IP-Multicasting Technology

http://www.intelligraphics.com/articles/ipmulticasting1_article.html

Table of Contents

History and Overview

1. IP-Multicast: A Brief History and Typical Applications

The development of a practical implementation of *IP Multicasting* can be traced to one Stanford University graduate student, Steven Deering, who, in the late 1980's, was working on a network-distributed operating system called "Vsystem". His challenge was to provide some protocol mechanism that would allow multicasted data to flow between IP subnetworks. This goal, of course, required that the data-streams be able to move through IP routers (network Layer-3 devices). In addition, since Steve was working with Ethernet as his LAN media, he needed to address the issue of MAC (Layer-2) multicast addressing. The work eventually led to his doctorate paper on the subject ("*Multicast Routing in a Datagram Network*" - December, 1991) and, subsequently, the premier IP-Multicasting IETF document - RFC 1112.

Initially, Dr. Deering's IP-Multicast solution suite consisted of two protocols. The first, *IGMP* (Internet Group Message Protocol), allowed an individual host machine to "join" and "leave" multicast groups by responding to queries by a locally attached multicast-capable router. The second, *DVMRP* (Distance Vector Multicast Routing Protocol) was designed to allow cooperating multicast routers to share information on connecting user nodes to multicast sources. A multicast source might be an audio, multimedia video, or other similar server.

The primary advantages to the multicasting approach are in three areas: "bandwidth", "network congestion", and "server load". The first advantage should be quite obvious since the total number of packets transmitted is the same regardless of the number of end-users who are listening to a multicast data-stream. This enormous savings in bandwidth leads directly to the second advantage. With less bandwidth consumed there is a lesser chance of these new applications causing unnecessary congestion of network segments. The final advantage, however, may not be so obvious. For server-farms, the multicast approach requires a much smaller set of resources (CPU, process threads, network interfaces, etc.) to handle a given set of end-users. In unicasting, a separate session (UDP or TCP) must be started for each interested end-user. Thus, multicasting holds great promise in relieving the strain on audio / video / data-cast / push servers.

The IGMP and DVMRP protocols provided the basis of the first practical multicast application infrastructure, the *MBone* (Multicast Backbone), in 1992. The MBone was actually a number of "islands" of multicast-capable routers spread around the world. Since the intervening Internet routers were not multicast enabled, IP "tunneling" was used to unicast the traffic between the router islands. The MBone has grown from a few hundred users in 1992 to more than 7000 users in late 1998. Initially, the routers were in fact Sun Sparc UNIX boxes running a custom version of *routed* - UNIX's routing daemon process. Within a few years, Cisco added multicast support to their operating system (IOS). So, a mix of Sparc and Cisco routers currently comprise the MBone.

Though it provided a start to making IP-Multicast practical, DVMRP was not a very scaleable solution. As with the unicast-based RIPv2 (Routing Information Protocol), the total number of router hops was limited to 32. Obviously, a router mesh "width" of 32 is considerably too small for deploying DVMRP on the world-wide Internet. To move closer to the goal of Internet-based IP-Multicast, three other protocols were developed - *PIM-DM* (Protocol Independent Multicasting - Dense Mode), *PIM-SM* (Protocol Independent Multicasting - Sparse Mode), and *MOSPF* (Multicast Open Shortest Path First).

Interestingly enough, the multicast group leave/join protocol, IGMP, first proposed by Dr. Deering in his doctorate thesis (and RFC 1112) is still the only deployed mechanism for

determining which user nodes are interested in participating in which multicast groups (multicast sources).

With these five basic protocols, many applications have emerged that are well suited for the IP-Multicast paradigm. In the MBone network, the primary applications are SDR for session directories, *VAT* for audio-only transmissions, *VIC* for video streaming transmissions and the *WB* tool for white-board collaboration. Many enterprises use multicasting for audio/video meetings, push technologies (such as stock tickers), gaming and simulations. Many new IP-Multicast applications are in the works now that the infrastructure seems to be more stable and standardized.

This series of articles focuses on these five (5) IP-Multicast protocols and does not consider the work in specialized/advanced areas such as bi-directional CBTs (Core Based Trees).

This first article reviews IP-Multicast addressing specifics and the protocols themselves. Future articles deal with issues that arise when using switches with multicasting, describe the step-by-step mechanics of processing IP-Multicast packets (I.e. multicast packet data-plane processing), enumerate the timing (HELLO timers, aging, timeouts, etc.), sizing (RAM tables, CAM tables), and decoding of the protocols.

2. IP-Multicast: Addressing

The IP (Internet Protocol) addressing scheme is based on a 32-bit value that is typically stated as four 8-bit bytes ("octets") in dotted-decimal format, e.g.

```
192.16.151.8" = 0xC0019708
```

The value describes a specific IP node on the network. If, however, the first four bits of the address are set to "1110", the address no longer specifies a specific node. Instead, the address indicates an *IP multicast address*, an IP address that describes a subset of all network nodes with IP stacks. A specific node must configure itself to "listen" for a given multicast address or set of addresses.

Another IP address that has similarities to the multicast set is that of the *IP broadcast address:* "255.255.255.255". This address is a special case and describes the recipient of the packet as all IP nodes, i.e. the improper subset of IP nodes. IP broadcasts are typically frowned upon as the basis for a protocol since CPU cycles, though wasted on those machines that are not interested in a given packet's contents, are still required for packet content decoding and processing.

