The MASC/BGMP Architecture for Inter-domain Multicast Routing

Satish Kumar (USC), Pavlin Radoslavov (USC), Dave Thaler (Merit), Cengiz Alaettinoglu (ISI), Deborah Estrin (ISI), Mark Handley (ISI)

Current multicast situation (one big domain) does not scale

- Address allocation: can collide with anyone in the world
- Route distribution: exponential increase in Mbone routes
- Tree construction: source-tree protocols floods data, membership; shared-tree protocols flood core lists

Solution: divide net into domains

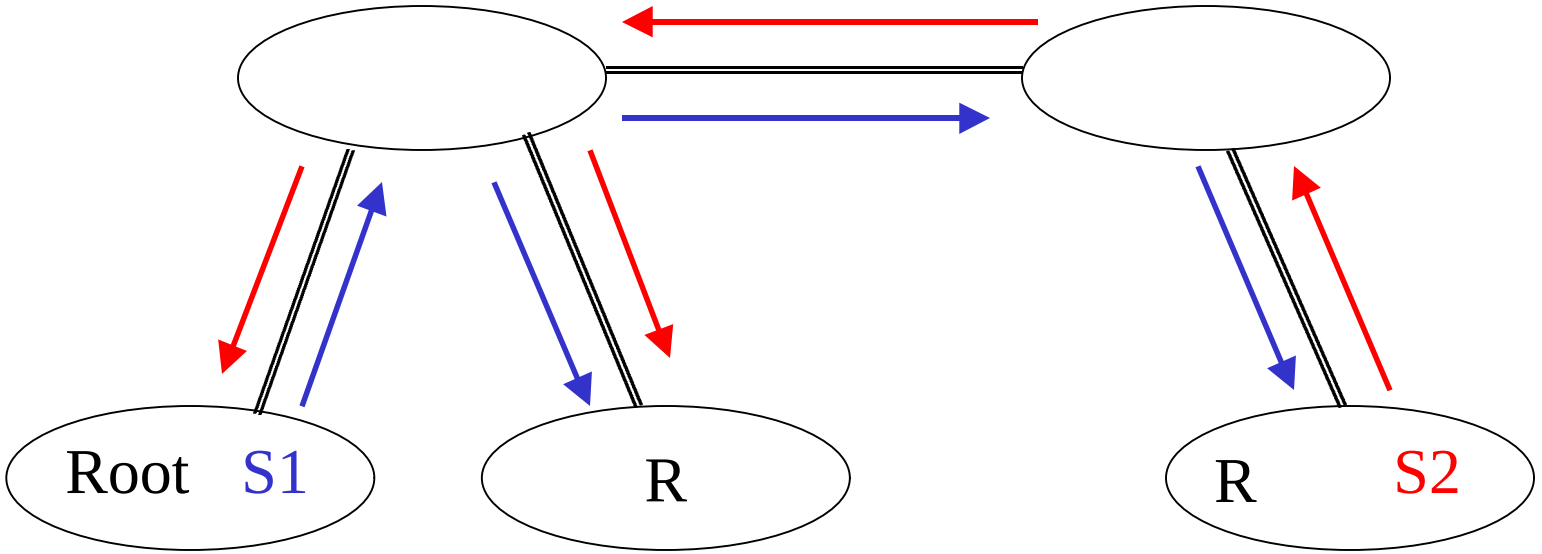
- Similar to solution for unicast (e.g. BGP)
- Domain autonomy adds stability and enables policy control
- Inside domains, can use existing mechanisms for address allocation, routing, and tree construction
- Between domains, need policy control

Also need to minimize “third-party dependencies” such as:

- Relying on PIM Rendezvous Point in someone else’s domain for general “infrastructure” groups (SDR, NTP, mtrace, etc)
- Data loss over another provider’s link
- Address allocation via single global authority

Goals

- Construct group-shared trees rooted in group initiator's domain
- Use bidirectional trees to minimize third-party dependencies



