IP Multicast

Luca Veltri

(mail.to: luca.veltri@unipr.it)

Corso di Reti di Telecomunicazioni C, a.a. 2008/2009

http://www.tlc.unipr.it/veltri

Unicast/Multicast/Broadcast Communications

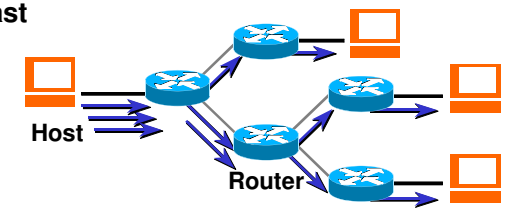
- Unicast: one to one communication
  - **one source and one destination (one-to-one relation)**
  - **in unicast routing, when a node receive a packet, it forwards the packet through only one output interface (link)**
  - **in connection-less (datagram) communications single unicast address is used for the destination**
- Multicast: (potentially) many to many communication
  - **one source and a group of destinations (one-to-many relation)**
  - **in multicast routing, when a node receive a packet, it may forward the packet through several output interfaces (links)**
  - **replication happens inside network nodes (routers or switches)**
  - **in connection-less (datagram) communications a group address or an address list is used for the destinations**
  - **may require dynamic management of group memberships**
- Broadcast: one to all communication
  - **one source, but all the others are the destinations**
  - **a broadcast address is used as destination**

Multiple Unicast Communications

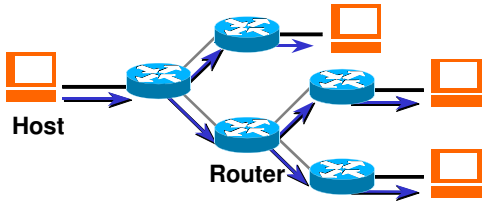
- Not all networks natively support multicast or broadcast
- When needed, multicast can be emulated through multiple unicast
- Multiple Unicast: one-to-many communication through multiple one-to-one communications
  - **each packet is duplicated and sent separately to every destinations**
- Big problem with multi unicast communications: bandwidth waste with multiple data flows
  - **with N receivers, sender must replicate the stream N times.**
  - **Consider good quality audio/video streams are about 1.5Mb/s**
  - **Each additional receiver requires another 1.5Mb/s of capacity on the sender network**

Multi Unicast vs Multicast

Multi Unicast

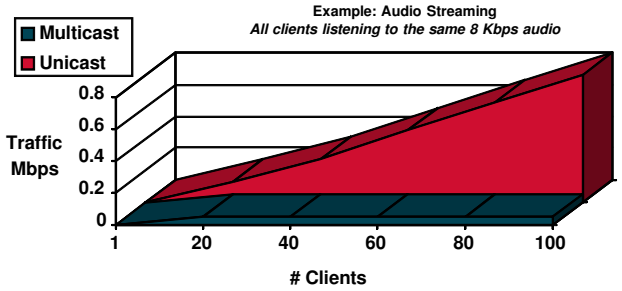


Multicast



Why Multicast?

- Efficiency and Performance
  - **Eliminates traffic redundancy - Better bandwidth utilization**
  - **Controls network traffic**
  - **Lesser host/router processing - reduces server CPU and output link loads**
- Distributed Applications
  - **Makes multipoint applications possible (same data to multiple receivers)**
- Receivers' addresses unknown



Multicast Applications

- Typical applications
  - **Multimedia conference (video, audio, digital whiteboard)**
  - **Video Distribution**
  - **Distance Learning**
  - **Software/File Distribution**
  - **Replicated Database Updates**
  - **Resource discovery (e.g., auto-topology)**
  - **Commercial apps (e.g., transactions, news distribution)**
  - **Routing protocols (e.g., both EIGRP and OSPF use multicast to send updates to neighbors)**
  - **Games (e.g., distributed arcades)**
  - **etc.**
- Examples
  - **VIC - Video conferencing**
  - **VAT/RAT - Audio conferencing**
  - **WB - Whiteboard**

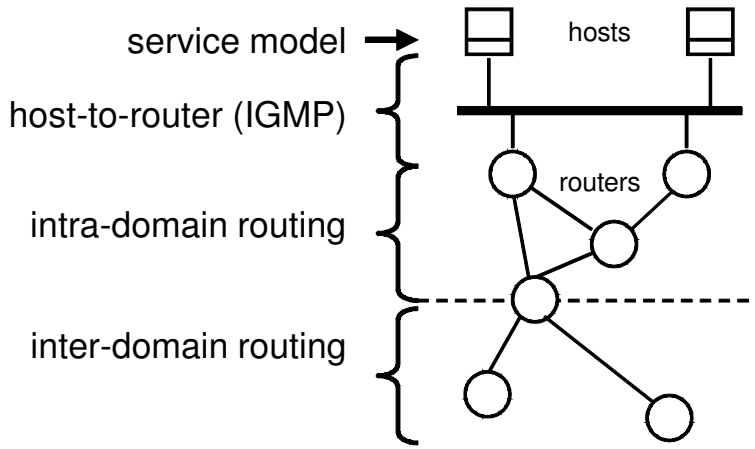
Disadvantages

- Possible packet duplication
  - **in case of path redundancy between source and destinations**
  - **latency in convergence of multicast protocols**
- Unreliable delivery
  - **TCP works only with unicast connections**
  - **UDP is normally used as transport protocol (best effort)**
- Possible network congestion
  - **no congestion control with UDP**
- Hence, more work is needed at application level, or new multicast transport protocols should be implemented on top of IP multicast
  - **however, reliable multicast is still an area open for much research**

Principles of IP Multicast

- Special IP addresses are used to identify multicast groups
- Hosts notify multicast routers about the multicast groups to which they (want to) belong
- Multicast groups are managed by the routers using multicast routing protocols

Components of IP Multicast

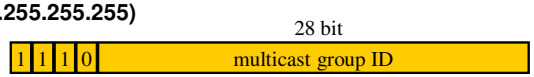


Original IP Multicast Service Model (RFC-1112)

- Each multicast group identified by a class D IP address
- A multicast IP packet reaches a subset (group) of hosts on the network; those hosts have indicated an interest in the multicast group address
- groups may be of any size
- Members of the group could be present anywhere in the Internet
- Members join and leave the group and indicate this to the routers
- Senders and receivers are distinct, i.e., a sender need not be a member
- Routers listen to all multicast addresses and use multicast routing protocols to manage groups

Multicast Group Addresses

- IP addresses from 224.0.0.0 to 239.255.255.255 are designated as multicast addresses
  Class D (224.0.0.0 - 239.255.255.255)



- Group addresses have inherent scope:
  - **Reserved for link scope: 224.0.0.0 -- 224.0.0.255**
    - These are never forwarded by any router
    - Some special addresses
      - 224.0.0.1: all multicast systems on a subnet
      - 224.0.0.2: all multicast routers on a subnet (224.0.0.22 = all IGMPv3 routers)
  - **Global scope (Internet-wide): 224.0.1.0 -- 238.255.255.255**
    - Can be delivered throughout the Internet
  - **Administrative/Limited scope (local): 239.0.0.0 -- 239.255.255.255**
    - Not forwarded beyond an organization's intranet (like RFC 1918)
    - Reusable

Multicast Group Addresses (cont.)

- Need to map multicast group addresses to to data link multicast addresses (e.g. Ethernet)
  - **RFC 1112 defines OUI 0x01005e (24bit)**
  - **25th bit is set to 0 (value 1 is reserved for further uses)**
  - **Low-order 23-bits of IP address map into low-order 23 bits of IEEE address (eg. 224.2.2.2–01005e.020202)**
- Ethernet and FDDI use this mapping

IP Multicast Service — Sending

- Uses normal IP-Send operation, with an IP multicast address specified as the destination
- Must provide sending application a way to:
  - **specify outgoing network interface, if >1 available**
  - **specify IP time-to-live (TTL) on outgoing packet**

IP Multicast Service — Receiving

- Two new operations:
  - **Join-IP-Multicast-Group (group-address, interface)**
  - **Leave-IP-Multicast-Group (group-address, interface)**
- Receive multicast packets for joined groups via normal IP-Receive operation

TTL Scope

- As in unicast IP forwarding, the TTL field of a multicast IP packet is decremented by every router that forwards it
- A source can choose any TTL for multicast packets that it transmits
- TTL controls how far (multicast) packets travel before being dropped (limiting the packet scope)
  - **prevents clients that are really far away from source**
- Router interfaces can be configured to drop multicast packets with a TTL less than some arbitrary (positive) value, rather than allowing the TTL to count down to zero
  - **Cisco IOS interface configuration command:**
    - ip multicast ttl-threshold 16
  - **These thresholds create TTL scope boundaries**
    - E.g. packets must start with a TTL of at least 16 to be forwarded beyond the campus network

TTL Scope (cont.)