In the original class-based description of IP addressing, IP-multicast addresses are also referred to as "Class D" addresses (having the first four bits equal to "1110"). Class D provides a range of IP-Multicast addresses from "224.0.0.0" through "239.255.255.255". All Class A, B, and C network addresses are designated as "unicast", i.e. designed to specify a specific node on an IP network. (Class "E" addresses, with the first five bits as "11110" - 240.x.y.z through 255.x.y.z, is reserved for use by the IETF to test experimental protocols on the Internet).

Since IP-Multicast was implemented after the IETF's acceptance of such non-class-based IP addressing schemes as CIDR ("Classless Internet Domain Routing"), all IP-Multicast routing protocols provide subnet masks in their routing tables and are typically *not* referred to as Class D addresses. These non-class-based routing protocols are analogous to their unicast cousins - RIPv2, OSPF, and EIGRP.

2.1 Well-known IP-Multicast Address Ranges

As mentioned above, the range for IP multicast addresses is between "224.0.0.0" and "239.255.255.255". The Internet Assigned Number Authority (IANA) has reserved certain

portions of this address space for "well-known" purposes. The first range of reserved addresses is:

224.0.0.0 through 224.0.0.255 (locally-scoped reserved)

```
224.0.0.1 All Hosts
224.0.0.2 All Multicast Routers
224.0.0.3 Unassigned
224.0.0.4 DVMRP Routers
224.0.0.5 MOSPF Routers
224.0.0.6 MOSPF Designated Routers (DR)
224.0.0.7 ST Routers
224.0.0.8 ST Hosts
224.0.0.9 RIPv2 Routers
224.0.0.10 IGRP Routers
224.0.0.11 Mobile-Agents
224.0.0.12 DHCP Server/Relay Agent
224.0.0.13 All PIM Routers
224.0.0.14 RSVP-Encapsulation
224.0.0.15 All CBT Routers
--- and so on to 224.0.0.255 -
```

This first range is locally scoped, i.e. a router *will never* forward them. The following well-known address space has a wider scope and routers *will* forward them.

224.0.1.0 through 224.0.1.255 (globally-scoped reserved)

```
224.0.1.0 VMTP Managers Group
224.0.1.1 NTP (Network Time Protocol)
224.0.1.2 SGI-Dogfight
224.0.1.3 rwhod
224.0.1.6 NSS (Name Service Server)
224.0.1.21 DVMRP on MOSPF
224.0.1.39 Cisco-RP-Announce
224.0.1.40 Cisco-RP-Discovery
```

Finally, an "Administratively Scoped" multicast address range has been set by the IANA for use within the confines of an origination and should never be seen on the Internet:

239.0.0.0 through 239.25.255.255 (Administratively-Scoped)

These addresses can be thought of as analogous to the 10.0.0.0/8 range given for administratively-scoped Class A unicast addresses (RFC 1918).

2.2 Mapping IP-Multicast Addresses to Ethernet MAC Addresses

One of the difficulties that Dr. Deering discovered while trying to deploy a practical IP-Multicast protocol suite on an Ethernet LAN was in mapping the range of IP multicast addresses to an equivalent set of Ethernet MAC multicast addresses. The 48-bit Ethernet MAC addressing scheme uses the least significant bit (LSB) of the first octet to specify unicast or multicast/broadcast - a 0 for unicast and a 1 for multicast/broadcast. This addressing is equivalent to saying that an Ethernet MAC address is multicast if the first byte is odd and, in the special case where *all* 48-bits are 1, that you have a MAC broadcast. Dr. Deering had to implement a many-to-one mapping of IP multicast addresses (groups) to MAC multicast addresses as follows:

- the upper 24-bits of the MAC address are *always* **0x01005E** (the OUI - Organizationally Unique Identifier);
- the next bit of the MAC address is always zero;
- the remaining 23-bits of the MAC address are set equal to the last 23-bits of the IP multicast address;

Note that this process leaves a 32:1 IP-to-MAC mapping. Why? Well, an IP address is 32 bits (MSB="bit 0" and LSB="bit 31"). This amount minus the 4 required leading Class "D" bits leaves 28 bits. With only 23-bits being available in the MAC address, bits 4 through 8 of the

IP multicast are not used in the MAC address. This result allows 2**5 or 32 for the number of IP multicasts that map to the *same* MAC address.

Why Dr. Deering did not simply "sign up" for a 16 contiguous block of upper-order 24-bit multicast prefixes (OUI) that would make up for the lack of address space in the lower 24 bits is an interesting historical question. The answer was based purely on finances - Dr. Deering's manager at Stanford could only afford one OUI to be reserved for his researcher's work, since the IEEE was charging $1,000 per OUI at the time. Thus, a deficit of $15,000 caused this strange dilemma, which would haunt multicast developers for the next decade!

2.3 Using IP-Multicast with non-Ethernet Media

While the canonical format of the FDDI (Fiber Distributed Data Interface) MAC multicast addresses make for reversed-octets (giving 0x80007Axxxxxx instead of Ethernet's 0x01005Exxxxx) on the wire, all other aspects of the mapping work as described in section 2.2 above.

Token Ring (TR) uses a similar format to FDDI and can, in theory, work with IP multicasting as well. However, many TR NICs (Network Interface Cards) cannot interrupt on a *given* multicast MAC address, which will cause unnecessary interrupts since the node must decode/analyze *every* packet as a possible multicast that might have interesting data. This fact is one reason that Ethernet is the LAN-of-choice when it comes to IP-Multicast.