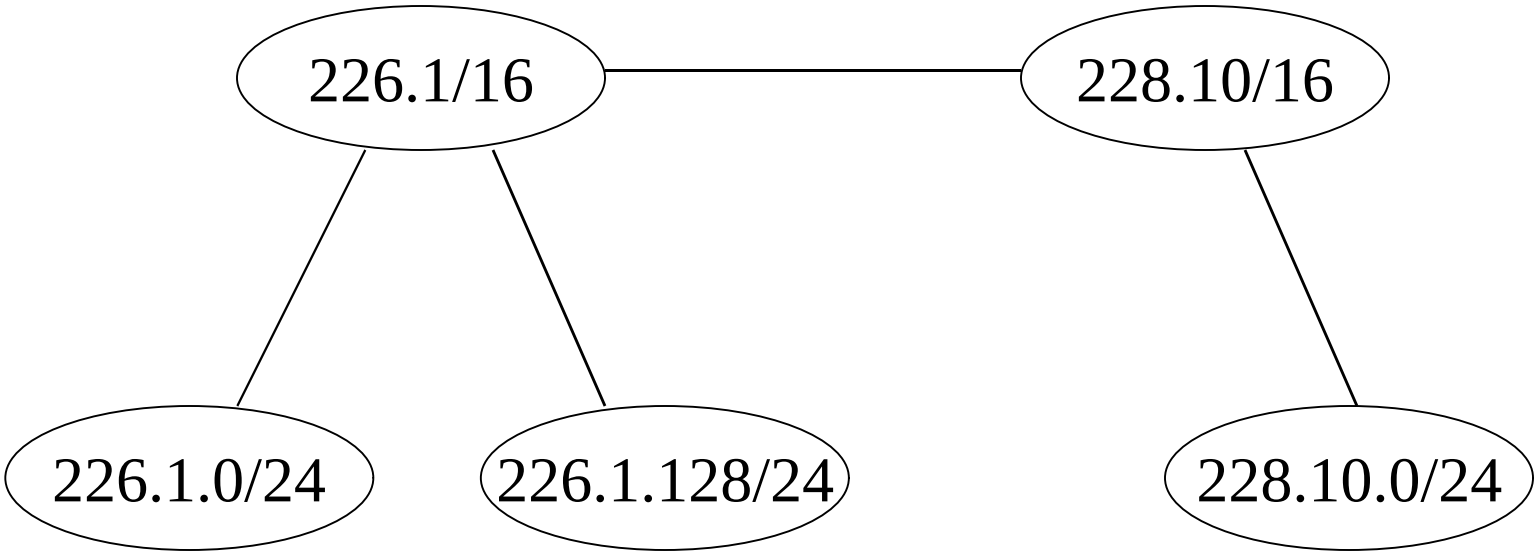
Goals (cont.)

- Use simple scalable mapping of group to tree root
- Add topological significance to group addresses to allow state aggregatability

Three parts to solution

- MASC associates aggregatable group-prefixes with domains
- BGP distributes routes to those group prefixes (“group routes”) subject to policy
- BGMP constructs bi-directional shared trees of domains

