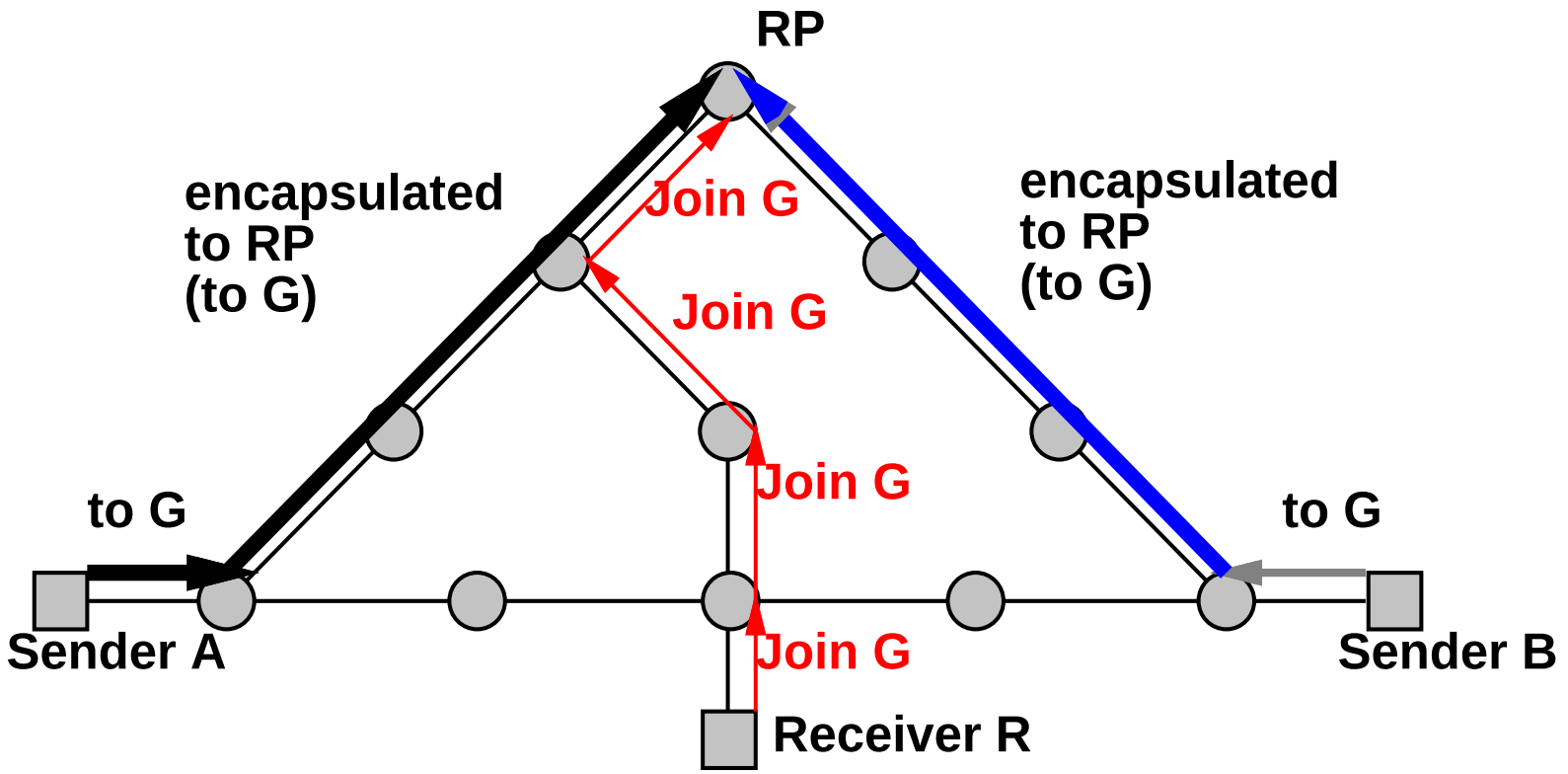
Result:

- **Core Based Trees (CBT)**
  - Francis, Ballardie and Crowcroft
- **Sparse-mode PIM (SM-PIM)**
  - Estrin, Farinacci, Jacobson, et al.

Sparse-mode PIM

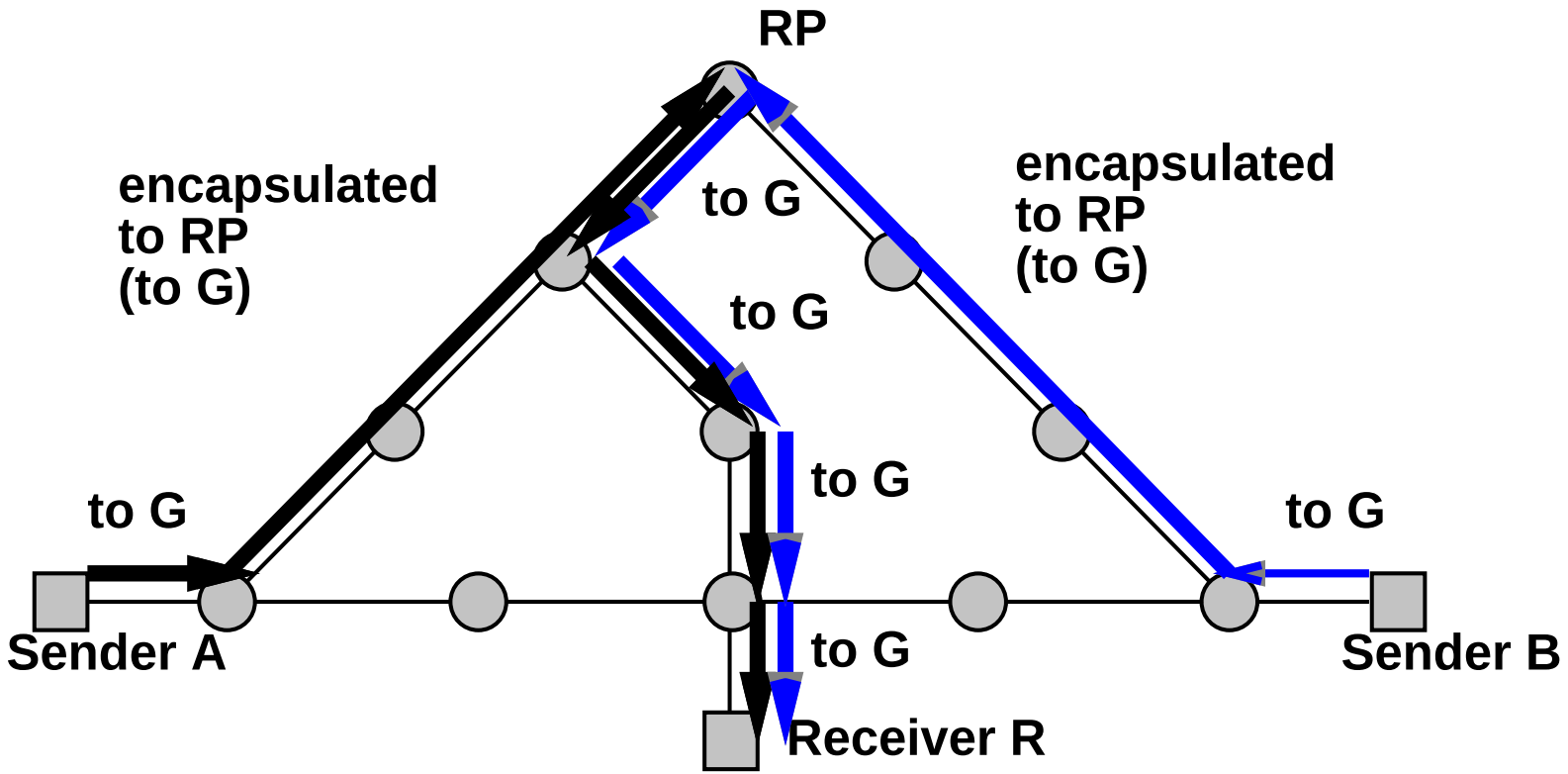
- **Instead of flooding and pruning to directly build a shortest path tree, Sparse-mode PIM initially builds a shared tree.**
- **The shared tree is built by sending join messages towards a Rendezvous Point (RP).**
- **Once data is flowing, the shared tree can be converted to a shortest path tree if required.**