3. IP-Multicast: The Protocols

This section gives a very brief overview of the operational characteristics of each of the five primary protocols of today's IP-Multicasting networks. *Many* other ancillary and experimental protocols are involved but will only be mentioned in the following sections as they relate to "the big five". Of the five, the IGMP membership and the PIM-SM routing protocols have the greatest deployment at the beginning of the new century. Any new enterprise/Internet rollouts will be unlikely to use DVMRP or even PIM-DM. MOSPF and some enhanced versions of BGPv4 will be competing with PIM-SM for greatest deployment for some years ahead. (In recent conferences held by Cisco on multicast-based video conferencing, PIM-SM and its ever-evolving versions were highly touted.)

[Special Note: if a user node wishes to participate in the Internet's Mbone applications and transmissions, DVMRP is still the *only* multicast routing protocol available - at least for now]

3.1 IGMP - Internet Group Message Protocol

IGMP is a very simple leave/join protocol that allows end-user nodes and their multicast-enabled routers to exchange messages that describe the wishes of the hosts to participate in multicast groups. Version one of this protocol (part of the RFC-1112 1991 document) was based on two primary messages - a Membership (Group) Query sent by the router and a Membership (Group) Report sent by the end-user nodes. Version two of the specification, which expands on the two primary messages, can be found in the RFC-2236 1997 document. A third version (RFC not yet assigned) has not yet been ratified by the IETF and adds a few additional features. Even with the additions of version two and version three, the resultant protocol is remarkably similar to Dr. Deering's original specification.

In IGMPv1, only two messages (Membership Query and Report) are defined. A "designated" (sometimes called the "dominant") multicast-enabled router, referred to as the "DR", sends out periodic queries to all nodes on a given attached network using the 224.0.0.1 multicast address. Each node that is configured to receive these multicast queries returns a separate report to all multicast groups (addresses) in which it wishes to participate after a random timer countdown. If another node on the segment hears a report for a specific group in which it

participates *before* sending its own report, this other node cancels its own transmission of the report. The reason is simple - the DR *only* needs to know that one end-user would like to hear from that group. A common misconception is that a list of participant IP addresses exists in the DR. DVMRP, for example, *only* has a list of sources and groups (a *(S,G) table* in multicast terminology) for a given router interface (e.g. (194.15.1.1, 239.1.1.1), where 194.15.1.1 is the IP address of the node that is sourcing the multicast and 239.1.1.1 is the multicast address/group chosen by the source node). The DR places the newly discovered (S,G) entries in its table and continues. This procedure is how a "join" is accomplished. Note that these message packets cannot move beyond the locally attached routers due to the fact that the TTL (Time To Live) field is set to 1.

If it does not hear a report back for a multicast group that earlier had at least one participant, the DR then clears that (S,G) from its table on that interface. This procedure is how a "leave" is accomplished. No explicit "leave" message exists in version 1 of IGMP. This procedure can cause problems, of course, if a user "surfs" through a series of multicast group transmissions and settles on just one. The DR believes (for a few minutes) that all those groups must be presented on this LAN segment. If this occurrence happens more than a few times a minute, an incredible amount of traffic can be reproduced on the segment even though no end-user really wants to receive it!

Version two of IGMP provides the ability for an unsolicited "leave" message to be sent to the DR. So, the impact described above is essentially squelched. In addition, version one does not specify a defined way to elect a DR out of a possible number of qualifying multicast-enable routers; whereas version two uses the lowest IP address of the attached router interfaces to establish the DR. Version two also allows the DR to query only a specific multicast group instead of querying for all multicast participants in any group. Finally, version two provides a "maximum response time" field in the Query packet that is used to fine-tune the amount of random countdown performed by the end-user nodes in responding to Membership Query packets.

Version three, when deployed over the next year of so, will add a security feature ability to IGMP, allowing each end-user host to select which multicast *source* address it wishes to listen to for a given multicast group. This feature may seem confusing at first since one would think that only one IP source would exist for a given multicast group. However, if a malicious user sources a group on his/her machine simultaneously to a valid multicast group currently in use, that user could potentially create a DOS (Denial of Service) attack that could severely disrupt audio/video or data-cast applications (e.g. send erroneous stock market ticker information, take over the audio portion of an important announcement, etc.).

3.2 DVMRP - Distance Vector Multicasting Routing Protocol

While the IGMP protocol is clearly defined as the method by which end-user nodes are added to the DR's (S,G) table, the DRs must still find a path from the multicast sourcing node to these end-users. This area is where multicast routing protocols are important, and DVMRP, described in RFC 1075, is the preeminent multicast routing protocol.

DVMRP, loosely based on the analogous IP unicast inter-router protocol RIPv2 (Routing Information Protocol - version 2), uses "distance vector" techniques based on Bellman-Ford algorithms. Both DVMRP and RIPv2 protocols use the concept of next-best-hop and do not have a total picture of the router mesh inside each DR (only who its immediate router neighbors are). DVMRP has the same router mesh "width" as RIP, i.e. 32 hops maximum. This restriction obviously limits the deployment to small and medium sized enterprises. The Internet cannot ubiquitously use DVMRP for this reason. In addition, DVMRP uses only one metric - the number of hops, for best route determination.

Interestingly, DVMRP is similar to RIP version 2, not version 1. This similarity to the later version of RIP is because DVMRP provides support for *classless* IP addresses, i.e. those that use the subnet mask for all subnet calculations. This approach of sending the subnet masks along with network addresses is characteristic of RIP version 2 and can also be referred to as VLSM - Variable Length Subnet Masking - in some texts.

DVMRP uses a separate Multicast Routing Table (MRT) to store information regarding the routes back to a given multicast source. Note the use of "multicast source" rather than the destination, as would be the case in RIP. This behavior is due to the fact that IP-Multicast looks at the spanning tree as traversed backwards - from the end-users back to the source rather than source to end-users. Since a multicast group (address) addresses a group of nodes instead of a specific node, this approach is the only way that routing makes sense in the multicast world.