- General standards for TTL (from IETF)
  - 1 for local net
  - 15 for site
  - 63 for region
  - And 127 for world

Internet Group Management Protocol (IGMP)

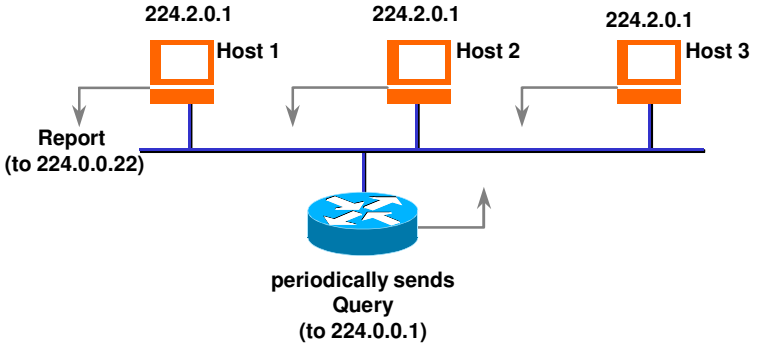
- The protocol by which hosts indicate their multicast group memberships (i.e. interest in receiving packets addressed to a particular multicast group G) to neighboring routers
- Routers solicit group membership from directly connected hosts
- IGMP messages aren't forwarded by routers
- IGMP was originally defined in RFC 1112
- IGMP v2 and IGMP v3 enhancements
  - **RFC 1112 specifies version 1, the original standard**
  - **RFC 2236 specifies version 2, the most widely used, backward-compatible with version 1**
  - **RFC 3376 specifies version 3, new standard, backward-compatible with version 2 and 1**

IGMPv3 Overview

- IGMP is an asymmetric protocol, specifying separate behaviors for group members (hosts or routers that wish to receive multicast packets) and multicast routers
  - **a multicast router that is also a group member performs both parts of IGMPv3**
  - **a system performs the protocol operation over all interfaces on which multicast reception is supported**
- The purpose of IGMP is to enable each multicast router to learn, for each of its directly attached networks, which multicast addresses are of interest to the systems attached to those networks
  - **IGMP version 3 added the capability for a multicast router to also learn which** sources **are of interest to neighboring systems, for packets sent to any particular multicast address**

- Multicast routers send General Queries periodically to request group membership information from an attached network
  - **These queries are sent to multicast address 224.0.0.1 (the all systems group). All hosts (and routers) listen to this group**
  - **Systems respond to these queries by reporting their group membership state (and their desired set of sources) with Current-State Group Records in IGMPv3 Membership Reports**
- One router on every subnet is designated as the IGMP Querier
  - **The querier is responsible for sending membership queries to the subnet to determine group membership**
  - **All routers on the subnet listen to the membership reports sent by hosts and maintain forwarding states accordingly, regardless of which router is the querier**

Membership Query/Report



IGMP Version 1

- Queries
  - **Querier sends IGMP query messages to 224.0.0.1 with ttl = 1**
  - **One router on LAN is designated/elected to send queries**
  - **Query interval 60–120 seconds**
- Reports
  - **IGMP report sent by one host suppresses sending by others**
  - **Restrict to one report per group per LAN**
  - **Unsolicited reports sent by host, when it first joins the group**

IGMP Version 2

- Changes from version 1:
  - **new message and procedures to reduce "leave latency"**
    - Host sends leave message if it leaves the group and is the last member (reduces leave latency in comparison to v1)
    - Router sends G-specific queries to make sure there are no members present before stopping to forward data for the group for that subnet
  - **Standard querier election method specified**
  - **Version and type fields merged into a single field**
- Backward-compatible with version 1
- Widely implemented

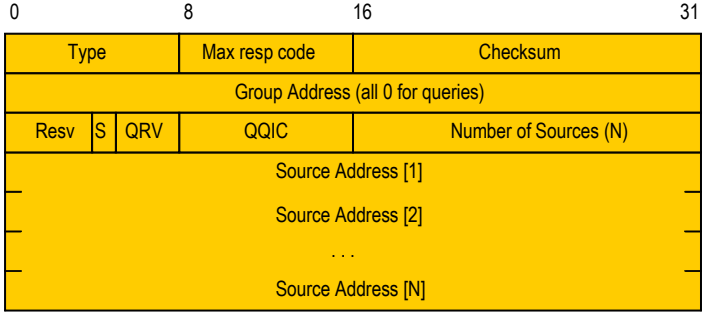
IGMP Version 3

- Changes from version 2:
  - **extension of service interface and protocol to enable hosts to:**
    - listen to only a specified set of hosts sending to a group
    - listen to all but a specified set of hosts sending to a group
  - **in IGMPv1 and IGMPv2, a host would cancel sending a pending membership reports if a similar report was observed from another member on the network;**
    in IGMPv3, this suppression of host membership reports has been removed
    - routers may want to track per-host membership status to implement fast leaves as well as track membership status for possible accounting purposes
    - report suppression does not work well on bridged LANs
    - simpler host's implementations (fewer messages to process)
- Backward-compatible with versions 1 & 2

IGMPv3 messages

- There are two IGMP message types of concern to the IGMPv3 protocol:
  - **Membership Query (type 0x11)**
    - Membership Queries are sent by IP multicast routers to query the multicast reception state of neighboring interfaces
  - **Version 3 Membership Report (0x22)**
    - Version 3 Membership Reports are sent by IP systems to report (to neighboring routers) the current multicast reception state, or changes in the multicast reception state, of their interfaces
- An implementation of IGMPv3 MUST also support the following three message types, for interoperation with previous versions of IGMP:
  - **Version 1 Membership Report (0x12)**
  - **Version 2 Membership Report (0x16)**
  - **Version 2 Leave Group (0x17)**
- IGMP messages are encapsulated in IPv4 datagrams, with an IP protocol number of 2
- Every IGMP message is sent with IP TTL=1

IGMPv3 Membership Query Message



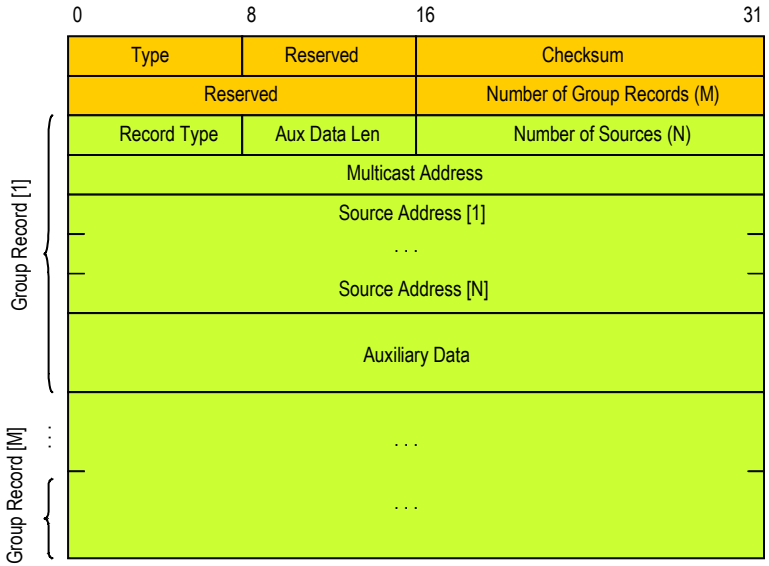
- Type: IGMP message type = 0x11 (Membership Query)
- Max Resp Code: max allowed time before sending a responding report
  - **the actual time allowed, called the Max Resp Time, is represented in units of 1/10 second**
- Checksum: The Checksum is the 16-bit one's complement of the one's complement sum of the whole IGMP message (the entire IP payload)