PIM-SM: Building the Shared Tree



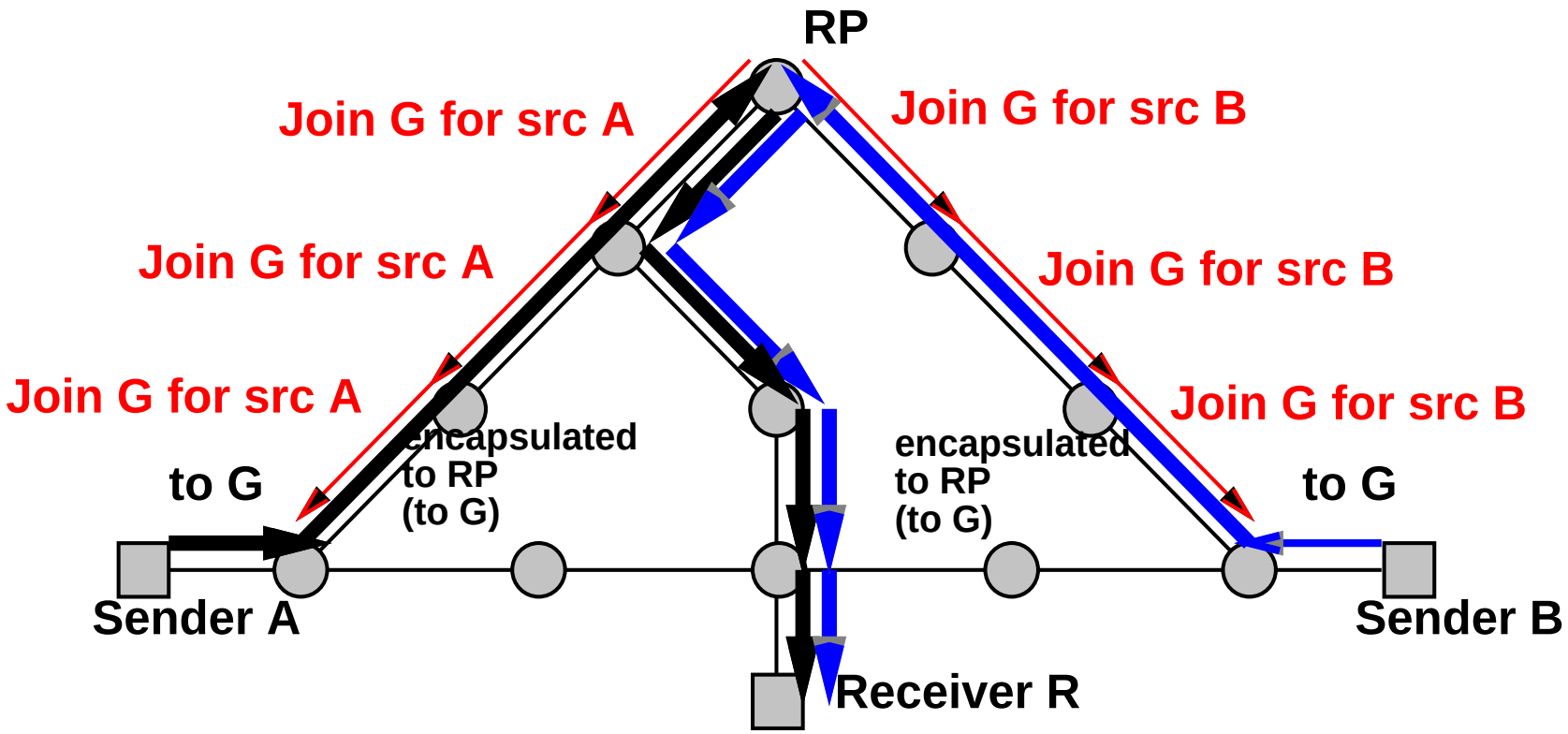
- **Sources A and B send data.**
  - Their local routers encapsulate it to the RP.
- **Receiver R joins the group.**
  - Its local router sends a join message towards the RP.

PIM-SM: Building the Shared Tree



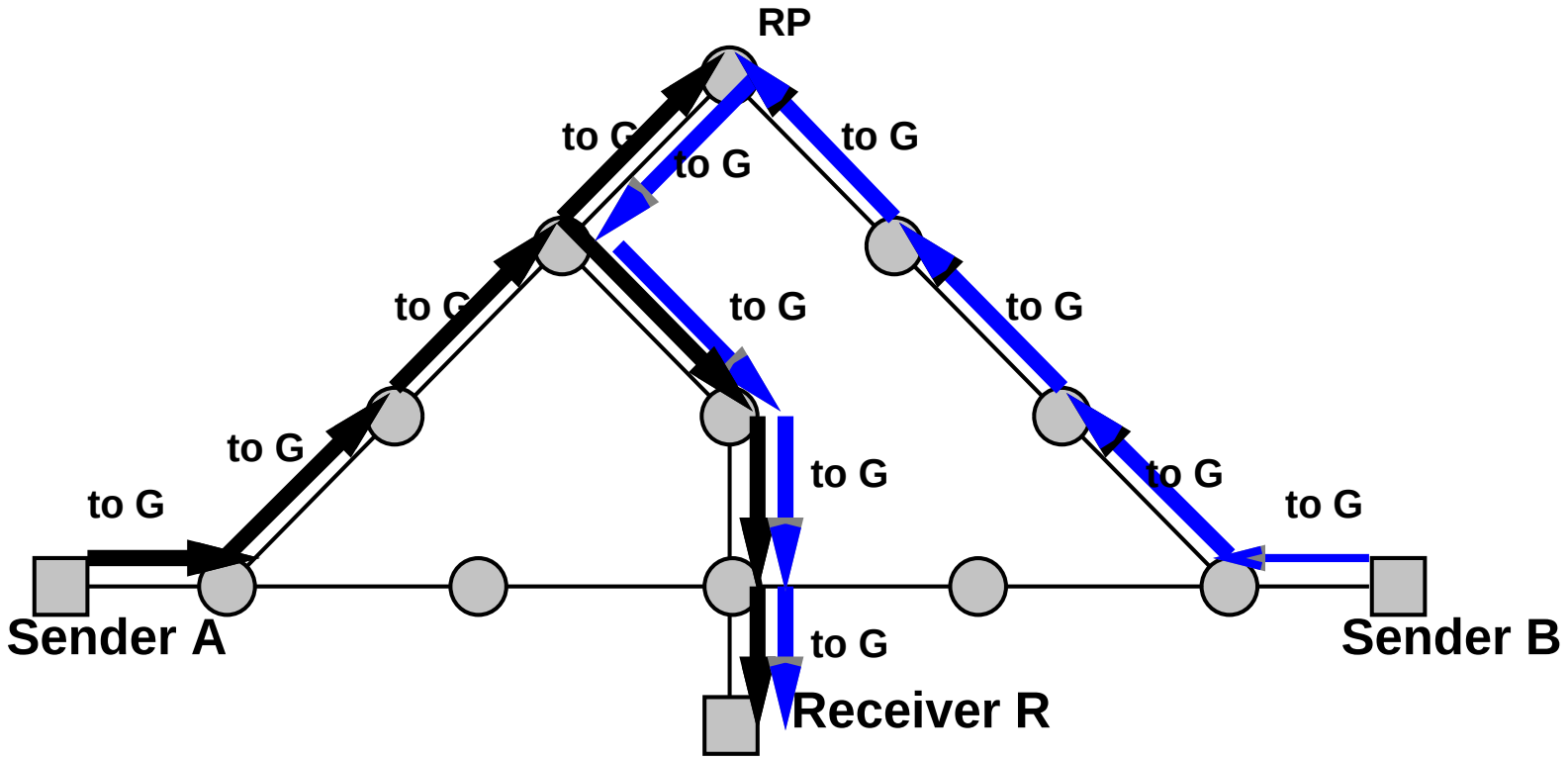
- Data starts to reach to R from A and B.

PIM-SM: Building the Shared Tree



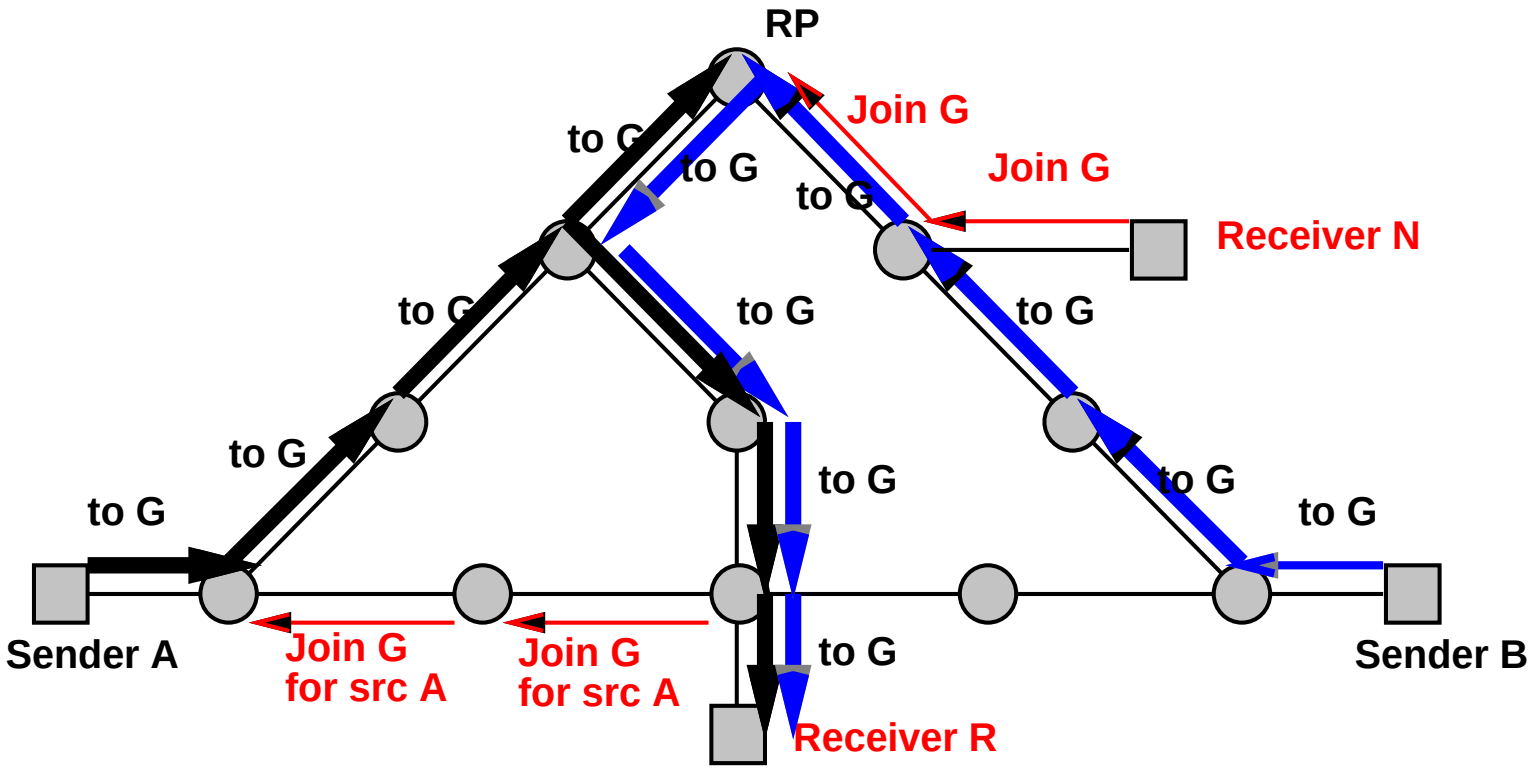
- **The RP also sends join messages back to A and B.**
  - This will end the encapsulation.