MASC associates aggregatable group-prefixes with domains

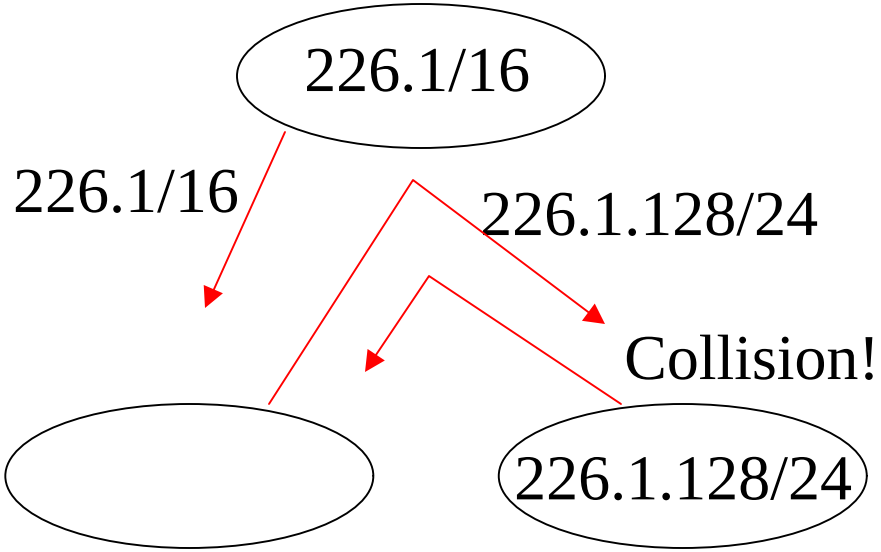


Allocations must be dynamic to adapt to usage patterns

MASC uses a claim-collide mechanism (summary)

- Claimant learns parent prefix and lifetime
- Claimant chooses a *sub-prefix* and lifetime
- Claimant sends claim to parent (if any) and siblings
- Claimant listens for collisions
- After timeout, claimant can use prefix
- Timeout based on maximum partition time

MASC uses a claim-collide mechanism (example)

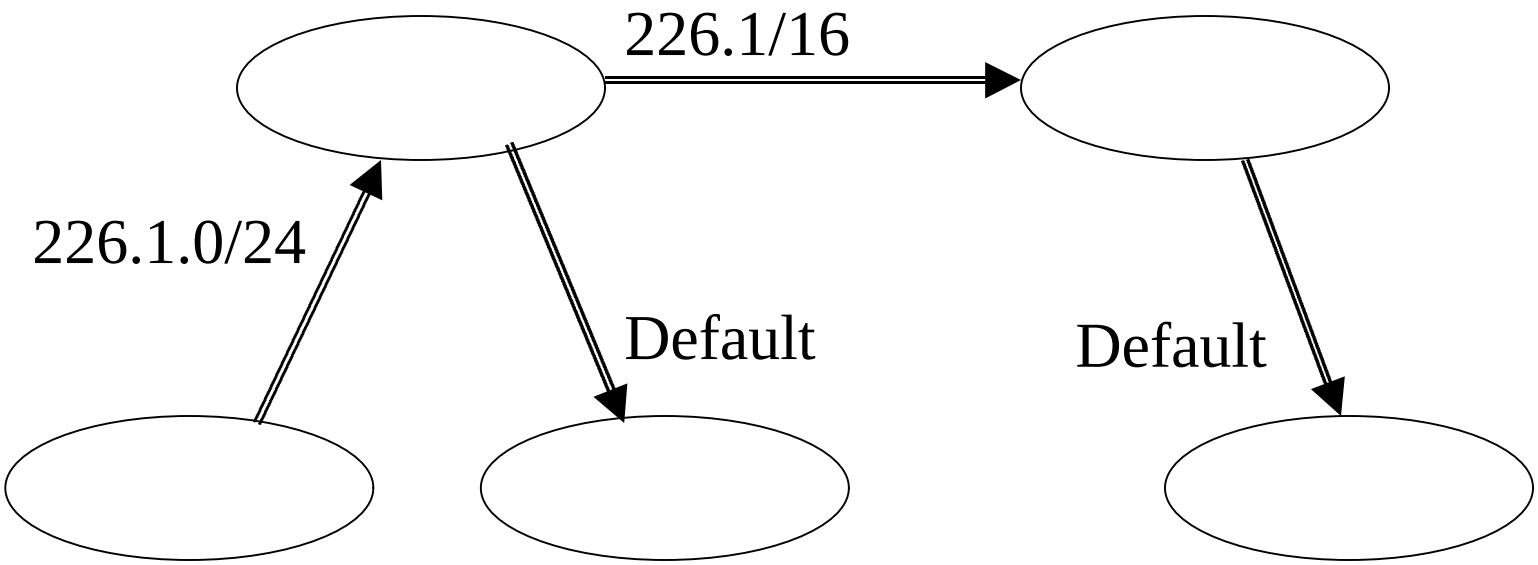


Collision causes loser to choose another prefix

Why claim-collide?