- The Group Address: is set to zero when sending a General Query, and set to the IP multicast address being queried when sending a Group-Specific Query or Group-and-Source-Specific Query
- Resv: reserved field
- S Flag: Suppress Router-Side Processing
  - **indicates to any receiving multicast routers that they are to suppress the normal timer updates they perform upon hearing a Query**
- QRV: Querier's Robustness Variable
  - **If non-zero, it contains the Robustness Variable value used by the querier; Robustness Variable indicates the number of retransmissions**
  - **this field allow synchronization on non-Queriers**
- QQIC: Querier's Query Interval Code
  - **specifies the Query Interval used by the querier, that is the interval between General Queries sent by the Querier (Default: 125 seconds)**
- Source Addresses: vector of N IP unicast addresses, where n is the value in the Number of Sources (N) field

Query Variants

- There are three variants of the Query message:
  - **A "General Query" is sent by a multicast router to learn the complete multicast reception state of the neighboring interfaces**
    - In a General Query, both the Group Address field and the Number of Sources (N) field are zero.
  - **A "Group-Specific Query" is sent by a multicast router to learn the reception state, with respect to a *single* multicast address, of the neighboring interfaces**
    - In a Group-Specific Query, the Group Address field contains the multicast address of interest, and the Number of Sources (N) field contains zero
  - **A "Group-and-Source-Specific Query" is sent by a multicast router to learn if any neighboring interface desires reception of packets sent to a specified multicast address, from any of a specified list of sources**
    - In a Group-and-Source-Specific Query, the Group Address field contains the multicast address of interest, and the Source Address [i] fields contain the source address(es) of interest

IGMPv3 Membership Report Message



- Type: IGMP message type = 0x22 (Version 3 Membership Report)
- Multicast Address: contains the IP multicast address to which this Group Record pertains
- Group Record Type:
  - **Current-State Record, in response to a Query**
    - MODE_IS_INCLUDE (1)
    - MODE_IS_EXCLUDE (2)
  - **Filter-Mode-Change Record, unsolicited**
    - CHANGE_TO_INCLUDE_MODE (3)
    - CHANGE_TO_EXCLUDE_MODE (4)
  - **Source-List-Change Record, unsolicited**
    - ALLOW_NEW_SOURCES (5)
    - BLOCK_OLD_SOURCES (6)

IGMPv3 Protocol for Group Members

- At host side, there are two types of events that trigger protocol actions on an interface:
  - **a change of the interface reception state, caused by a local invocation of *IPMulticastListen* by upper layer**
  - **reception of a Query**
- Both events may trigger the sending of IGMPv3 Reports
- IGMPv3 Reports are sent with an IP destination address of 224.0.0.22
  - **to which all IGMPv3-capable multicast routers listen**
- A change of interface state causes the system to immediately transmit a State-Change Report from that interface
  - **the type and contents of the Group Record(s) in that Report are determined by comparing the filter mode and source list for the affected multicast address before and after the change**

- The reception of a Query causes the system to send a response
  - **the system does not respond immediately**
  - **it delays its response by a random amount of time, bounded by the Max Resp Time value derived from the Max Resp Code in the received Query message**
  - **Before scheduling a response to a Query, the system must first consider previously scheduled pending responses and in many cases schedule a combined response; the system must be able to maintain the following state:**
    - A timer per interface for scheduling responses to General Queries
    - A per-group and interface timer for scheduling responses to Group-Specific and Group-and-Source-Specific Queries
    - A per-group and interface list of sources to be reported in the response to a Group-and-Source-Specific Query

- When the interface timer expires, one Current-State Record is sent for each multicast address for which the specified interface has reception state
  - **The Current-State Record carries the multicast address and its associated filter mode (MODE_IS_INCLUDE or MODE_IS_EXCLUDE) and source list**
  - **Multiple Current-State Records are packed into individual Report messages**
  - **Instead of using a single interface timer, implementations are recommended to spread transmission of such Report messages over the interval [0, MaxRespTime]**

- When a group timer expires
  - **if the list of recorded sources for the that group is empty (i.e., it is a pending response to a Group-Specific Query) and the interface has reception state for that group address, then a single Current-State Record is sent for that address**
    - The Current-State Record carries the multicast address and its associated filter mode (MODE_IS_INCLUDE or MODE_IS_EXCLUDE) and source list
  - **if the list of recorded sources for that group is non-empty (i.e., it is a pending response to a Group-and-Source-Specific Query) and the interface has reception state for that group address then a Current-State Record is sent**
    - the contents of the Current-State Record is determined from the interface state and the pending response record
    - if the resulting Current-State Record has an empty set of source addresses, then no response is sent

- The maximum value of the query response timer (D) is:
  - **in version 1: D=10 seconds**
  - **version 2 and 3: the membership query contains a field that specifies the value of D, in tenths of a second**

IGMPv3 Protocol for Multicast Routers

- A multicast router performs the protocol operation over each of its directly-attached networks
  - **on each interface the router MUST enable reception of multicast address 224.0.0.22, from all sources**
- Multicast routers need to know only that *at least one* system on an attached network is interested in packets to a particular multicast address from a particular source
  - **a multicast router is not required to keep track of the interests of each individual neighboring system**
  - **however routers may want to track per-host membership status on an interface; this allows routers to implement fast leaves as well as track membership status for possible accounting purposes**
- IGMPv3 is backward compatible with previous versions of the IGMP protocol
  - **IGMPv3 multicast routers MUST also implement versions 1 and 2 of the protocol**

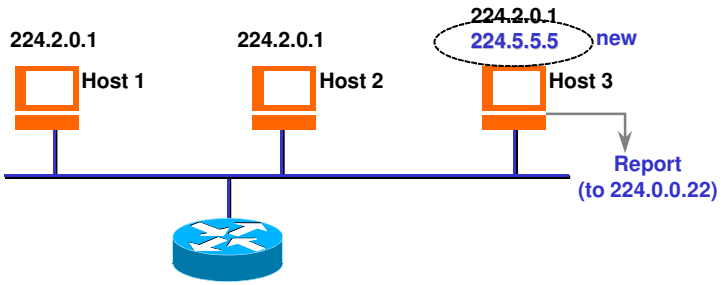
- Multicast routers send General Queries periodically to request group membership information from an attached network
  - **these queries are used to build and refresh the group membership state of systems on attached networks**
- Systems respond to these queries by reporting their group membership state (and their desired set of sources) with Current-State Group Records in IGMPv3 Membership Reports
- To enable all systems on a network to respond to changes in group membership, multicast routers send specific queries
  - **A Group-Specific Query is sent to verify there are no systems that desire reception of the specified group or to "rebuild" the desired reception state for a particular group**
  - **Group-Specific Queries are sent when a router receives a State-Change record indicating a system is leaving a group**

- When a group membership is terminated a multicast router queries for other members of the group or listeners of the source before deleting the group (or source) and pruning its traffic
  - **A Group-Specific Query is sent to verify there are no systems that desire reception of the specified group or to "rebuild" the desired reception state for a particular group**
    - it is sent when a router receives a State-Change record indicating a system is leaving a group
  - **A Group-and-Source Specific Query is used to verify there are no systems on a network which desire to receive traffic from a set of sources**
    - it is only sent in response to State-Change Records and never in response to Current-State Records

Joining a Group

- When a host wishes to join a group G it immediately sends an unsolicited Group Membership Report
  - **Speeds up the join process when the host is the first on the subnet to join group G**
- When any router on a subnet receives a membership report for group G
  - **if there are no states for group G, the router creates (*,G) state, and sets the oif (output interfaces) to the interface on which the report was received**
  - **if one or more states exist for group G, the oif list for every state involving G is updated to include the interface on which the report was received. If the interface is already in the oif list, its timer is refreshed**