PIM-SM: Building the Shared Tree



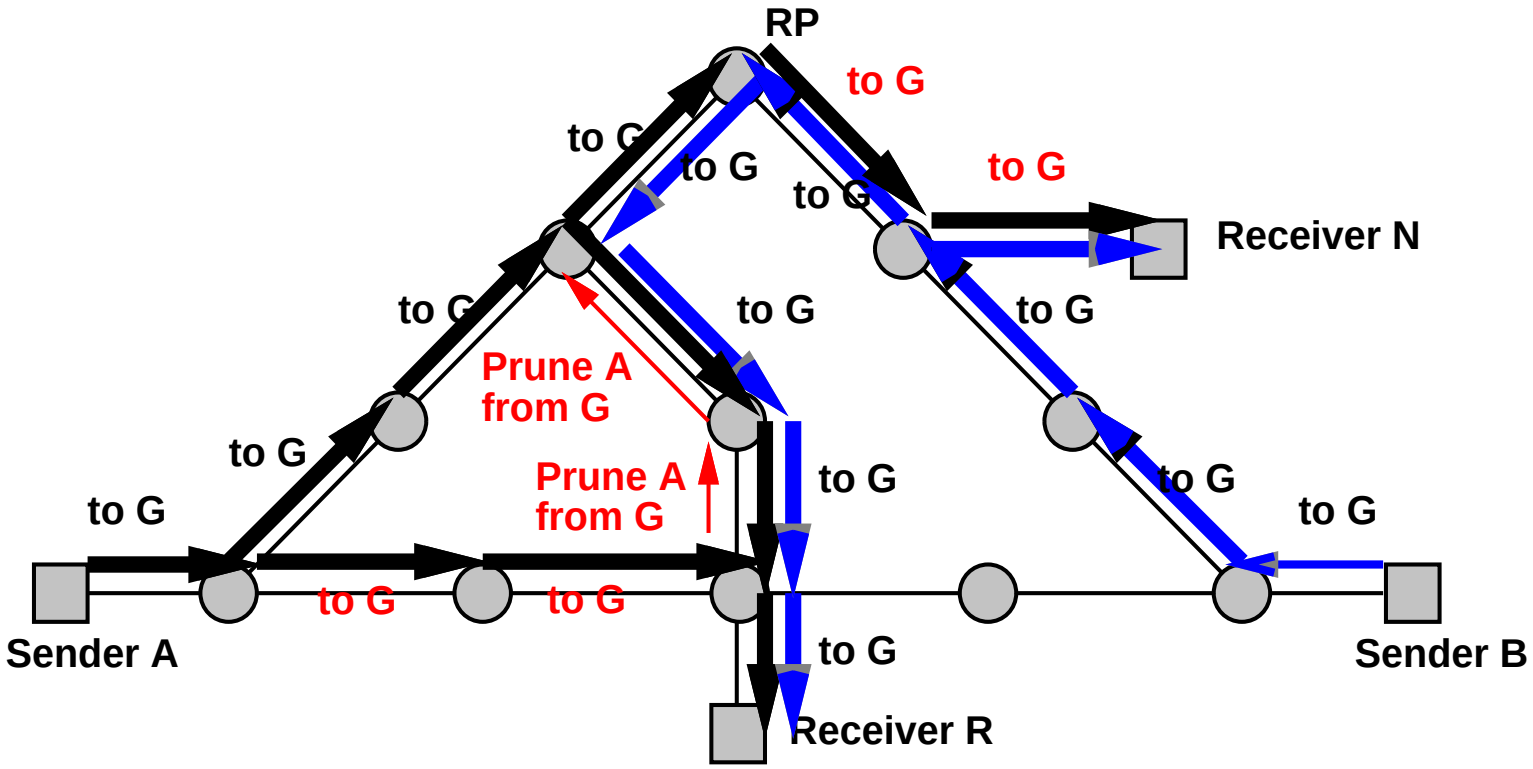
- The data now flows natively on the shared tree.

PIM-SM: Shortest Path Tree



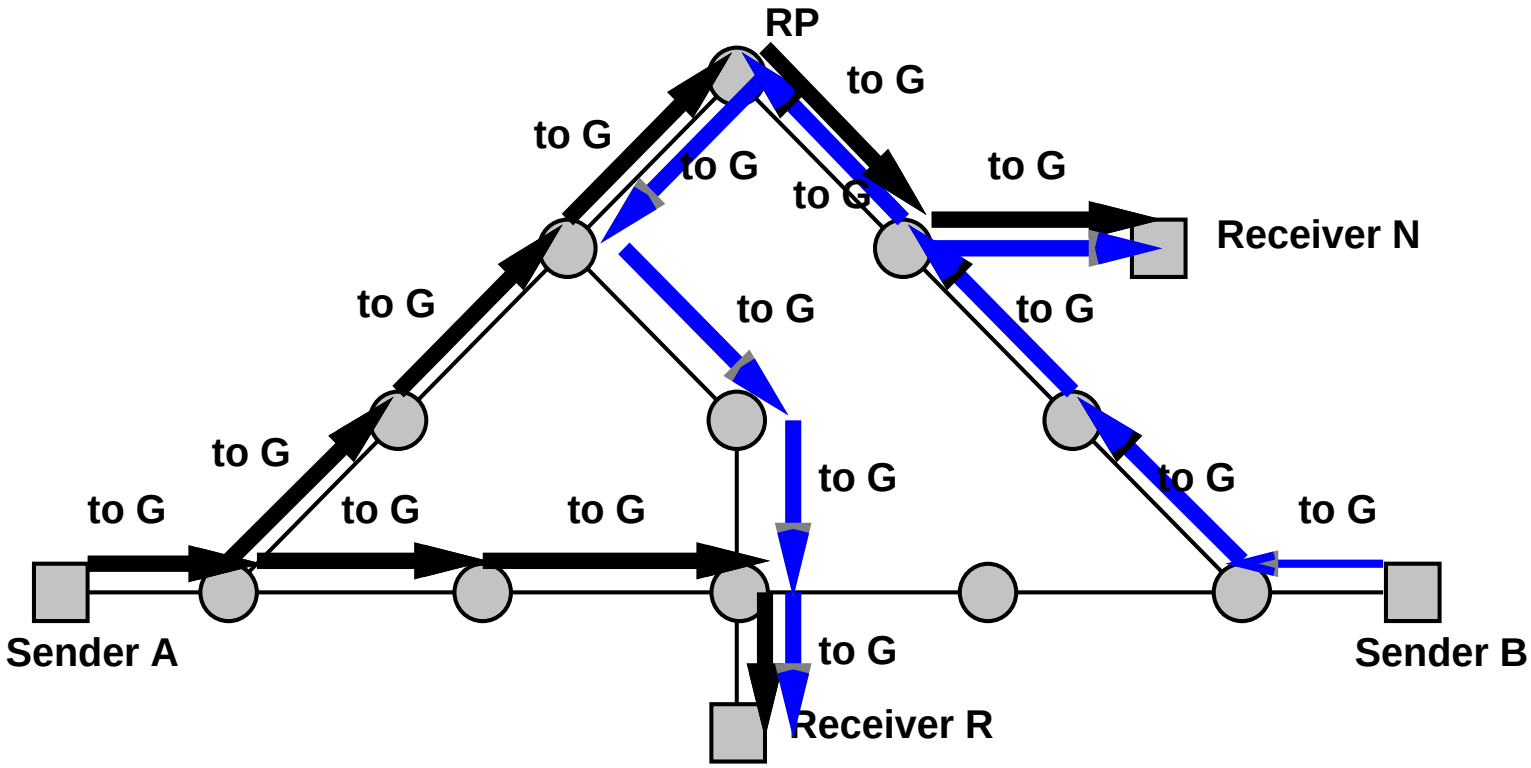
- **A is sending high rate traffic.**
  - R's local router decides to switch to SP tree.
  - It sends an {S,G} Join towards A.
- **A new receiver N also joins.**

PIM-SM: Shortest Path Tree



- **Traffic from A now flows two ways to R's local router**
  - This triggers an {S,G} prune to be sent towards RP.
- **N also receives traffic on the shared tree.**