The multicast spanning tree is built through a series of *floods*, *prunes*, and *grafts*. A flood refers to the DVMRP insistence that all DVMRP multicast routers must transmit multicast packets to *all* outgoing interfaces. Of course, this insistence seems to be a bit of overkill since many of the DVMRP routers may have no end-user nodes that are interested in the multicast traffic (i.e. their (S,G) tables, discerned by IGMP, are empty). In this case, these routers send a "prune" message back "up the tree" (also called "upstream"), stating that they are not interested in the multicast traffic. This pruning effect is only transient, however. For after a couple of minutes, the pruned branches re-grow, giving a chance for the effected routers to either join back with the main tree (if users have requested multicast group traffic via IGMP) or just send another prune message.

Further, if it feels ready to re-enter the tree immediately, a multicast router can send a "graft" message upstream instead of waiting for this de-pruning process. Interestingly, current implementations of DVMRP maintain information on pruned branches but *do not* actually delete them from the internal database. With the high volume of prune/grow-back/graft operations in a typical multicast network, just toggling a state field in the entry saves router CPU cycles in exchange for the modest amount of additional memory required to track all branches.

The resultant spanning tree is referred to in a variety of ways in IP-Multicast documents, including as a "dense mode source distribution tree", a "shortest path tree" (SPT), or a "truncated broadcast tree". (The term "dense mode" refers to the fact the multicast traffic will penetrate much of the overall network and that this traffic flow is deliberate and desirable). Of the three terms, SPT is the most common designation. An SPT is in direct contrast to those protocols that use a "shared tree" that is based not on the multicast source's router, but rather on a designated "rendezvous point" router. PIM-SM uses this shared tree approach and will be discussed in greater detail in a later section.

To learn about its adjacent neighbors, a DVMRP router sends periodic "Hello" messages on all of its interfaces using the "All DVMRP router" multicast address - "224.0.0.4". Upon receiving a Hello message, the DVMRP router places its interface address in the "Neighbor List" field of the message. When it receives a "Hello" message and identifies its *own* interface address in this field, the router knows that a two-way adjacency has been successfully formed between itself and the neighbor that sent the message.

The DVMRP MRT typically has the following entries: source network, source network subnet mask, administrative distance, no-of-hops metric, uptime, expiration timer, next hop address and interface going towards the source, and information about the neighbor that sent the DVMRP route message. The entire multicast routing table is periodically transmitted to all DVMRP neighbors, helping to keep all DVMRP routers in synchronization with each other. As with RIPv2, convergence of a changed topology (down/up link-state change) within the

router mesh can take time. Entries in the table are periodically deleted using the expiration timer and are re-learned from subsequent neighbor route updates.

As with IGMP, the DVMRP protocol takes advantage of the TTL (Time To Live) field of the IP header to specify the extent of the packets in terms of router mesh width. Typical TTL values are:

TTL Value	Scope of DVMRP Packet
0	Restricted to the same host
1	Restricted to the same subnetwork
32	Restricted to the same site
64	Restricted to the same region
128	Restricted to the same continent
255	Unrestricted in scope

In addition to the previously mentioned MRT, the DVMRP-enabled router also creates a Multicast Forwarding Table (MFT). This forwarding table is simply a vector of (S,G) values with their associated incoming interface, outgoing interface(s) and "prune status". The MFT is used by the line-card to quickly make the correct outgoing interface decision based on the multicast source and group. Note that the prune status is provided so that traffic is not forwarded out to branches that have requested that they not be part of the active tree.

During normal operation of the multicast router, a check is made using the MFT to ensure that a multicast packet seen on a given interface corresponds to the route back to the source that owns the group. This check kills any packet that was received on some *other* port due to a non-convergent router mesh, typically because of a recent topology change. This check is referred to as a Reverse Path Forwarding (RPF) check.

If the packet passes the RPF check, it is forwarded out to all interfaces downstream. Note that "pruning" messages may have severely depopulated this interface list. This reduction is crucial for a dense mode protocol such as DVMRP since without pruning, multicasting might look more like IP broadcasting, at least up to the final-hop multicast routers!

As previously stated, DVMRP does *not* scale well and is very "chatty". In addition, DVMRP requires a great deal of router memory to maintain the separate multicast routing and forwarding tables. This protocol is, however, the easiest multicast routing protocol to understand and is viable for small-to-medium networks, especially if only LANs are involved and most end-users need/want to receive much of the transmitted multicast traffic.

3.3 PIM - Protocol Independent Multicasting

A number of years ago, the IDRM (Inter-Domain Multicast Routing) working group of the IETF began development of a multicast routing protocol that would operate regardless of the unicast routing protocol being used. Indeed, one of the goals of PIM is to use existing routing/topology tables and *not* create multicast specific ones. This approach is in contrast to DVMRP, discussed in the previous section, and its use of a Multicast Routing Table.

The primary goal of the PIM protocol is to provide superior sparse mode operation, however, PIM does have a dense mode model as well. The committee's decision to provide both modes allowed PIM to be used as a total solution to IP-Multicast and not depend on DVMRP or MOSPF as the dense mode protocol. These two modes, sparse and dense, operate rather differently and are discussed in more detail in the next two subsections.