Joining a Group: Example

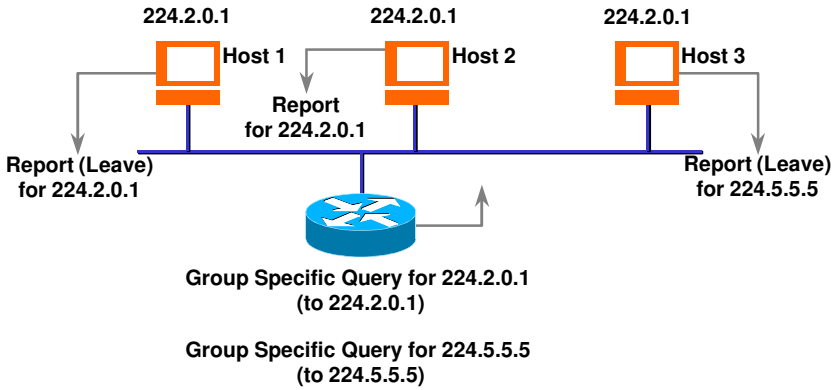


Leaving a Group

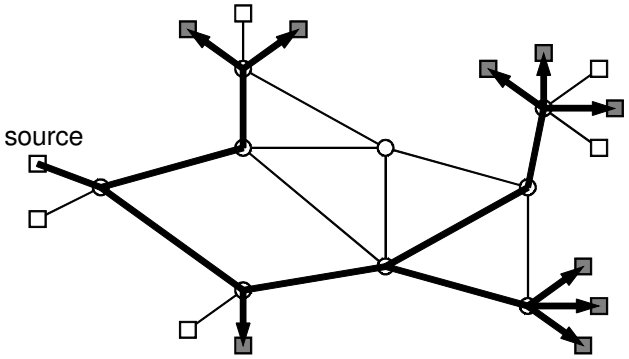
- When an host wishes to leave a group G it may send a Membership Report message (or a Leave Group message, in IGMPv2)
  - **Sent to 224.0.0.22 (all IGMPv3 routers) since other hosts don't care when any particular host leaves**
- The router sends a Group-Specific Membership Query
  - **Addressed to the group, G**
  - **Hosts follow the same rules as for the general query (i.e. delay before replying)**

- When a router does not receive any reply on an interface for a group G for which forwarding states exist:
  - **The outgoing interface timer for every state involving G continues to count down**
  - **When the timer for that interface in a G state expires, the interface is removed from the oif list for that state**

Leaving a Group: Example



Multicast Routing and Functions



- Routing (path determination)
- Packet forwarding and possibly replication
- Dynamic membership - path pruning/grafting

Multicast Routing

- Per il trasporto del traffico multicast occorre che nei nodi della rete geografica siano attivate le funzioni necessarie per l'instradamento multicast (multicast routing)
  - **si deve creare un albero che ha come radice la sorgente e come foglie i menbri del gruppo multicast**
  - **l'istradamento unicast si basava sul solo indirizzo di destinazione - *destination based routing***
  - **l'istradamento multicast si basa sia sull'indirizzo della sorgente che sul gruppo multicast (destinazione) - *source-based routing***
- Sono stati individuati differenti algoritmi di instradamento multicast
  - **muticast routing algorithms**
- Da cui sono stati definiti differenti protocolli di instradamento multicast
  - **muticast routing protocols**

Multicast Algorithms: Flooding (inondazione)

- Estremamente semplice
  - **se il pacchetto è ricevuto per la prima volta, allora il router provvede a replicare il pacchetto ricevuto e a ritrasmetterlo attraverso tutte le proprie interfacce, ad eccezione di quella dalla quale il pacchetto è pervenuto**
- la difficoltà riscontrata consiste proprio nel determinare se il pacchetto è stato effettivamente ricevuto per la prima volta
  - **Una soluzione potrebbe essere quella di tenere traccia di tutti i pacchetti pervenuti al router, ma richiederebbe elevate risorse di memorizzazione e elaborazione**
- L'algoritmo non richiede per il funzionamento informazioni sull'instradamento
  - **non comporta perciò la predisposizione di alcuna tabella di instradamento multicast**
- Utilizzato nei protocolli di instradamento unicast, (e.g. OSPF) per scambiare le informazioni di routing tra i nodi

Multicast Algorithms: Spanning Tree

- Soluzione più efficiente del flooding
- Utilizzata ad esempio dai bridge per interconnettere diverse LAN in modo da evitare percorsi chiusi (loop)
- L'algoritmo agisce in una prima fase per individuare i rami che costituiscono lo spanning tree (l'albero ricoprente)
  - **si identificano le interfacce dei router agli estremi dei rami dell'albero ricoprente**
- Durante l'instradamento ciascun router replica i pacchetti multicast sulle sole interfacce/rami dello spanning tree (ad eccezione dell'interfaccia da cui il pacchetto è arrivato)
- Si evitano cosi' possibili loop
- Traffico solo su una porsione della rete (albero)
- Non distingue il routing per i differenti gruppi multicast

Multicast Algorithms: Reverse Path Forwarding

- Reverse Path Forwarding (RPF)
  - **alla ricezione di un pacchetto multicast un router analizza l'indirizzo della sorgente "S" e quello dell'interfaccia "I" attraverso la quale è arrivato il singolo pacchetto**
  - **se "I" si trova sul percorso più breve verso "S" (ovvero se "I" è l'interfaccia usata dal router per instradare i pacchetti verso "S"), allora il pacchetto è replicato ed è inoltrato verso tutte le interfacce ad eccezione di "I"**
  - **nel caso non si sia verificata la condizione precedente, il pacchetto è scartato**
- L'algoritmo richiede una tabella che indichi per ciascuna sorgente l'interfaccia del nodo sul percorso più breve verso la sorgente
  - **a questo scopo, potrebbe essere utilizzata la tabella di instradamento unicast**
  - **siccome il routing non è in generale simmetrico, alcuni protocolli realizzano una tabella ad hoc**

Multicast Algorithms: RPF and prunes

- Variante dell'algoritmo RPF con "potatura"
- L'albero multicast è potato di tutti i rami a cui non è attestato alcun nodo interessato al gruppo multicast
  - **I nodi foglia senza membri del gruppo "G" inviano uno messaggio di potatura (prune) al router multicast a monte**
  - **il router a monte è così informato che non deve inoltrare ulteriore traffico multicast destinato a G verso il router a valle**
  - **in questo modo, partendo dalle foglie e ripercorrendo l'albero verso la radice, sono potati i rami sui quali è inutile inoltrare traffico**
- L'algoritmo RPF con potatura introduce il concetto di appartenenza ai gruppi e richiede che i router tengano traccia dello stato dell'albero per gruppo e per sorgente
  - **lo stato dell'albero viene aggiornato periodicamente**

Multicast Algorithms: Shortest Tree

- Shortest Path Tree
  - **L'algoritmo Shortest Path Tree individua il cammino più breve tra la sorgente (radice dell'albero) e ognuno dei ricevitori (le foglie dell'albero)**
  - **i due più noti algoritmi utilizzati sono quello di Bellman-Ford e quello di Dijkstra**
- Steiner Tree
  - **L'albero di Steiner è quello che rende minimo il numero di collegamenti utilizzati per connettere i membri di un gruppo all'interno di un grafo**
  - **i cammini prescelti, ottimizzano la condivisione delle connessioni e NON la distanza dalla sorgente alle destinazioni**

Dijkstra's algorithm

- The basic operation of Dijkstra's algorithm is edge relaxation
  - **if there is an edge from u to v, then the shortest known path from s to u (d[u]) can be extended to a path from s to v by adding edge (u,v) at the end**
  - **This path will have length d[u]+w(u,v)**
  - **If this is less than the current d[v], we can replace the current value of d[v] with the new value**
- Edge relaxation is applied until all values d[v] represent the cost of the shortest path from s to v
- The algorithm is organized so that each edge (u,v) is relaxed only once, when d[u] has reached its final value

Dijkstra's algorithm (pseudo-code)

```
 1  function Dijkstra(G, w, s)
 2     for each vertex v in V[G]                        // Initializations
 3           d[v] := infinity
 4           previous[v] := undefined
 5     d[s] := 0                                        // Distance from s to s
 6     S := empty set
 7     Q := V[G]                                        // Set of all vertices
 8     while Q is not an empty set                      // The algorithm itself
 9           u := Extract_Min(Q)
10           S := S union {u}
11           for each edge (u,v) outgoing from u
12                 if d[u] + w(u,v) < d[v]              // Relax (u,v)
13                       d[v] := d[u] + w(u,v)
14                       previous[v] := u
```