PIM-SM: Shortest Path Tree



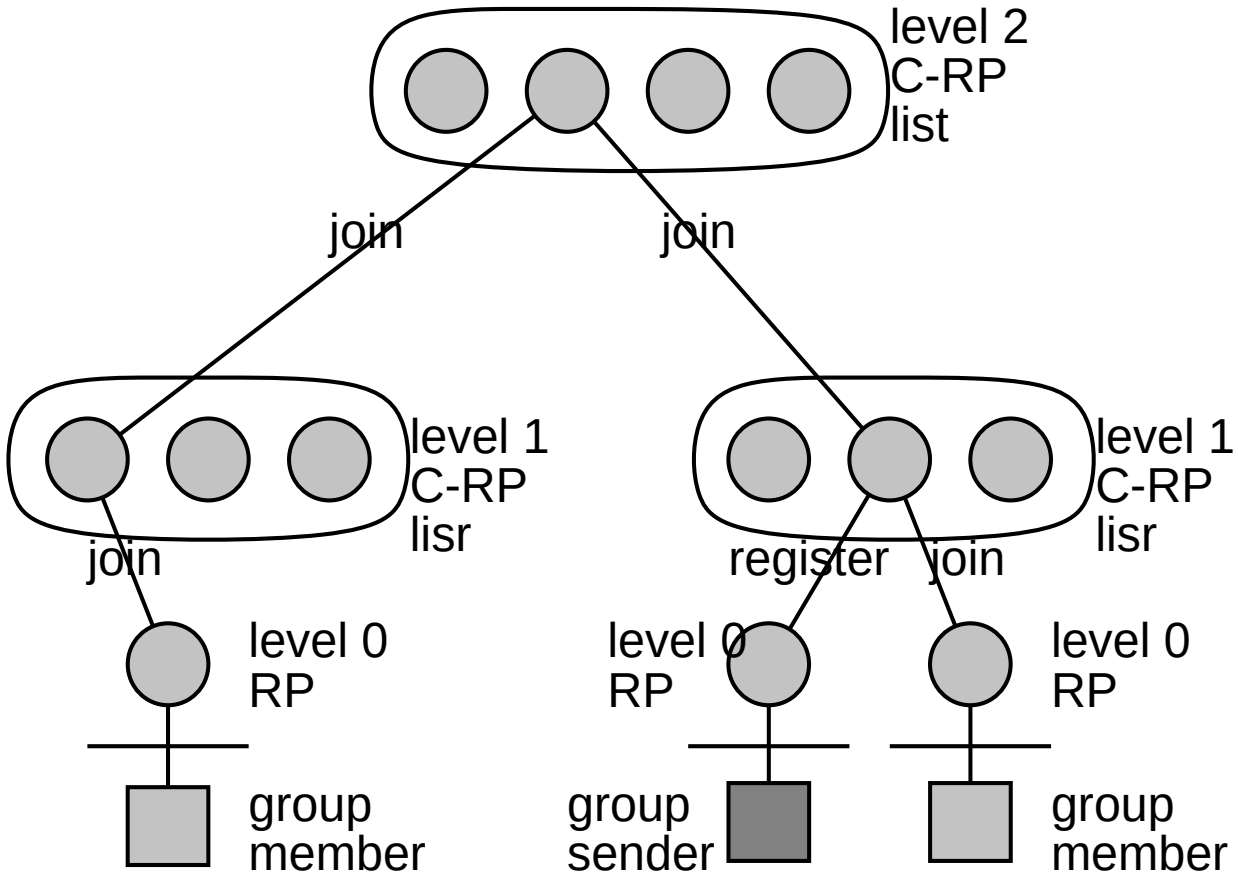
- R receives traffic from A on the SP tree
- R receives traffic from B on the shared tree
- N receives traffic from A and B on the shared tree

PIM-SM: Finding the RP

- **The problem with PIM and CBT is how the local routers discover the address of the RP.**
  - PIM-SM v1 did this through router configuration.
- **The idea of getting the hosts to tell their local routers the RP address was extensively discussed, but eventually dismissed:**
  - It changes the service model (and hence all hosts).
  - It introduces a single point of failure.
  - It requires the host to know which routing protocol is being used.
  - The logical address becomes a physical address without the level of indirection provided by a routing protocol.

Hierarchical PIM

- **Handley/Crowcroft/Wakeman, 1995**
- **Arrange the world into a hierarchy of domains.**
  - Find the RP by hashing into a pre-distributed list of candidates at each level.
  - Each RP joins to an RP at the level above.



Sparse-mode PIM (version 2)

- **Took the idea of hashing into a set of candidate RPs from HPIM, but didn't adopt the hierarchy.**
- **Number of candidate RPs scales linearly with size of domain, so SM-PIMv2 does cannot scale globally.**
- **Additional problem:**
  - The world is divided up into many ISPs.
  - ISPs do not trust each other.
  - An ISP doesn't want to depend on an RP in another ISP for multicast service between its own customers.

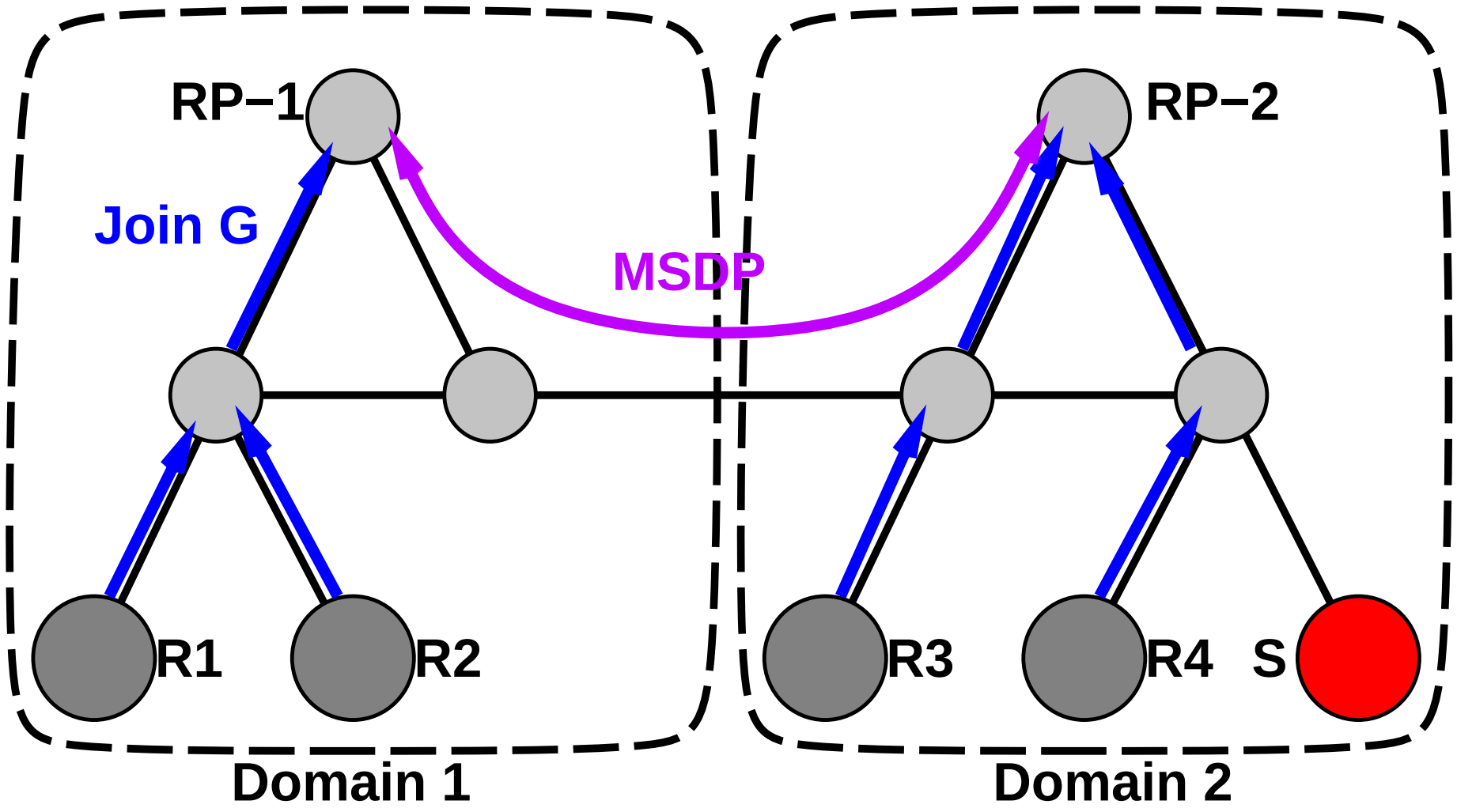
Multicast Source Distribution Protocol (MSDP)

- **Even if SM-PIM scaled, ever ISP wants the RP to be in their domain.**

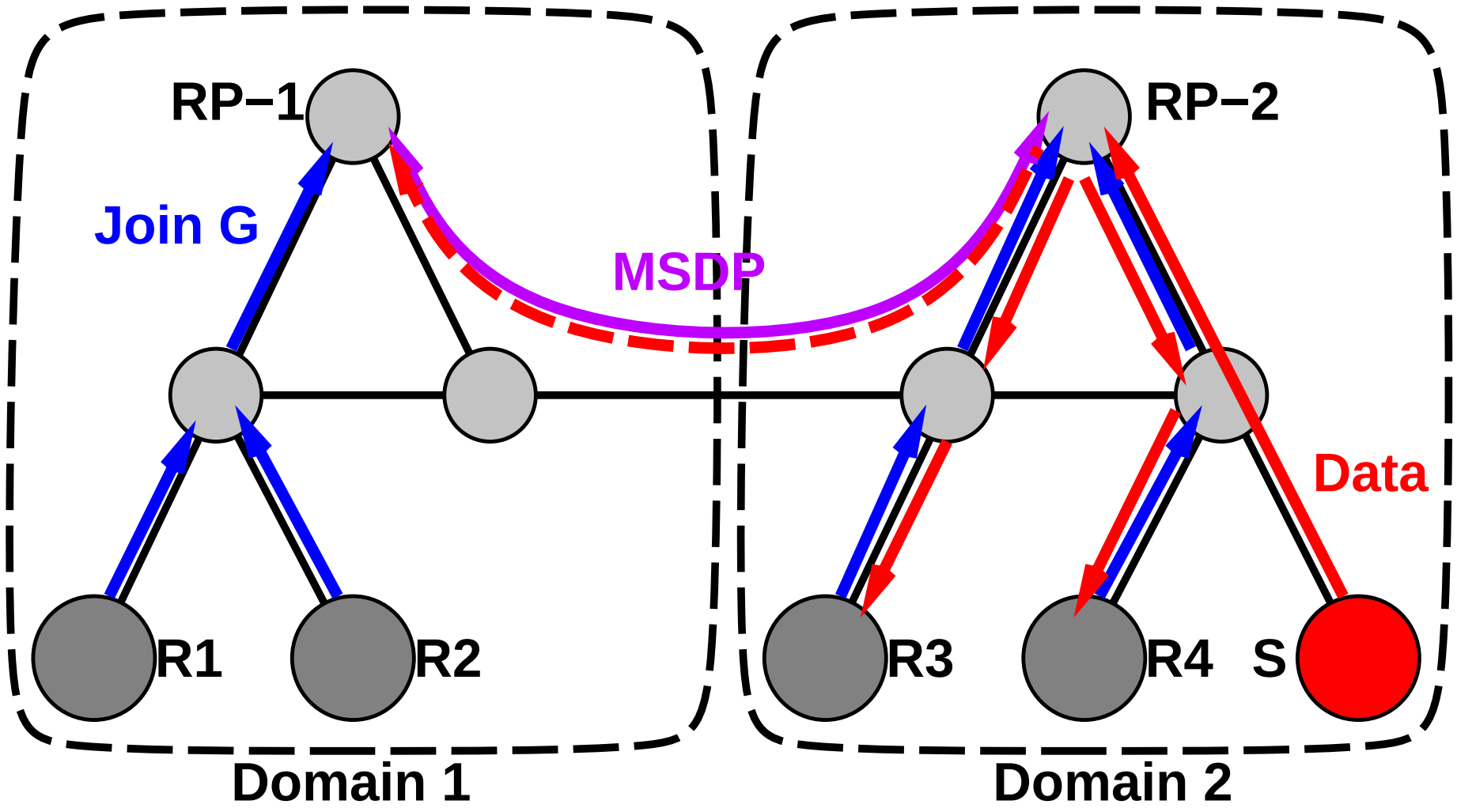
- **Solution:**
  - Put an RP in every domain.
  - Connect them together with some interdomain glue.