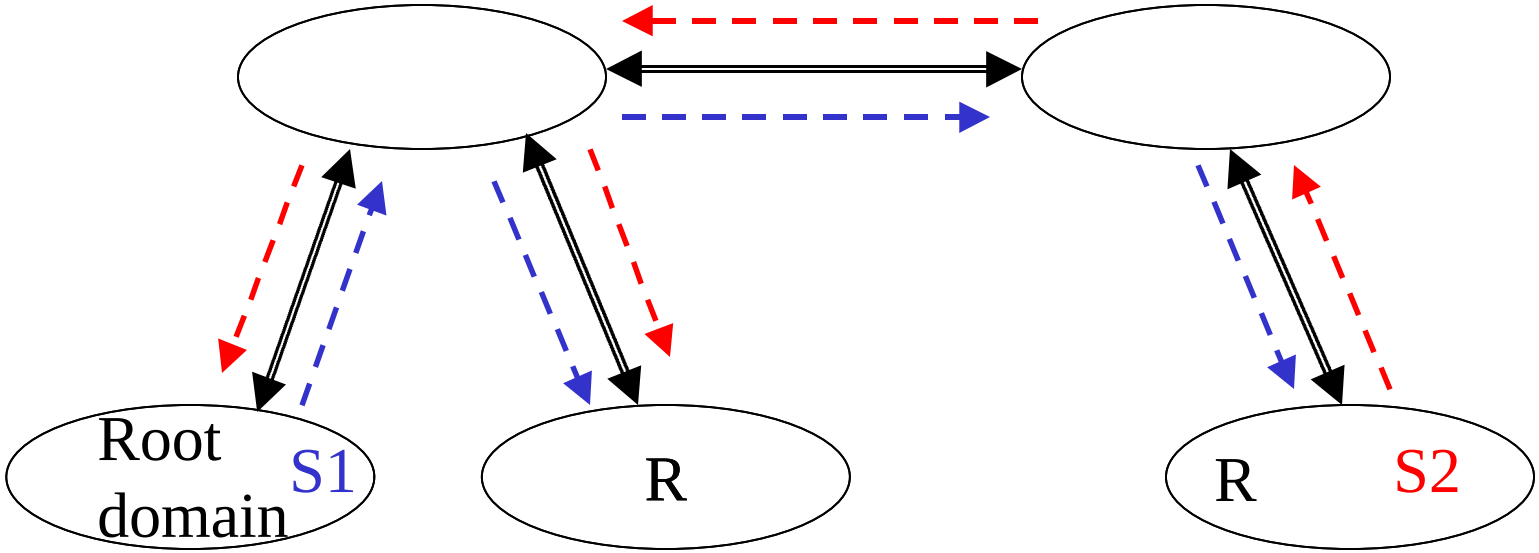
- Query-response has third-party dependency at top level
- Query-response with multiple servers introduces synchronization complexity
- Claim-collide is same at all levels
- Claim-collide appears simpler, and more robust