Bellman-Ford algorithm

- This algorithm, like Dijkstra's algorithm uses the notion of edge relaxation but does not use with greedy method
- Again, it uses d[u] + d[u, v] as an upper bound of d[v]
- The algorithm progressively decreases an estimate d[v] on the weight of the shortest path from the source vertex s to each vertex v in V until it achieve the actual shortest-path
- The Bellman-Ford algorithm is remarkable in its simplicity, and it can be used with graphs in which some of the edge weights are negative

Bellman-Ford algorithm (pseudo-code)

```
function BellmanFord(list vertices, list edges, vertex source)
   // This implementation takes in a graph, represented as lists of vertices
   // and edges, and modifies the vertices so that their distance and
   // predecessor attributes store the shortest paths.

   // Step 1: Initialize graph
   for each vertex v in vertices:
       if v is source then v.distance := 0
       else v.distance := infinity
       v.predecessor := null

   // Step 2: relax edges repeatedly
   for i from 1 to size(vertices):
       for each edge uv in edges:
           u := uv.source
           v := uv.destination             // uv is the edge from u to v
           if v.distance > u.distance + uv.weight:
               v.distance := u.distance + uv.weight
               v.predecessor := u
```

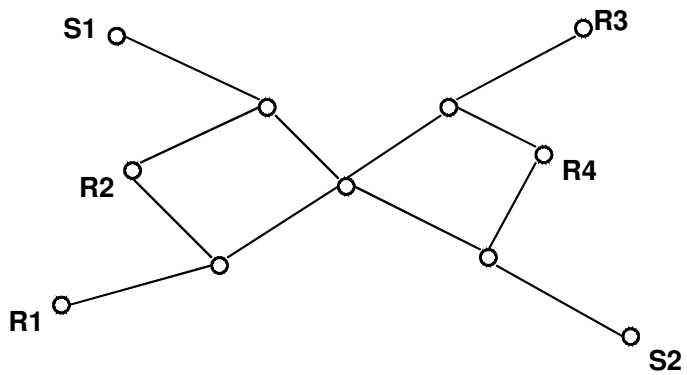

Multicast Routing Protocols: Overview

- Types of delivery trees
  - **per-source, per-group trees (Source-Based)**
  - **per-group trees, shared by all sources (Shared-Tree)**
- How delivery trees are built between senders and receivers
  - **On demand, in response to data arrival**
    - flood data, then prune (Dense Mode)
    - flood membership info and and build tree as data arrives (MOSPF)
  - **Explicit control**
    - send explicit joins and keep join state (Sparse Mode)
    - in case, use one of more “meeting places” (RP)
- Topology database (routing table) construction
  - **build own routing table**
  - **use unicast routing table**
- Intra or Inter-domain

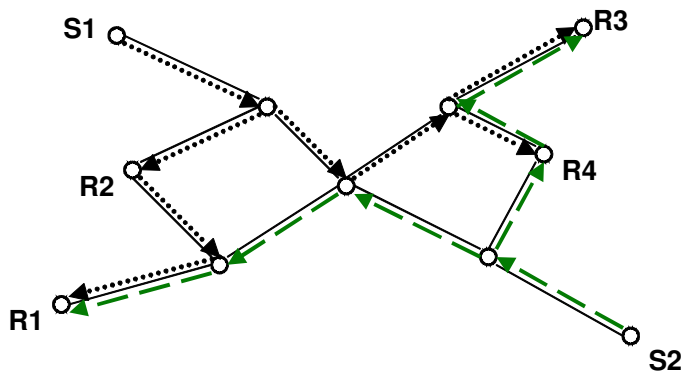
Source-Based Tree vs Shared-Tree

- SBT (Source-Based Tree)
  - **ha come obiettivo la costruzione di alberi di distribuzione da ogni sorgente verso l’insieme completo dei ricevitori di un gruppo multicast**
  - **tanti alberi quante sono le sorgenti di traffico multicast**
  - **La costruzione di ogni albero avviene seguendo la strada più breve tra sorgente e destinazioni del traffico**
  - **scarsa scalabilità**
  - **e.g. DVMRP e PIM-DM**
- Shared-Tree
  - **sviluppata con l’obiettivo di superare le limitazioni di SBT**
  - **prevede la costruzione di un unico albero di distribuzione multicast intorno a un router (o a più di uno) chiamato core o RP (Rendez-vous Point)**
  - **Per ogni gruppo multicast viene individuato un core router che opera come punto di raccolta dei flussi multicast e come radice dell’albero di distribuzione**
  - **migliore scalabilità**
  - **possibilità di costituire percorsi di instradamento non ottimi**
  - **e.g. CBT e PIM-SM**

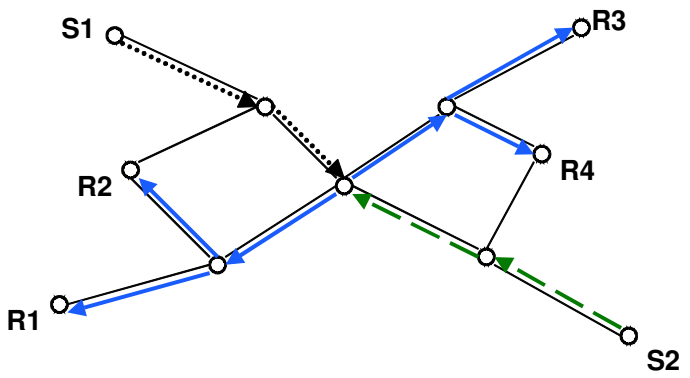
Topology to Illustrate Types of Delivery Trees



One Tree Per Source



One Tree Shared by All Sources



Dense Mode vs Sparse Mode

- Dense Mode
  - **impiegano strategie basate su inondazioni e potature periodiche**
  - **non sono consigliabili per reti di grosse dimensioni, a causa dell'elevato carico del traffico "di segnalazione" introdotto**
  - **sono adatti a contesti caratterizzati da un'alta concentrazione di utenti multicast**
  - **e.g. DVMRP, MOSPF & PIM-DM**
- Sparse Mode
  - **adesione esplicita**
  - **rendono così minimo il traffico "di segnalazione"**
  - **indicati per le reti di grosse dimensioni e più in generale per quei contesti in cui la densità di utenza multicast è bassa**
  - **e.g. CBT & PIM-SM**

Protocol Dependent vs Protocol Independent

- I protocolli di routing multicast utilizzano informazioni di routing unicast per costruire gli alberi di distribuzione del traffico multicast
- Due approcci:
  - **build own routing table (Protocol Dependent)**
    - protocolli di routing multicast che costruiscono le proprie tabelle di instradamento unicast, disaccoppiate da quelle realmente utilizzate per instradare il traffico unicast
    - e.g. DVMR
  - **use unicast routing table (Protocol Independent)**
    - protocolli di routing multicast che utilizzano le informazioni di instradamento unicast generate da altri protocolli (OSPF, RIP, routing statico)
    - e.g. PIM-DM e PIM-SM

Current IP Multicast Routing Protocols

- DVMRP — Distance-Vector Multicast Routing Protocol
  - **broadcast-and-prune,**
    unidirectional per-source trees,
    builds own routing table
- MOSPF — Multicast Extensions to Open Shortest-Path First Protocol
  - **broadcast membership,**
    unidirectional per-source trees,
    uses unicast routing table

Current IP Multicast Routing Protocols (cont.)