- **MSDP is such a glue.**
  - But it's a glue designed only to satisfy short-term scaling requirements.

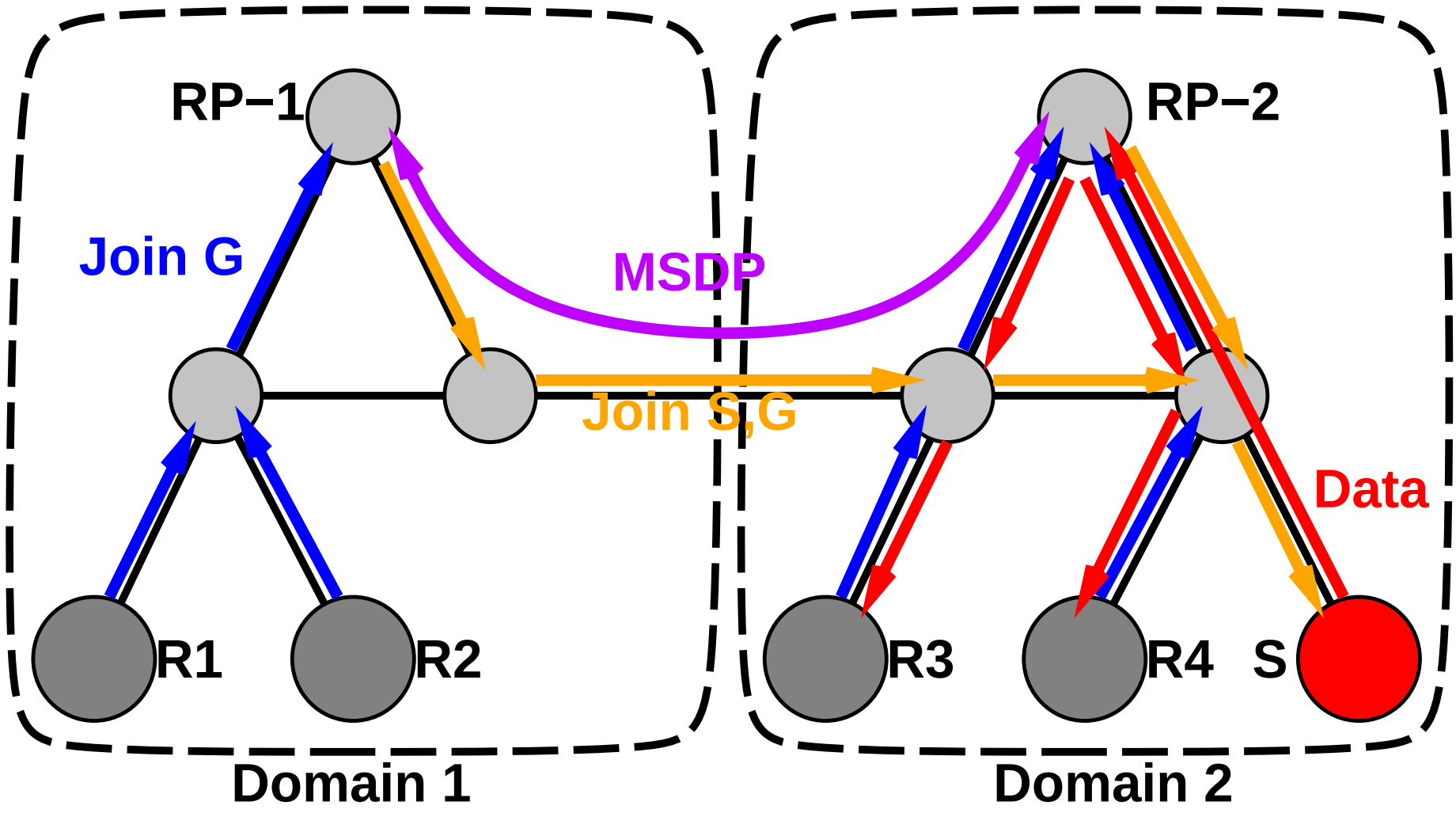
Multicast Source Distribution Protocol (MSDP)



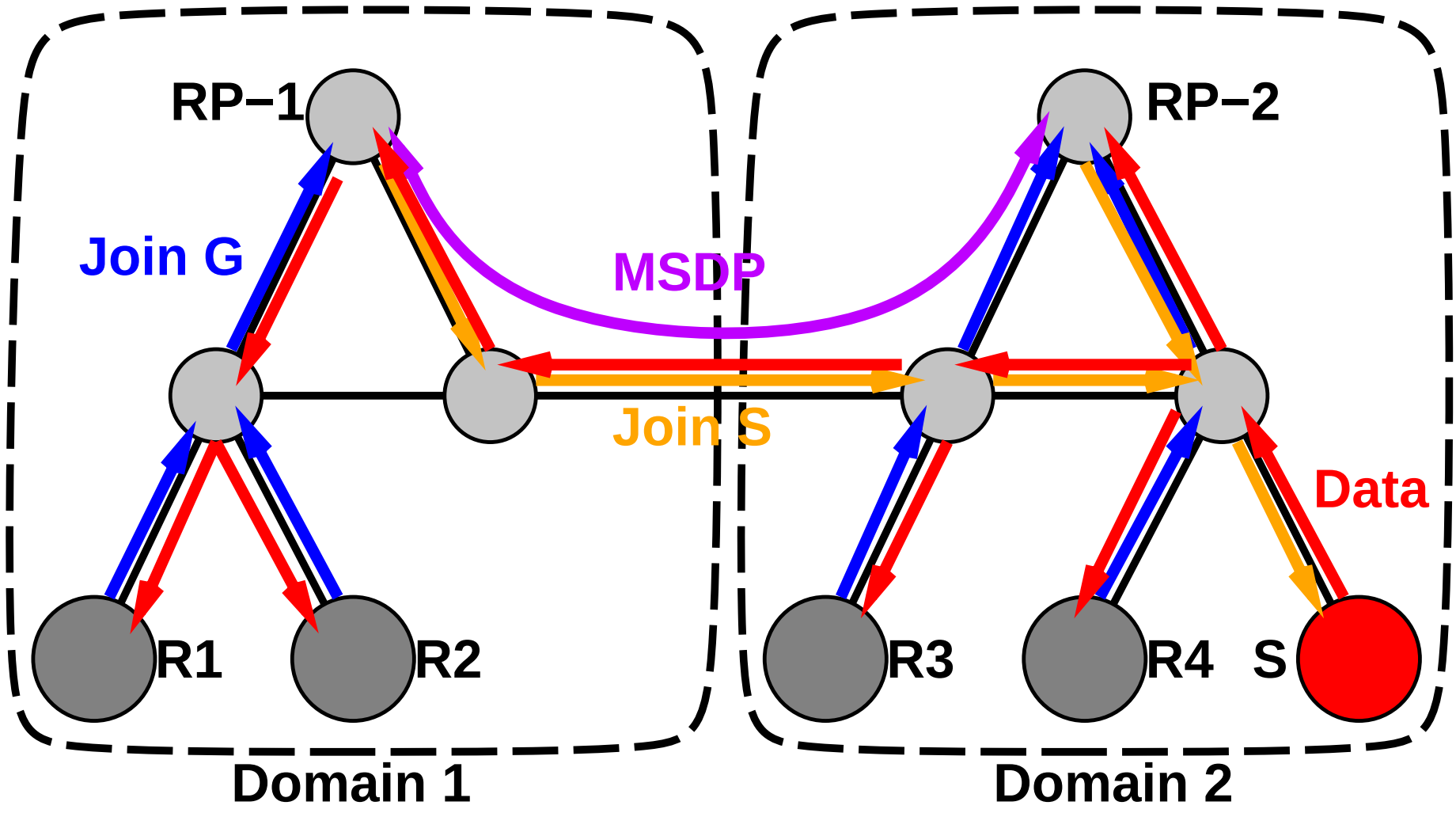
Multicast Source Distribution Protocol (MSDP)



Multicast Source Distribution Protocol (MSDP)



Multicast Source Distribution Protocol (MSDP)



Multicast Address Allocation

- **Architecture Overview**
- **MADCAP**
- **AAP**
- **MASC (and its relation to BGMP)**

Multicast Address Allocation

- **Since 1993, multicast address allocation has been performed by session directories.**
  - sd - McCanne/Jacobson, 1993
  - sdr - Handley, 1994

- **Basic model is to send periodic multicast announcements of sessions.**
  - Inform potential users that the session exists and how to join it.
  - Informs everyone else you're using an address, so they randomly choose addresses not in use.