Multiprotocol BGP distributes group routes subject to policy



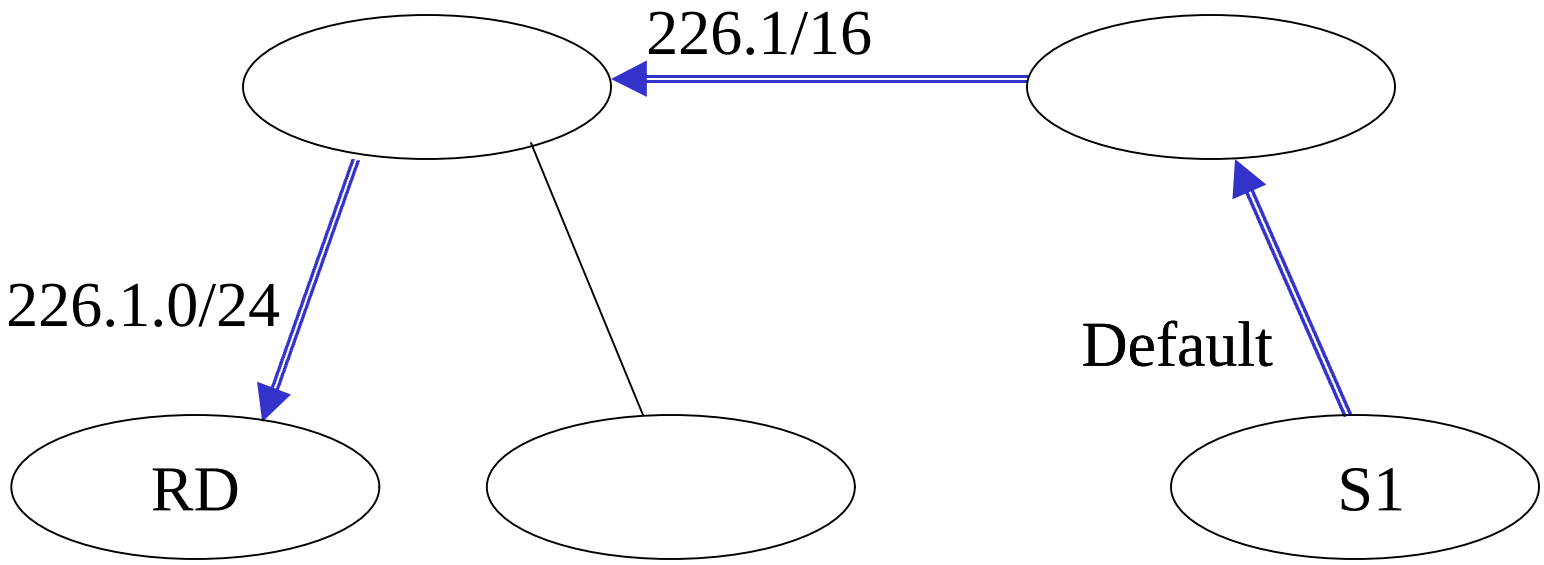
Policy is realized through selective propagation of group routes

BGMP constructs bi-directional shared trees



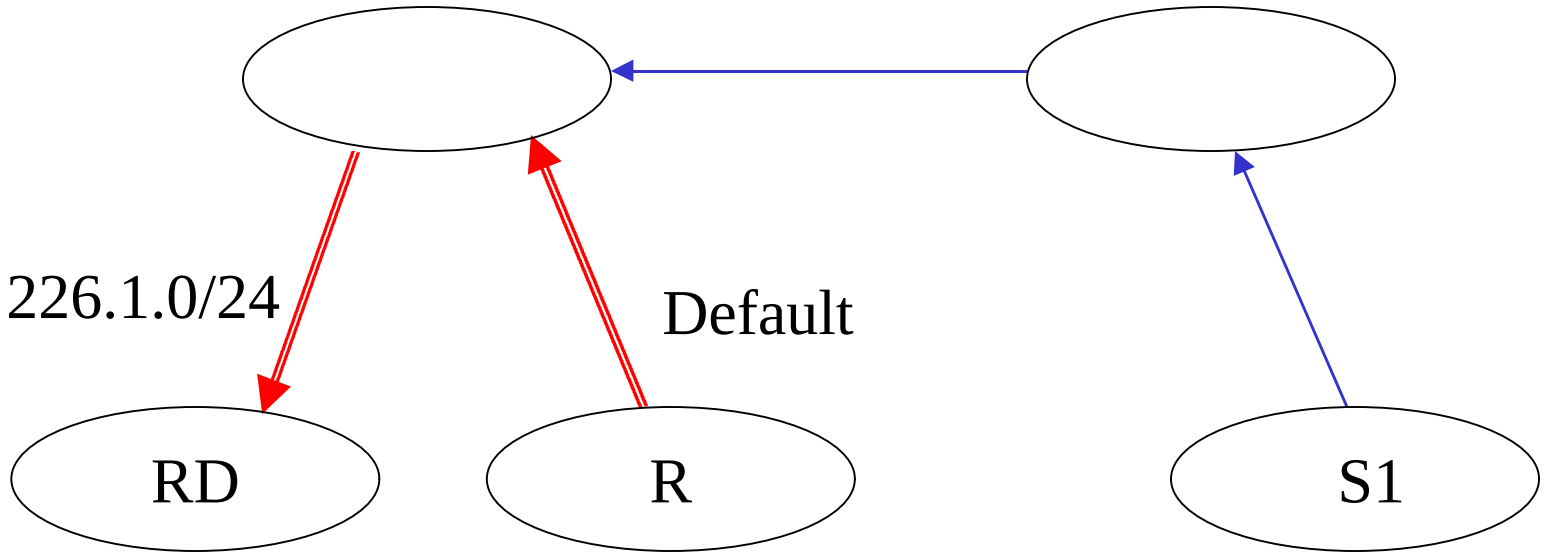
- A group's tree is rooted at the creator's domain (not a single router)
- BGMP uses intra-domain routing inside