3.3.1 PIM-DM - Dense Mode

While both DVMRP and PIM-DM are both dense mode multicast protocols with a SPT (Shortest Path Tree) model, PIM-DM uses a very different approach to the in-line egress processing of the multicast group packets. While DVMRP uses a MRT and MFT to calculate

which ports to transmit to for a given (S,G) combination, PIM-DM blindly transmits the multicast packet to *all* interfaces, as long as that interface has not been pruned. PIM-DM accepts this additional packet duplication in order to operate independently of the unicast routing tables and their resultant topology. In addition, no parent/child databases need be created using this very simplistic model. Needless to say, PIM-DM should only be used when a large percentage of the end-users require multicast traffic and very little of the users require WAN links to reach the sources (i.e. bandwidth is plentiful).

3.3.2 PIM-SM - Sparse Mode

When one refers to the PIM protocol for IP-Multicasting, one is usually referring to the sparse mode of operation. PIM-SM is one of the few IP-Multicast approaches that provides a more efficient mechanism for multicasting when only a small percentage of end-users need to listen to the group traffic and/or when WAN links are used to access the multicast sources. PIM-SM uses RPT (Rendezvous Point Trees) as its primary spanning tree, making use of single point of "rendezvous" (the "RP") between sources and recipients. The RP is a network manager-specified, multicast-enabled router that is usually quite close to the multicast sources. Because end-users are downstream from the RP based distribution tree, the designation for a particular multicast group is (*,G), i.e. *all* multicast groups are sourced from the same point - the RP. Thus, the existence of multiple RPs is possible, each being responsible for some subset of all required multicast groups.

Some difficulties must be overcome for this shared tree approach. Since the RPT is *unidirectional* and can only flow *from* the RP to the end-user, how do the RP and other downstream routers discover the additions of new group users? (Remember, the end-users "sign up" for group access using the standard IGMP protocol discussed previously). Also, how does a given last-hop router initially know the IP address of the RP?

The method of discovery of new group users involves a standard SPT (Shortest Path Tree) from the last-hop router back to the RP. Standard unicast routing tables are used and a "PIM Shared Tree Join" is performed. The only necessity is for each router along the way, all the way up to the RP itself, to add the (*,G) entry for the required multicast group. In this way, the end-user has "joined" the RPT for subsequent multicast packets for this group. When a given last-hop router discovers (via IGMP) that there are no more end-users for a given multicast group, a "Shared Tree Prune" message can be sent up the SPT towards the RP so that timeouts are not needed to prune the proper branches.

Several possible methods exist for a given last-hop router to initially know the IP address of the RP. The most straightforward approach would be to statically configure the RP's IP address into each router that might participate in multicasting. While simple, this method certainly does not scale well! Cisco proposed a more automatic method, based on a protocol that they call "Auto-RP" which uses a "Cisco-RP-Discovery" multicast packet with well-known address "224.0.1.40" and a "Cisco-RP-Announce" which uses "224.0.1.39". Version two of PIM-SM offers another option, outlining a *bootstrap* process to discover the RP's address. Whatever method is used, the multicast routers must know the RP's IP address since standard unicast routing is used to implement a group join operation. (Remember that PIM does not maintain its own routing table and depends on the unicast routing tables already existing in the router.)

One feature of PIM-SM that is not available in other sparse mode protocols (such as bi-directional Core-Based Trees, CBTs) is the ability for a given last-hop router to ask for a direct SPT back to a given multicast source *without* requiring the source to link to the shared RP tree. This feature allows a given sourcing node the option of providing service directly to a set of end-users without routing its multicast payloads through the RP.

How does a given multicast source provide its packet traffic to the RP? PIM-SM uses another SPT from the RP back to the source to solve this concern. The RP knows that the source exists due to a unicast packet that is sent from the source directly to the RP's IP address using a special PIM message called a "Source Registration". Once the unicast packet is received, the RP can now make the reverse connection back to the sourcing node.

3.4 MOSPF - Multicast Open Shortest Path First

Just as DVMRP was based on the RIPv2 distance-vector-based unicast routing protocol, the MOSPF (RFC 1584) multicast protocol is an extension of OSPF, a "link-state" unicast routing protocol. While distance-vector algorithms use simplistic best-route metrics (RIP uses just one - the number of hops) and know only about the directly attached neighbors, a "link-state" routing protocol uses a rich variety of metrics as well as a full topological model of the network within each OSPF-enabled router. OSPF routers require that a great deal more memory be at their disposal to create these topological maps. After a topology change, the map is re-calculated using a "Dijkstra" algorithm to provide the shortest path between any two given networks/subnetworks (hence the protocol's name).

OSPF (the latest version is v2 and is described in RFC 2328) uses a hierarchical representation of the router mesh and divides all routers and their associated network segments into "areas". Area "0" is special and typically referred to as the "backbone area". All other areas must tie at least one of their area routers into Area 0 (called Area Border Routers or ABRs). Thus, traffic that must route from one area to another must traverse Area 0.

OSPF, and thus MOSPF, uses a flooding technique to advise other OSPF routers of topology information. Unlike the RIP approach of sending the entire routing table in each flood, OSPF sends only the "delta" information, i.e. what has changed in the topology. This approach provides a significant bandwidth savings in those router-rich networks. The inter-router OSPF messages are called LSAs (*Link-State Advertisements*). In fact, the primary difference between OSPF and MOSPF was the addition of a new LSA that provides multicast group-specific information to the router mesh. In this way, MOSPF routers learn which ports of the router have interested listeners for a given multicast group. A Designated Router (DR) must be established and SPT (Shortest Path Trees) are used, just as with DVMRP, but the tree routing is performed with the Dijkstra algorithm, not with DVMRP's least-hops approach.