Informed Random Address Allocation

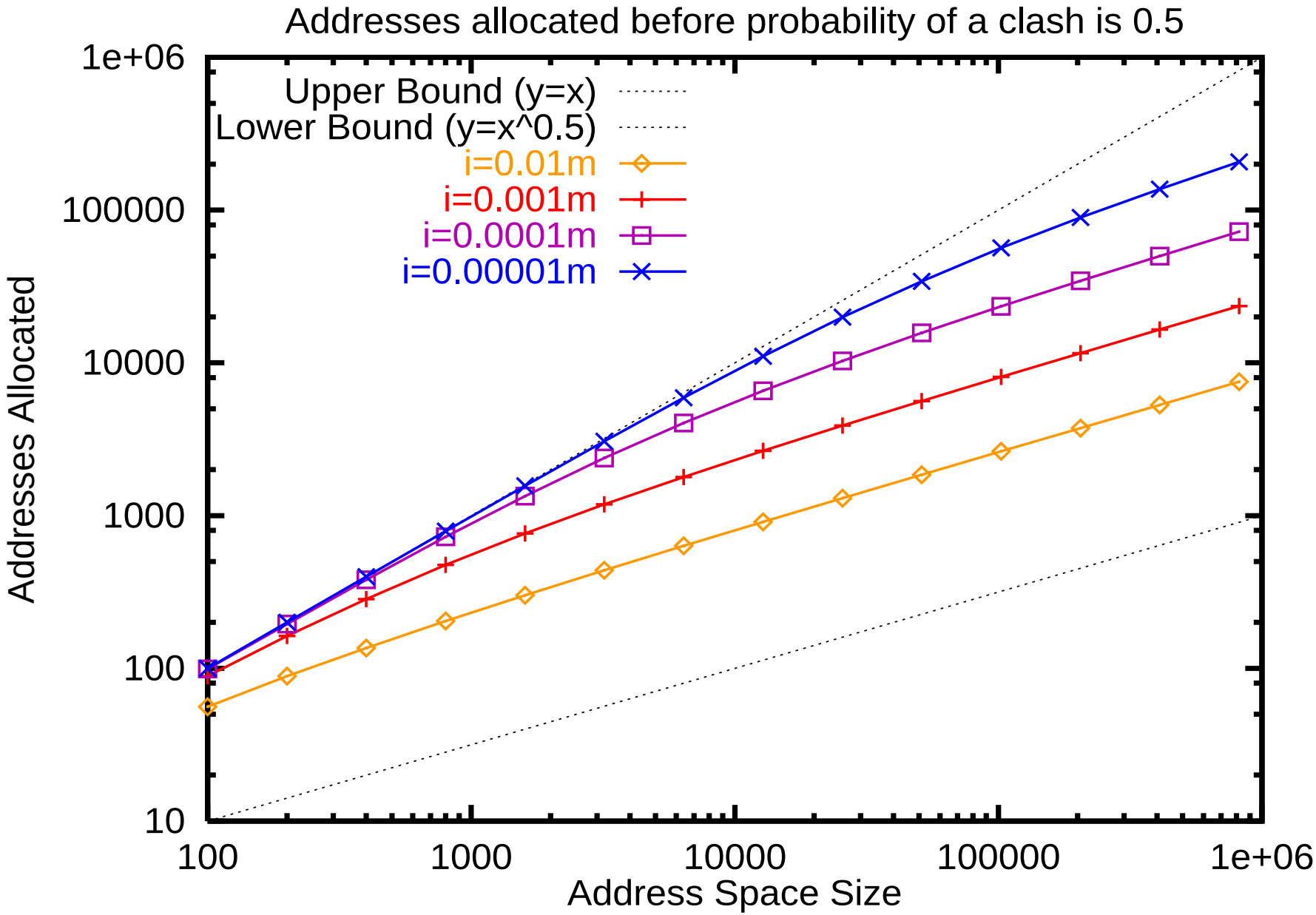
- **With Random Address Allocation the probability of a clash becomes significant with ~sqrt(n) addresses allocated out of a space of n.**

- **Informed Random Address Allocation avoids clashing with sessions already announced.**
  - How does it scale?

Packet Loss and Propagation Delay

- **Assume:**
  - mean session length: 2 hours
  - mean advance announcement: 2 hours
  - mean end-to-end delay: 200ms
  - mean packet loss rate: 2%
  - each announcement resent every 10 mins
- **Mean end-to-end delay ~(0.98*0.2)+(0.02*600) = 12 secs**
- **Therefore approximately 0.1% (12 secs/4 hours) of sessions are not visible at any time.**

Packet Loss and Propagation Delay



Why doesn't it scale?

- **As the number of addresses allocated increases, the number that have just been allocated but not yet seen because of packet loss and propagation delay becomes signficant.**

- **But the model is really nice:**
  - Decentralized
  - Robust
  - No configuration needed
  - Always available

- **How can we remedy the scaling problems and keep the good properties?**

Hierarchical Multicast Address Allocation

- **On short timescales, use a mechanism similar to session announcements within limited domains.**
  - Lower propagation delay.
  - Lower packet loss.
  - Relatively few addresses need allocating.

- **On long timescales, use a similar mechanism to allocate ranges of addresses to domains.**
  - Can wait a long time to see if a collision occurs, and back off if one does.
  - Removes the effect of packet loss.
  - Relatively few ranges need all allocating.

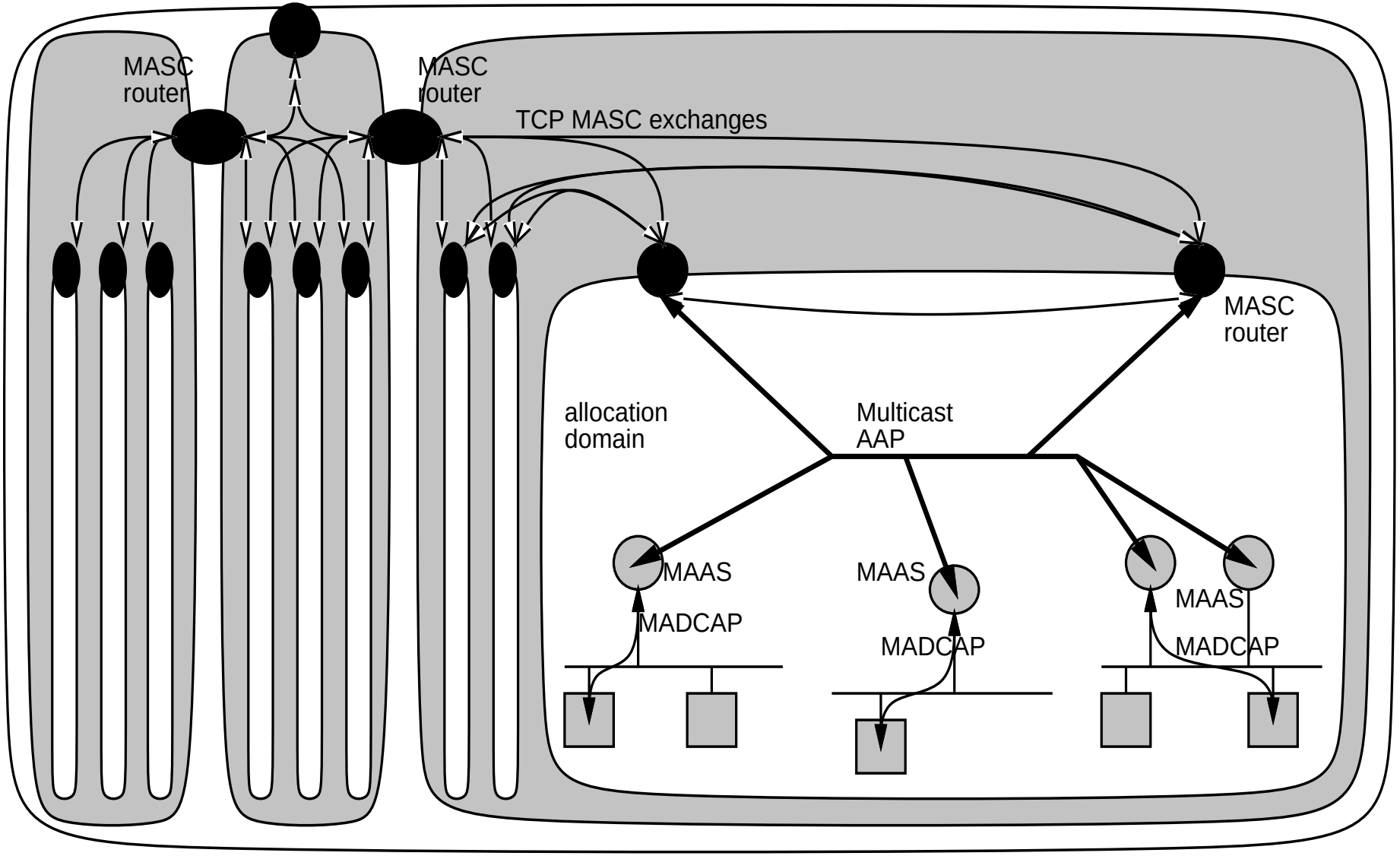
Multicast Address Allocation Protocols

- **Inter-domain: MASC**
  - Allocates ranges of addresses to domains in a dynamic hierarchical manner.
  - Uses TCP instead of multicast so we can use this to bootstrap inter-domain multicast routing.
  - Protocol is pretty simple.
  - Manual configuration.