Data from sender-only domains just follows group routes



Forwarding occurs as in unicast

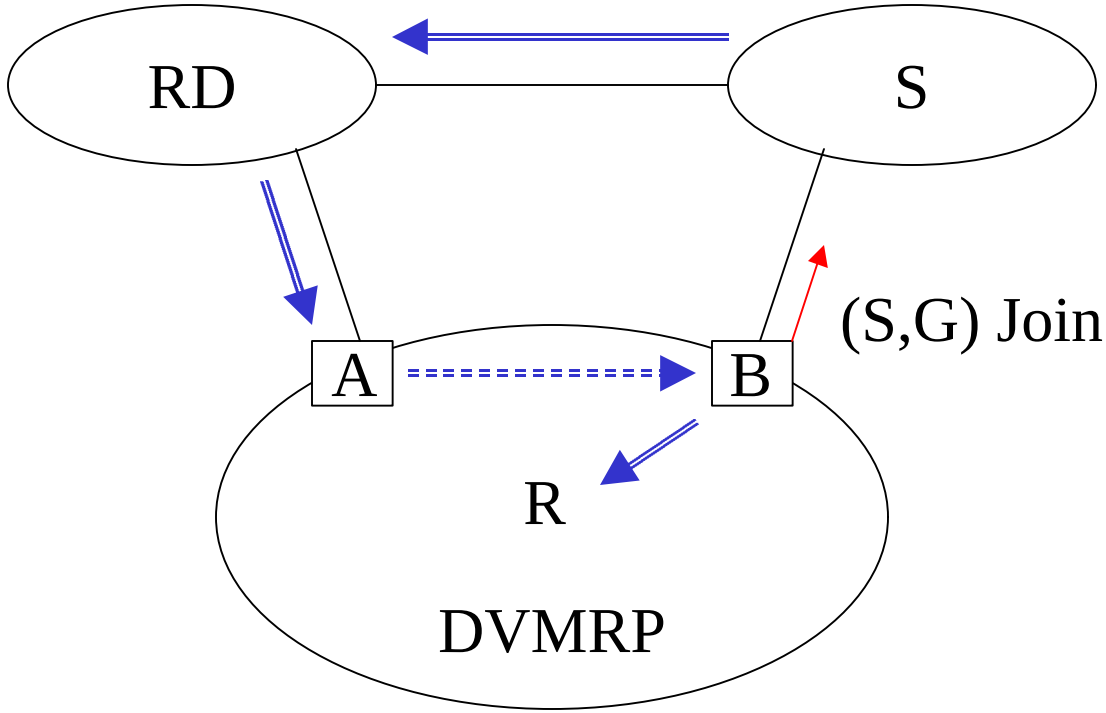
Joins from receiver domains also follow group routes