In order to reduce the computational effort required for the MOSPF router to figure all SPTs for all possible (S,G) combinations, the algorithm is only run after a multicast packet for a given group is seen in real-time. Then, and only then, does the MOSPF router place the resultant forwarding information into its own Multicast Forwarding Table (MFT). Even with this streamlining technique, a MOSPF router mesh could easily become overwhelmed computationally if a large number of sources and their users came joined at the same (or nearly the same) time. As a result, MOSPF is currently not a very viable IP-Multicast solution (although some MOSPF deployments do exist in corporate networks).

3.5 mroutes - Static Multicast Routing

Most vendors implement, in addition to some combination of the above IP-Multicast routing protocols, a straightforward way of adding routing and forwarding information statically, i.e. by using the router's management console/software to directly enter which incoming interfaces point back to multicast sources and which outgoing interfaces are to be transmitted to for a given multicast group. Although somewhat "brute force" and requiring a great deal of personnel time to maintain, this method is warranted for certain special scenarios - most of them dealing with security, e.g. firewalls to the Internet, secure areas of the enterprise network, anti-hack and anti-spoofing policies, and others. Cisco refers to its manual (static) multicast routing technique as *mroutes*.

Conclusion

While IP-multicasting has evolved since Dr. Deering first conceived and implemented it, the technology remains grounded in his initial work. This article has covered many of the basics of IP-multicasting and its affiliated protocols. The next articles in this series will explore the use of switches and will reveal additional technical details about the operation and use of IP-multicasting.

IP-Multicasting Technology Part 2: Switches vs. Routers

1. IP-Multicast: Using Switches with IP-Multicast

Previously in our review of IP-Multicast history and technology, we concentrated on OSI Layer-3 devices, i.e. multicast-enabled routers. However, due to the substantial deployment of high-speed Layer-2 Ethernet switches, developers must also be aware of the issues involved with the use of switches in multicasting.

1.1 Making Switches Multicast-Aware: IGMP Snooping, CGMP, and GARP

With the prevalence of Layer-2 Ethernet switches, problems concerning the use of such switches for directing multicast traffic have come to light. Many of these problems have been addressed by a variety of clever techniques, including *IGMP Snooping, Cisco's Group Management Protocol (CGMP), and IEEE's Group Address Resolution Protocol (GARP).*

One of the primary difficulties encountered in multicasting over Layer-2 switches (or bridges) is that the normal response to multicast traffic is the immediate flooding of the packets to every interface on the switch (other than the interface from which the packet arrived). This response is typical of a switch when either a multicast or broadcast MAC address is identified in the destination field of an Ethernet packet. If, somehow, the switch could determine which interfaces should be used for egress processing, the amount of unwanted multicast traffic could be drastically reduced. Thus, our question becomes: how can a switch determine the appropriate output interfaces for a particular multicast? To do so, the switch must access the type of information that is available within the IGMP protocol. However, with only the Layer-2 header available (MAC addresses and protocol types), the switch cannot differentiate between IGMP traffic (to and from the Designated Router) and normal multicast traffic. The following approaches were developed to resolve this issue.

IGMP Snooping

The first technique for resolving this dilemma is simple to implement but may not scale well, particularly in lower-end switch models. Referred to as "IGMP Snooping", this approach requires that the switch decode the IP header (Layer-3 information) by examining IP "protocol" field in order to separate out IGMP messages from normal multicast traffic. However, without some type of hardware (ASIC) assist, this additional decode process can be quite taxing on a central CPU-based switch. In fact, IGMP snooping may cause such switches to arbitrarily discard a large number of packets during times of multicast peaks. However, with the proper hardware assist, IGMP snooping is a viable solution. IGMP snooping is available on a variety of mid-to-high end switches.

Cisco's Group Management Protocol (CGMP)

The second approach, CGMP, is proprietary to Cisco and involves a router-to-switch multicast-group information exchange protocol. A mid-to-high end Cisco switch (Catalyst) can receive multicast-group join/leave messages from a multicast-enabled Cisco router (6000, 7000, etc.). These join/leave updates are then used by the switching logic to provide better multicast filtering through the switch fabric. CGMP provides such messages as "Add Port to Group", "Delete Port from Group", "Assign Router Port", "De-assign Router Port", "Delete Group", and "Delete All Groups".

Group Address Resolution Protocol (GARP)

A third approach is the IEEE's GARP protocol, whose primary purpose is to maintain VLAN group information. GARP can be extended to also provide (S,G) lists that allow the switch to map multicast groups to egress ports in much the same way that a VLAN is a list of MAC addresses that belong to a specific broadcast domain.

Regardless of which of these three methods is used, the primary CAM table to be updated with the multicast group information (mapped to a multicast MAC address) is called the "*Switch Forwarding Table*" or SFT.

While all of the above methods have been widely deployed, the most common is CGMP particularly due to the size of the installed base of Cisco multicast-enabled equipment. IGMP Snooping is the next most popular approach. IEEE GARP is a distant third and has much work left to be done in order to provide a viable solution.

1.2 Dealing with IP-Multicast Traffic Overflow

The increasing acceptance of IP-Multicast as a viable protocol suite with a whole spectrum of applications within the enterprise intranet has raised bandwidth concerns. In particular, corporate network managers possess the very real concern that multicast popularity might lead to a dramatic downturn in available bandwidth on a given network segment. This bandwidth concern would be especially problematic in the switched fabric due to the switch's inherent "flooding" behavior with multicast packet flows. An early technique presented to control this flood of multicast over switched networks was that of "multicast metering" or "rate-limiting". Metering would control the data flows so that a given level of multicast packets (packets/sec or percentage of theoretical maximum bandwidth) was never exceeded for a given network segment, i.e. the switch would throw away packets to satisfy the specified rate.