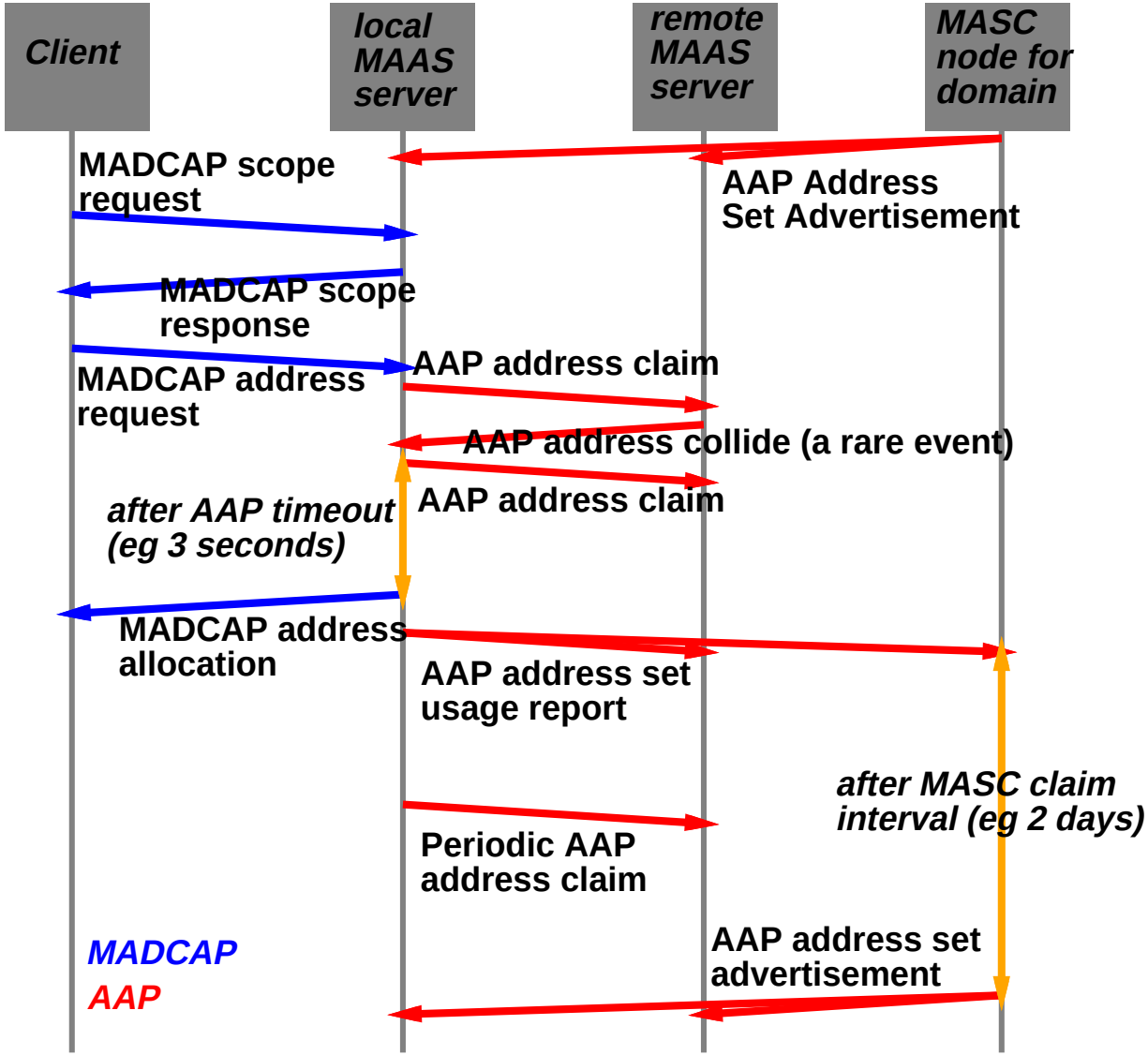
- **Intra-domain: AAP**
  - Allocates individual addresses from ranges (typically provided by MASC).
  - Protocol is very simple (announce/listen)
  - No configuration whatsoever needed.

- **Client-server: MADCAP**
  - Simple request/response protocol to get an address from a server so that not all hosts need to run AAP.

Multicast Address Allocation Overview



Multicast Address Allocation Example



Address Allocation and Interdomain Multicast

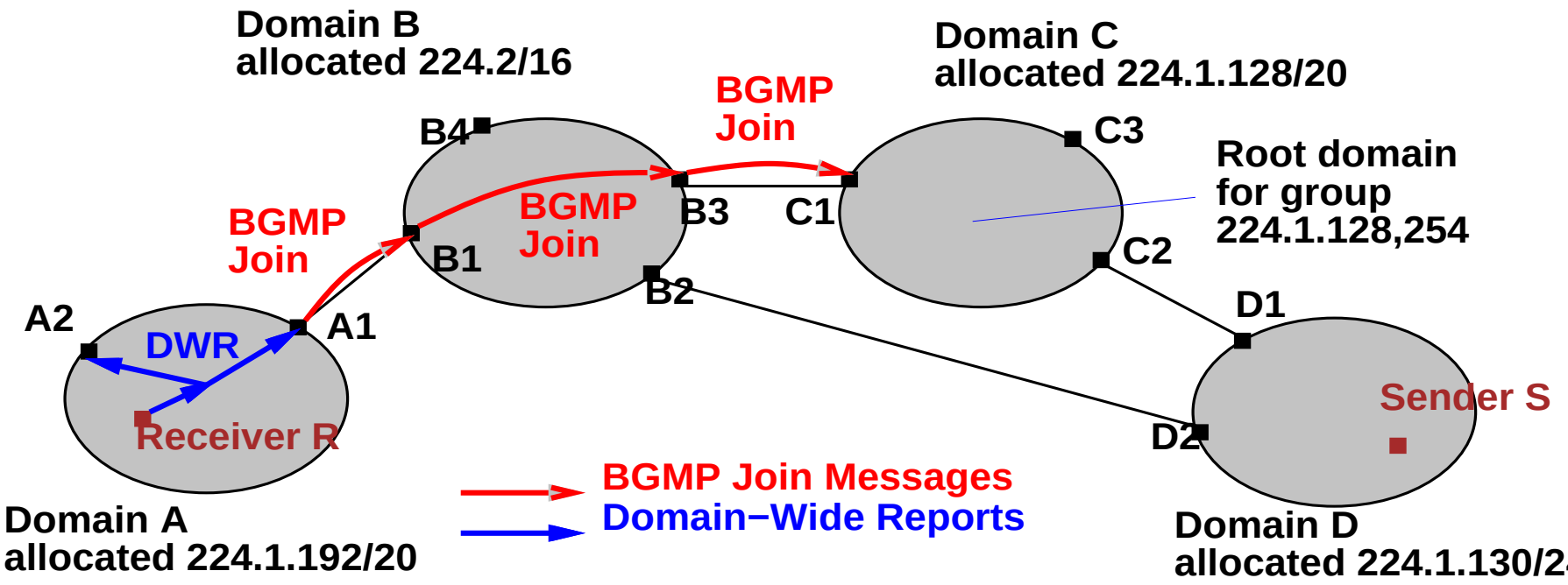
- **If multicast addresses assigned to domains are allocated dynamically and hierarchically (eg with MASC):**
  - We can advertise routes in a routing table for the class-D address ranges.
  - We call this the G-RIB (Group Routing Information Base).
  - It agregates very nicely because of the way it's allocated.
- **We can then build a sparse-mode multicast routing protocol that uses this information to route join messages towards a virtual core.**
  - Finally a way that solves the rendezvous (RP-location) problem without needing to have hosts know anything about routing!

BGMP - Interdomain Multicast

BGMP stands for ***Border Gateway Multicast Protocol***

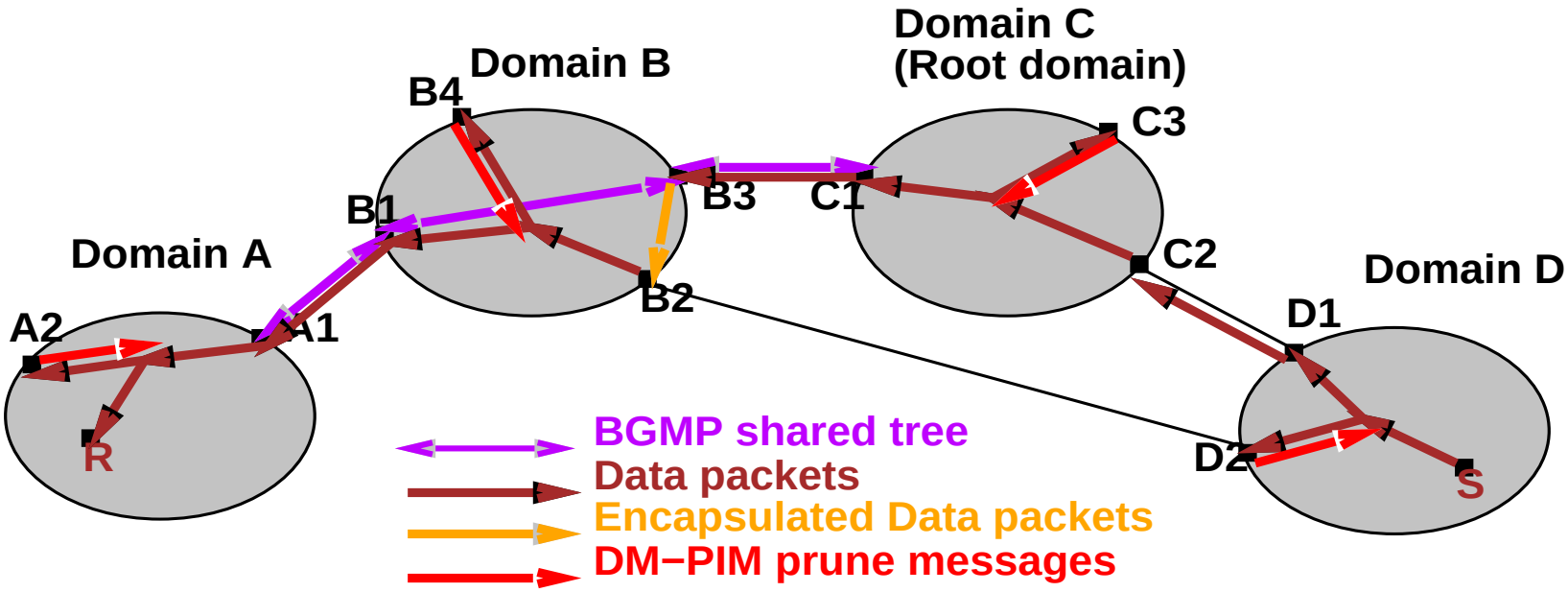
- **It's roughly analogous to BGP for unicast routing**
  - Introduces hierarchy to multicast routing
  - Provides interoperability between other multicast routing protocols
  - Allows DVMRP, PIM, etc to be deployed on scales where they work well
- **Due to the tie-in with address allocation it provides a scalable solution to the rendezvous problem.**
- **Also it should help us reduce multicast forwarding state slightly by dynamic aggregation.**