Root domain
for 226.1.0/24

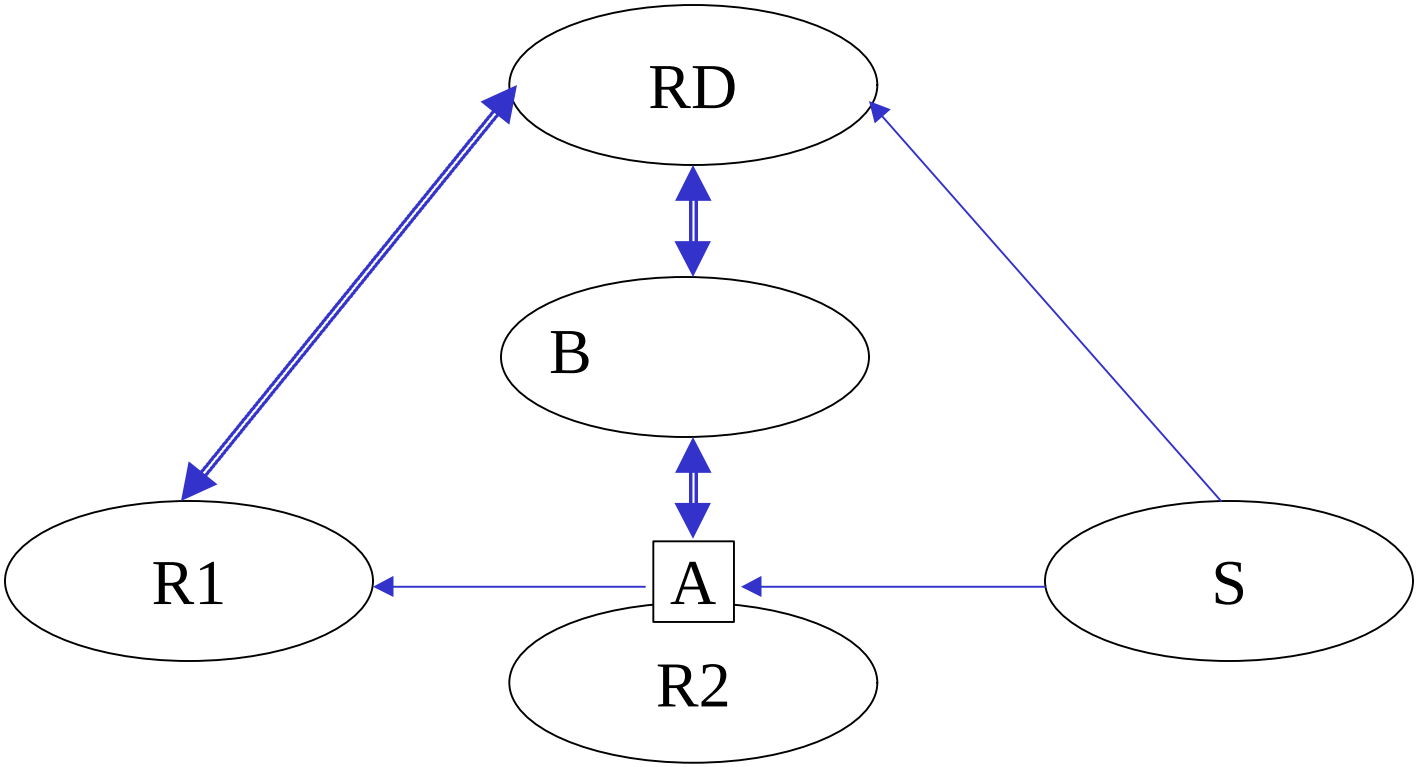
Receiver
joins 226.1.0.3

SPT-based domains require encapsulation from group tree



Encapsulation is avoided with source-specific branch

Source *trees* are incompatible with bidirectional (*,G) trees



Duplicates or black holes can form!

BGMP's (S,G) branch stops at first on-tree router

- Result is a "Hybrid" bidir shared tree with some unidir (S,G) branches

- Hybrid tree path 20% longer than SPT
- Bidir shared tree path 30% longer than SPT
- Unidir shared tree path 100% longer than SPT

BGMP/MASC Architecture Summary

- Third-party dependencies are minimized
- Use of BGP allows policy control of trees
- Topological significance of group addresses allows state aggregation
- Source-specific branches avoid encapsulation without causing loops