- PIM-DM — Protocol-Independent Multicast, Dense-Mode
  - **broadcast-and-prune,**
    unidirectional per-source trees,
    uses unicast routing table
- PIM-SM — Protocol-Independent Multicast, Sparse-Mode
  - **uses meeting places ("rendezvous points"),**
    unidirectional per-source or shared trees,
    uses unicast routing table
- CBT — Core-Based Trees
  - **uses meeting places ("cores"),**
    omnidirectional shared trees,
    uses unicast routing table

Multicast Routing Protocols: DVMRP

Distance-Vector Multicast Routing Protocol

- Dense mode protocol - Broadcast and prune
  - **Source trees created on demand**
    based on RPF rule
- Uses own copy of the routing table with a protocol similar to RIP
- Used with Mbone
- Many implementations
  - **mrouted, Bay, …**
  - **Cisco**

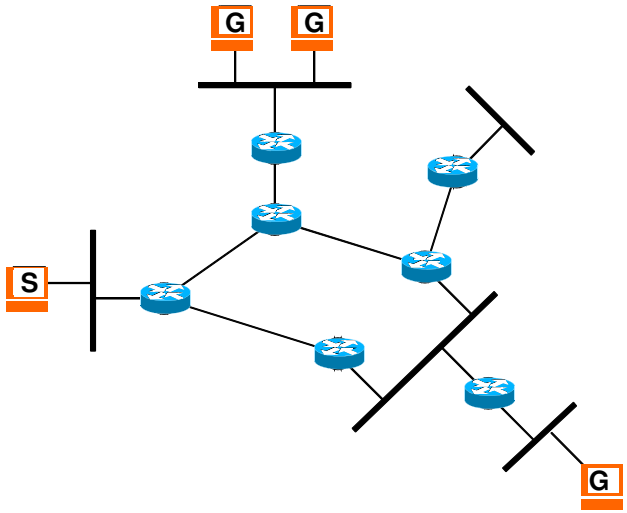
Distance-Vector Multicast Routing Protocol

- DVMRP consists of two major components:
  1) **a conventional distance-vector routing protocol (like RIP) which builds, in each router, a routing table like this:**

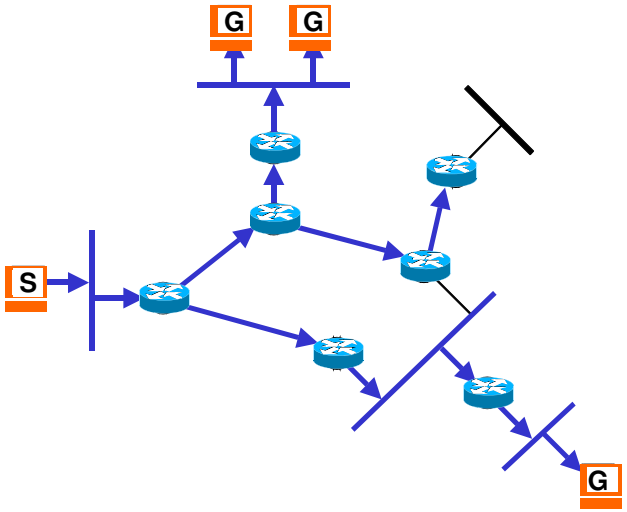
subnet (src)	shortest dist	via interface (ingress)
a	1	i1
b	5	i1
c	3	i2
…	…	…

  2) **a protocol for determining how to forward multicast packets, based on the routing table and routing messages of (1)**

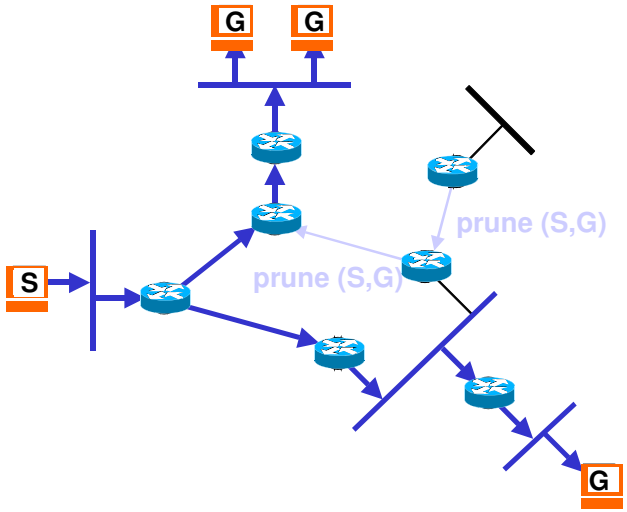
Example Topology



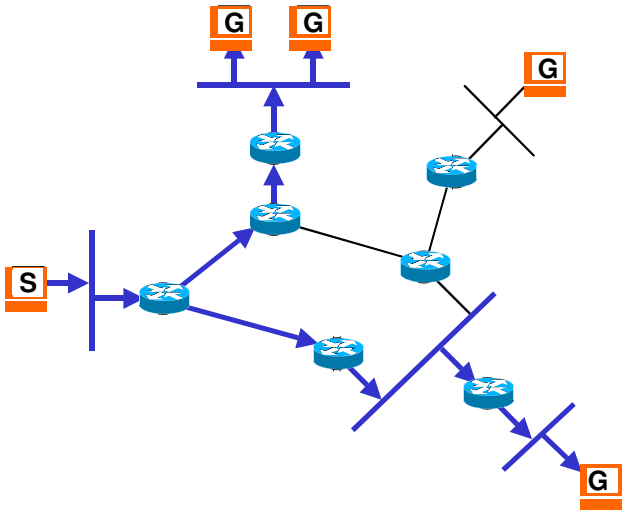
Phase 1: Truncated Broadcast



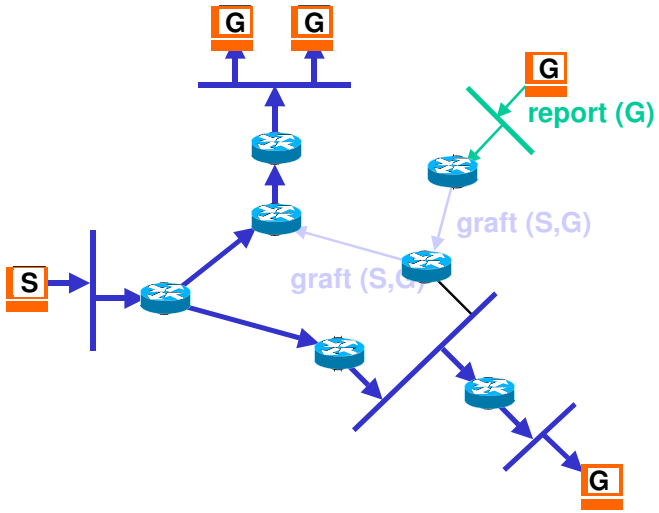
Phase 2: Pruning



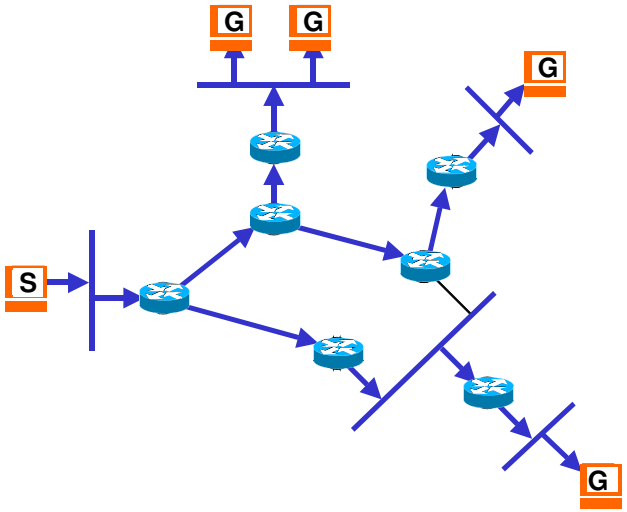
Steady State



Grafting on New Receivers



Steady State after Grafting



Multicast Routing Protocols: PIM

Protocol Independent Multicast (PIM)

- "Protocol Independent"
  - **does not perform its own routing information exchange**
  - **uses unicast routing table made by any of the existing unicast routing protocols**
- PIM-DM (Dense Mode) - similar to DVMRP, but:
  - **without the routing information exchange part**
  - **differs in some minor details**
- PIM-SM (Sparse Mode), or just PIM - instead of directly building per-source, shortest-path trees:
  - **initially builds a single (unidirectional) tree per group , shared by all senders to that group**
  - **once data is flowing, the shared tree can be converted to a per-source, shortest-path tree if needed**