This approach proved to be a poor solution since many standard unicast protocols use some multicasting to get their messages around (e.g. OSPF HELLO routing updates and Spanning Tree Bridge Protocol Data Units (BPDUs)). The loss of these crucial control messages could easily meltdown a network during topology changes! Most network managers now consider this form of indiscriminant rate-limiting to generally be a bad idea.

However, such multicast metering might be useful if the packet could be classified as vital (BPDU, OSPF HELLO, etc.) vs. non-critical (audio multicast frames) with some sort of priority tagging scheme. Just such a scheme is discussed in the IEEE 802.1p and IEEE 802.1Q standards as well as other, vendor-proprietary, solutions. If the switch could handle this classification in a relatively low cycle-demanding manner, then multicast overflow could be dealt with efficiently.

2. IP-Multicast: To Switch or Route?

The initial requirements for dealing with multicast traffic within a switch/router device seems to suggest that a VLAN switching algorithm could be used to differentiate the traffic. After all, a multicast packet riding on the Ethernet frame format uses a MAC multicast address. However, the mapping between IP multicast addresses and a given MAC address is 32-to-1. Thus, a strictly Layer-2 interface lookup algorithm would not be able to differentiate between an entire block of 32 multicast addresses that may be in use (i.e. that may have a multicast source/group assigned to them). For example, the octet 224.1.1.1 has the exact same MAC multicast ("01:00:5E:01:01:01") as 225.1.1.1 and 226.1.1.1. Remember that the compression takes place at bits 4 through 8 of the 32-bit IP multicast address.

Apparently, multicast IP addresses (groups) that do not collide when translated into their equivalent multicast MAC addresses should thus be used. This address selection is precisely

what the industry accomplishes when multicast addresses are assigned to multicast sources. This "solves" our first obstacle to switching (instead of routing) a multicast packet.

Unfortunately, however, whether a given multicast packet is actual user-data (video, audio, data-cast, push, etc.) or just part of the control and management of a multicast protocol is impossible to determine. A case in point is IGMP, which transmits a "Membership Report" packet that uses the multicast group it wishes to join as the packet's IP destination address and, thus, the same MAC address as data that will be later sent on that same group.

Using one of the techniques discussed above (IGMP snooping, CGMP, and GARP), one can build a MFT ("Multicast Forwarding Table") for packet-egress decision logic, i.e. to which interfaces should a multicast packet be forwarded. With this table, multicasters would be able to more closely adhere to the adage "*switch when you can, route when you must*". As it turns out, this MFT would be essentially equivalent to the SFT ("Switch Forwarding Table") mentioned above.

By using traditional routing techniques for all multicast traffic, the problem is greatly simplified since a variety of IP-Multicast routing protocols exist (four of them discussed in our previous article). These routing protocols will build interface-forwarding tables based on the packet's multicast group (destination IP address) and, possibly, the multicast source. Remember that with the PIM-SM and CBT protocols, the source is typically the RP (a network manager-specified, multicast-enabled router that is usually quite close to the multicast sources), which is a well-known address to the router.

Thus the forwarding of a multicast packet is highly dependent on the source of the interface-forwarding information - via a Layer-2 method such as IGMP Snooping or CGMP, or a Layer-3 method such as DVMRP or PIM. However, a generic MFT can be constructed that uses either the Ethernet MAC Destination Address (48-bit) or the IP Destination Address (32-bit) as its key, depending on a configured option - the *Multicast-Router-Flag* (MRF). In this way, we use the packet's IP-DA if the MRF is set or the packet's MAC-DA if the MRF is cleared to find our egress interfaces. The MFT would be typically implemented as a hardware CAM as is the case with the VLAN switching tables.

To summarize our options for using multicast-forwarding in the MFT:

Route (Layer-3 decision): Uses the IP-DA for a key to the MFT. The MFT is maintained by the forwarding tables established by DVMRP, PIM-SM or MOSPF. (PIM-DM is a special case that floods the packet to all interfaces, unless an interface has its "prune status" set).

Switch (Layer-2 decision): Uses the MAC-DA for a key to the MFT. The MFT is maintained by the information collected from IGMP Snooping, CGMP router-to-switch protocol, or via the GARP protocol.

The most common setting, of course, is to clear the MRF (=0) and use the switch (Layer-2) lookup to implement fast-switching of IP-Multicast. This setting assumes that some mechanism for mapping of multicast grouping is active (e.g. IGMP Snooping, CGMP, or some future, workable version of GARP). Of course, if no Layer-2 protocol is available, the MRF must be set on (=1) in order to use the available IP-Multicast routing protocols.

Importantly, the TTL (Time To Live) field in the IP header of all IGMP join/leave control packets will *always* be set to one (1). This setting means that IGMP will not be accidentally transmitted to other egress interfaces, since the packet will be discarded after it is processed since its TTL will be decremented to zero.

Conclusion

As IP multicasting continues to grow in popularity, developers and network managers must be aware of the options available for (and challenges associated with) routed and switched IP

multicast packet transmission. In our next and final article on IP multicasting, we will explore timing and sizing the protocols as well as decoding the protocols.

IP-Multicasting Technology Part 3: Protocol Timing, Sizing and Decoding

1. IP-Multicast: Timing and Sizing the Protocols

The various IP-Multicast protocols offer different levels of throughput and reliability. For effective IP multicasting, the timing and sizing requirements for the IP Multicasting protocols must be considered. The different timing values for each protocol are a key part of differentiating the performance requirements for your IP-Multicast applications. Values such as timer and timeout values, as well as table structure and sizing, are used to accurately calculate the timing and sizing requirements for each protocol. While the decision criteria for the selection of a particular protocol is more complex than a simple tradeoff of speed versus reliability, no single multicast protocol meets the needs of all multicast requirements.