BGMP - Interdomain Multicast



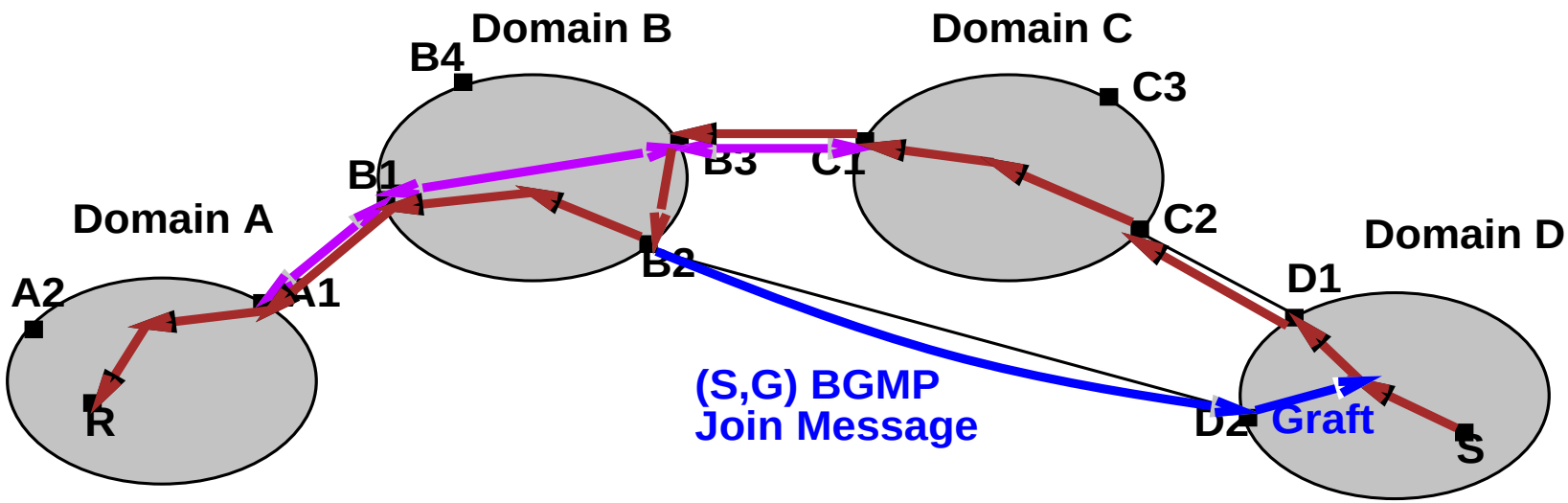
- In this example, each domain runs PIM-DM, with BGMP running inter-domain.
- Receiver R joins the group. A Domain Wide Report generated by its local router informs domain A's BGMP border routers.
- This causes a BGMP join message to be sent towards the root-domain for the group.

BGMP - Shared Tree of Domains



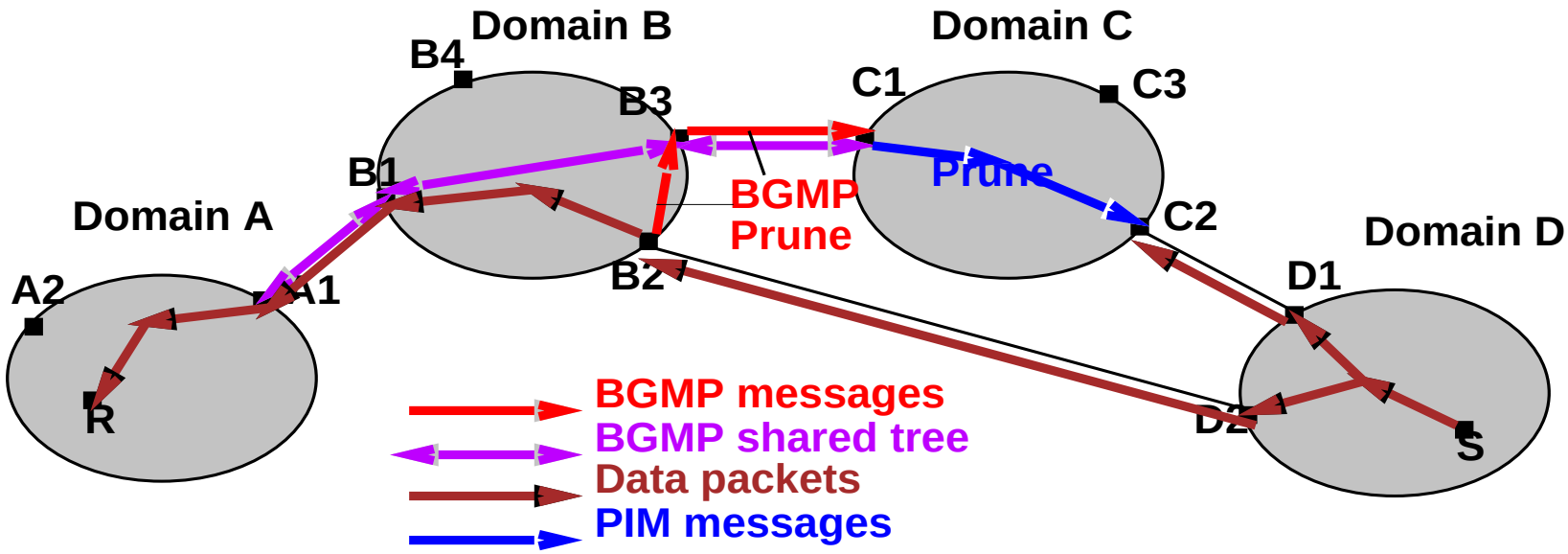
- Sender S starts transmitting.
- The packets are flooded to D's border routers. BGMP router D1 sends the packets towards the root domain for the group. From there, they follow the shared tree of domains to domain A.
- Within each domain, PIM-DM floods and prunes.
- B3 encapsulates to B2 because it is not the correct entry router for domain B.

BGMP - Shortest Path Branch



- To prevent the encapsulation and optimize the tree, B2 sends a source-specific BGMP join towards A.

BGMP - Shortest Path Branch



- Once traffic starts flowing from D2 to B2, a source specific BGMP prune is sent to B3 to stop the encapsulation and prune off the shared tree.
- Data now flows along a more optimal path.
- Not all topologies can do this...

BGMP Root Domain Location

- **BGMP uses a hierarchical division of the multicast address space to know where the root domain for each address prefix is located.**
- **This information could be statically configured, but we expect to it be generated dynamically using MASC address allocation, and distributed to BGMP border routers using MBGP.**

- **For more information:**
  - BGMP and MASC Internet Drafts give protocol details
  - There was architecture overview paper in Sigcomm 98.

An Inter-domain Multicast Architecture

- **Finally we have an inter-domain multicast routing architecture that scales and maintains Steve Deering's very nice Internet Standard multicast service model:**
  - Receivers announce their interest
  - Senders just send
  - Routers conspire to deliver data from senders to receivers

- **The conspiracy is a more complex than we expected:**
  - AAP for intra-domain address allocation
  - MASC for inter-domain address allocation
  - MBGP to distribute the G-RIB
  - BGMP to use the G-RIB to build inter-domain multicast trees
  - DVMRP/PIM/CBT/MOSPT/whatever for inter-domain multicast routing.

Aren't there simpler solutions?

- **Only one with the Internet Standard service model:**
  - HPIM would have worked, but builds less efficient trees.
- **What if we change the service model?**
- **Two noteworthy solutions:**
  - Express (Holbrook/Cheriton)
  - Simple Multicast (Perlman/???)

Express

- **Very similar to SM-PIM, except:**
  - Hosts cannot join a group without specifying a source.
  - Hosts only ever join to sources.
- **Receivers must find out who the sources are through some other mechanism such as web or being refered by another source.**
- **Express multicast only solves the one-to-many distribution problem, noty the binding/rendezvous problem.**
  - Cannot build sdr-style directory.
  - Cannot use it for resource location.
  - Source-discovery is application-level: unlike unicast IP, sources need to register in some way through the app-level directory before they can send a multicast packet.
- **It is simple though!**
  - Address allocation is local to a source

Simple Multicast

- **Very similar to CBT, expect:**
  - Hosts must list the core address in their join messages.
  - Join to the core building a bidirectional shared tree.

- **Receivers must find out the core address through whatever mechanism they use to find the group address.**
- **Hosts need changing to allow them to inform their local routers what the core address is.**

- **Can build session directories and use it for resource discovery, but introduces a dependency on a remote core that ISPs say is unacceptable for PIM!**
  - A lot of mechanism is needed to detect core failures and switch t backup core/addresses.

- **Address allocation is local to a core.**
  - Makes allocation fairly simple, although still need a protocol.
  - Increases address space at the expense of needing to carry core address in data packets as an option.

So what should multicast routing do?

- **BGMP/MASC allows the routers to do all multicast routing.**
  - Preserves the service model.
  - Allows failures to be routed around.
  - Hosts don't know about routing.

- **Express only builds source-based trees.**
  - Pushes rendezvous into applications.
  - Hosts must join sources directly.

- **Simple Multicast builds shared trees to a named core.**
  - Pushes core discovery and failover mechanisms into applications.
  - Hosts must join group/core pair directly.