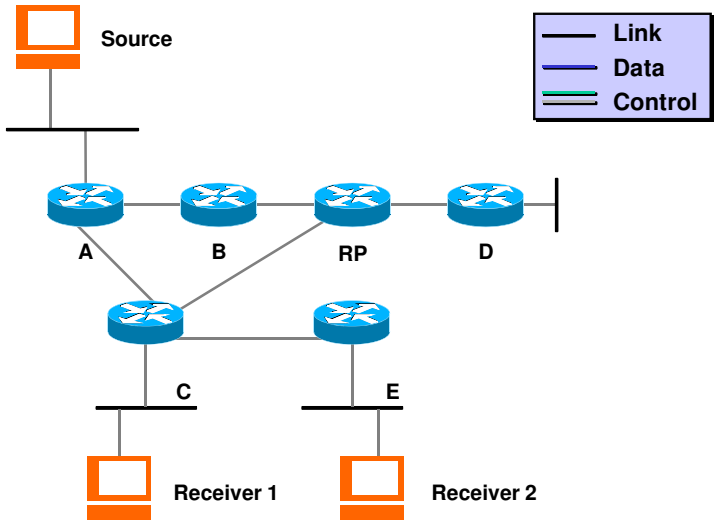
Dense Mode PIM

- Broadcast and prune "ideal" for dense groups
- Source trees created on demand based on RPF rule
- If the source goes inactive, the tree is torn down
- Fewer implementations than DVMRP
- RFC 3973 "Protocol Independent Multicast - Dense Mode (PIM-DM)", January 2005

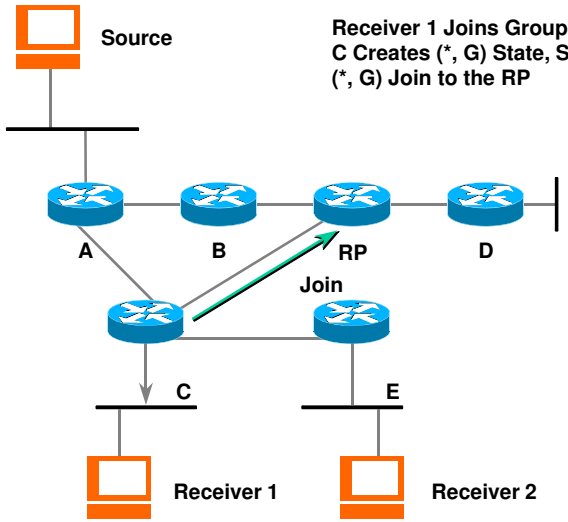
Sparse Mode PIM

- Only one RP is chosen for a particular group
- RP statically configured or dynamically learned (Auto-RP, PIM v2 candidate RP advertisements)
- Data forwarded based on the source state (S, G) if it exists, otherwise use the shared state (*, G)
- Draft: draft-ietf-pim-sm-v2-new-10.txt

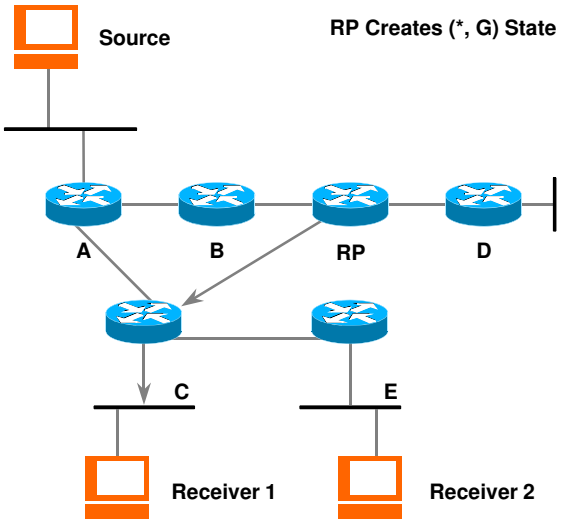
Sparse Mode PIM Example



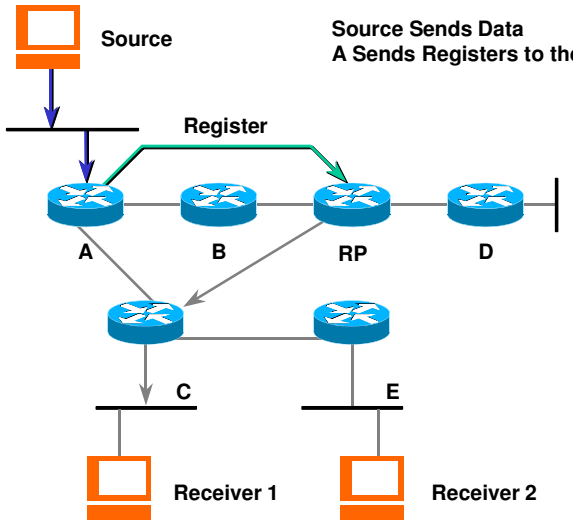
Sparse Mode PIM Example



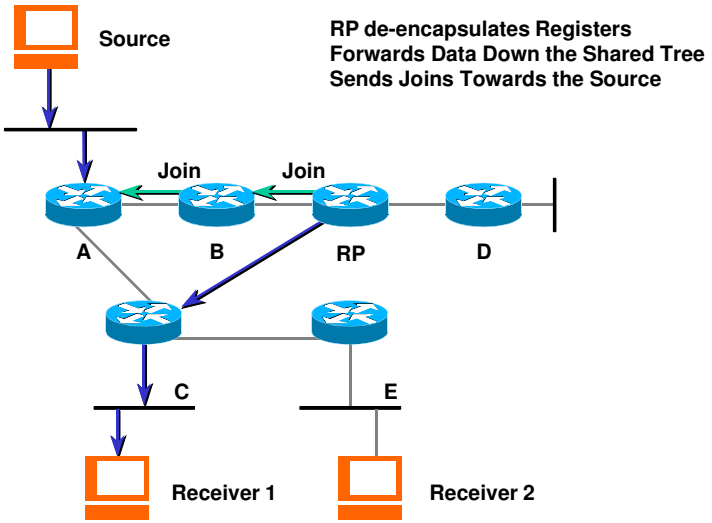
Sparse Mode PIM Example



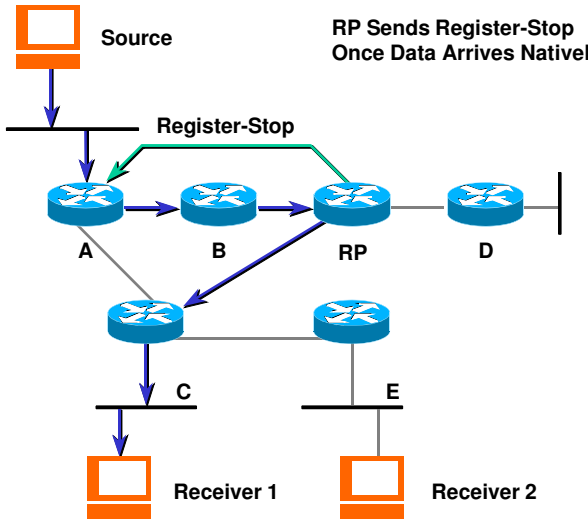
Sparse Mode PIM Example



Sparse Mode PIM Example

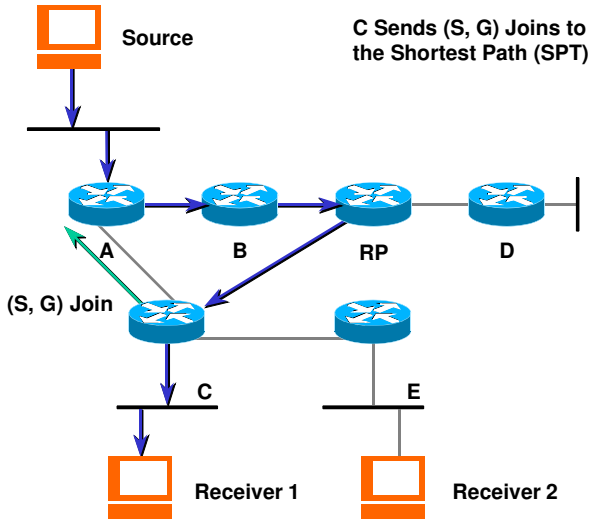


Sparse Mode PIM Example



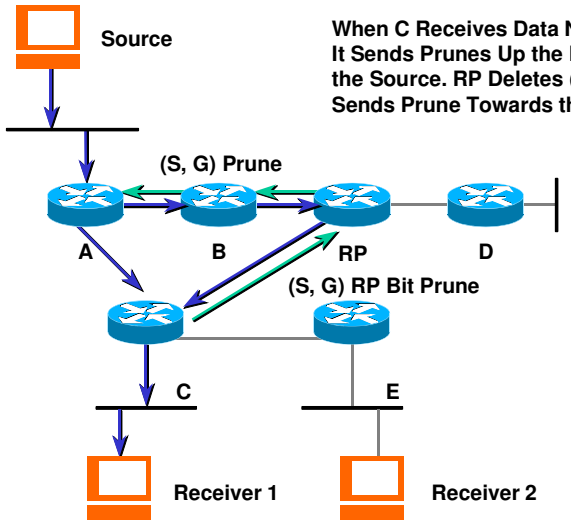
RP Sends Register-Stop
Once Data Arrives Natively

Sparse Mode PIM Example



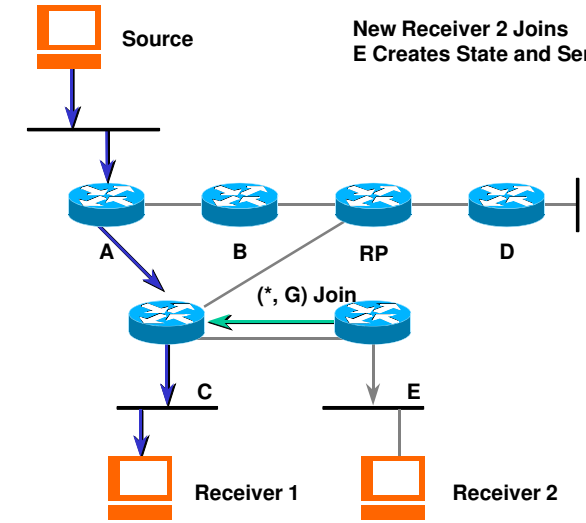
C Sends (S, G) Joins to Join
the Shortest Path (SPT) Tree

Sparse Mode PIM Example



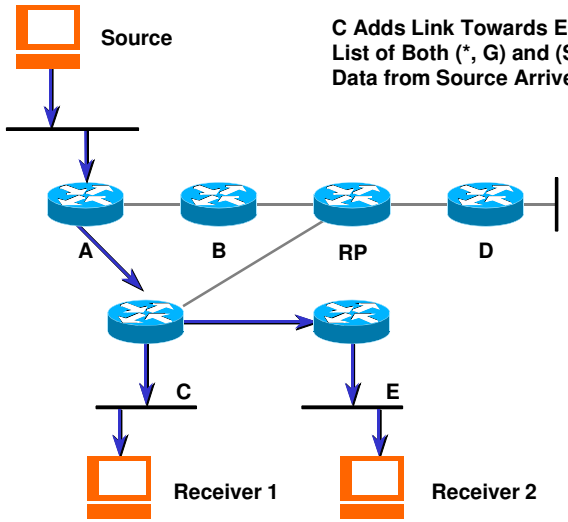
When C Receives Data Natively,
It Sends Prunes Up the RP tree for
the Source. RP Deletes (S, G) OIF and
Sends Prune Towards the Source

Sparse Mode PIM Example

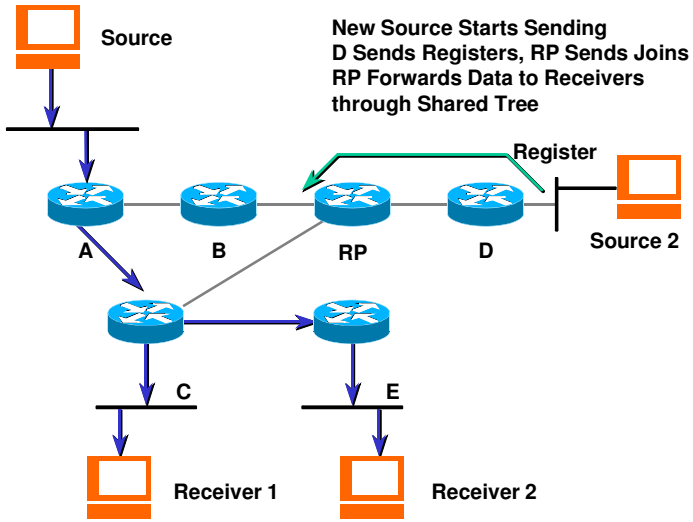


New Receiver 2 Joins
E Creates State and Sends (*, G) Join

Sparse Mode PIM Example

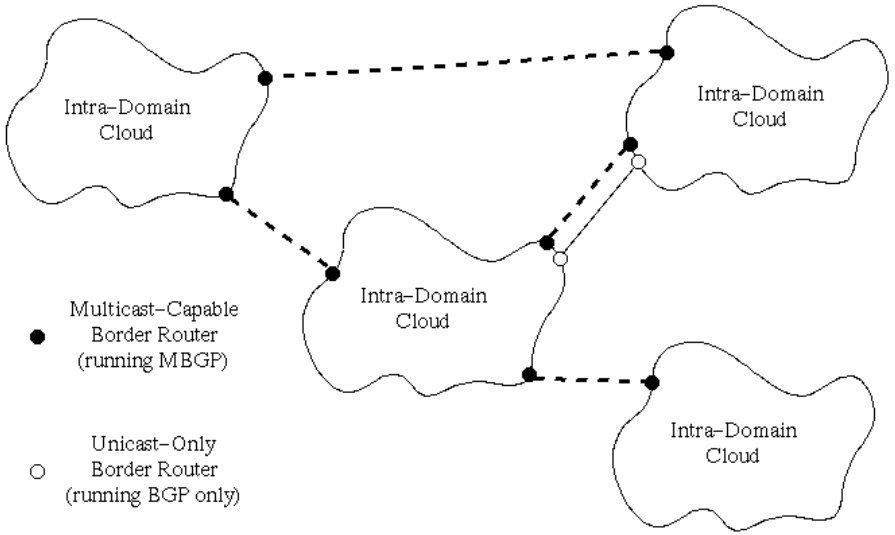


Sparse Mode PIM Example



Inter-Domain Route Exchange

- Exchange multicast reachability between Autonomous Systems (AS)
  - **Just like unicast routes are exchanged with BGP**
  - **Protocol is "Multiprotocol extensions to BGP" (RFC 2283)**
    - Also known as "Multicast" BGP (MBGP)
    - Also known as BGP4+
- MBGP is available and deployed today.
  - **Multiple vendors: Juniper, Cisco, Nortel, 3Com, IBM**
- Allows different unicast/multicast topologies



Issues

Deployment Obstacles—Technical Issues

- Source tree state will become a problem as IP multicast gains popularity
  - **When policy and access control per source are the rule rather than the exception**
  - **10,000 three member groups across the Internet**
- Hopefully we can upper bound the state in routers based on their switching capacity
- ISPs don't want to depend on competitor's RP
  - **Do we connect shared trees together?**
  - **Do we have a single shared tree across domains?**
  - **Do we use source trees only for inter-domain groups?**

- Unicast and multicast topologies may not be congruent across domains
  - **Due to physical/topological constraints**
  - **Due to policy constraints**
- Need inter-domain routing protocol that distinguishes unicast versus multicast policy

Deployment Obstacles—Non-Technical Issues

- How to bill for the service
  - **Is the service what runs on top of multicast?**
  - **Or is it the transport itself?**
  - **Do you bill based on sender or receiver, or both?**
- How to control access
  - **Should sources be rate-controlled?**
  - **Should receivers be rate-controlled?**
- Making your peers fan out instead of you (save replication in your network)
  - **Closest exit vs latest entrance — all a wash**
- Multicast-related security holes
  - **Network-wide denial of service attacks**
  - **Eaves-dropping simpler since receivers are unknown**