As discussed in previous articles, the most popular IP-Multicast protocol for an enterprise is clearly PIM-SM (Protocol Independent Multicasting - Sparse Mode), with DVMRP (Distance Vector Multicast Routing Protocol) a distant second. The Mbone (Multicast Backbone) still uses a great deal of DVMRP; however, the Mbone is primarily a core Internet infrastructure and not one that most enterprises deal with internally. PIM-DM (Protocol Independent Multicasting - Dense Mode) would be third statistically, with MOSPF (Multicast Open Shortest Path First) being utilized the least. An exterior Internet routing protocol, BGPv4, has recently been extended to provide for inter-AS multicast routing and has been designated "MBGP" (Multicast BGP).

In polling a representative set of enterprises, research suggests that the *minimum* required IP-Multicast group active population, e.g. (S,G) count in DVMRP and (*,G) count in PIM-SM, is approximately 256, with the typical population being closer to 2,000 for many organizations. The maximum observed in the subset polled was close to 10,000 active groups on a corporate network. Military and defense contractors would most likely have requirements for an even higher active group count.

Ascertaining the total maximum active multicast groups supported for a given vendor's router/switch product line is not a simple task since many protocols, unicast and multicast, must share available RAM and CAM space. Clearly, though, the high-end Cisco and Nortel Networks routers can easily manage tens of thousands of simultaneous multicast groups. This fact is another reason that MOSPF (described in previous articles) is not highly favored as the IP-Multicast protocol of choice for large installations, since the operating overhead required to maintain the unicast and multicast topologies becomes increasingly excessive.

The data below details timing information for various IP Multicasting protocols:

IGMP (Internet Group Management Protocol) Version 1	
Membership Query	Every 60 seconds ("Query Interval")
Membership Report	0-10 second random countdown
Leave Latency	(3 * Query Interval) = 180 seconds
IGMP (Internet Group Management Protocol) Version 2	
Membership Query	Every 125 seconds ("Query Interval")
Membership Report	Random countdown based on value specified in Membership Query (.1 increments) with default equal to 100 (10 seconds - as with V1)
Querier Election Timeout	(2 * Query Interval) = 250 seconds
IGMP (Internet Group Management Protocol) Version 3	
Same as for Version 2	Same as for Version 2
DVMRP (Distance Vector Multicast Routing Protocol)	

Neighbor Discovery Hello	Every 30 seconds
Neighbor Adjacency Timeout	(3 * Neighbor Discovery Msg.) = 90 seconds (Nortel uses = 140 seconds)
Multicast Routing Table Update	Every 60 seconds (similar to RIP)
Route Expiration Timer	200 seconds
Prune Reset	Every 120 seconds
DVMRP Routing Table	Source Subnetwork & Subnet Mask, Incoming Interface, Outgoing Interface(s), Metric (Hop Count), TTL, and Status
DVMRP Forwarding Table	(S,G), TTL, Incoming Interface, Outgoing Interface(s), Prune Status
PIM-DM (Protocol Independent Multicasting - Dense Mode)	
PIM Hello Message	30 seconds
Neighbor Adjacency Timeout	(3.5 * PIM Hello Msg.) = 105 seconds
PIM Neighbor Table Entry	Neighbor Address, Interface, Uptime, Expiration Timer, Mode (Dense, Sparse), Designated Router ("DR") Flag
Prune Reset	Every 180 seconds
Prune Delay Timer	3 seconds
PIM-SM (Protocol Independent Multicasting - Sparse Mode)	
PIM-SM Forwarding State	Entries deleted every 180 seconds
(*, G) Join Refresh Messages	Sent upstream every 60 seconds
MOSPF (Multicast Open Shortest Path First)	
Router LSA Interval	Transmitted every 30 minutes if no topology change before then to provoke a Router LSA.
Multicast-Group LSA Interval	On-demand only. MOSPF has very few timers.

2. IP-Multicast: Decoding the Protocols

Our final area of discussion is decoding the various protocols. This section provides a relatively complete decode analysis of the Ethernet Frame Format (Layer-2, Data Link Header), PIM-SM, PIM-DM, and IGMP (the leave/join protocol common to all IP-Multicast routing protocols). While this data may not be relevant from an application perspective, it is valuable for debugging purposes.

(Note: All values in this section that begin with the prefix "0x" are hexadecimal values. Values without the "0x" prefix are decimal.)

2.1 Ethernet Decodes (Layer-2, Data Link Header)

There are four types of Ethernet (CSMA/CD) frame types:

- V2 (Version 2.0) (the original "Ethernet" frame type)
- 802.2/LLC
- 802.3/SNAP
- Novell Raw

To ascertain which of the Ethernet types is present in a given frame, developers may use the decode logic presented in this section.

2.1.1 Address Fields Are Common To All Ethernet Formats

Octets 0 - 5: The *MAC Destination Address* (48-bit, 6-octets)

```
if field = '0xFFFFFFFFFFFF',
  then frame is a MAC broadcast

if field's LSB of first octet is '1' and the address is not broadcast,
  then frame is a MAC multicast
else the frame is a MAC unicast
```

Octets 6 - 11: The *MAC Source Address* (48-bit, 6-octets)

All MAC Source addresses must be unicast (e.g. from one source)

2.1.2 Decode Length/Type Field

Octets 12 - 13: The *Length or Type* Field (16-bit, 2-octets)

if field <= 0x5DC

then field is a *"Length"* and frame is a either an 802.2/LLC, 802.2/SNAP, or Novell Raw frame. The length is the total octets contained in Layers 3 thru 7 payload and *does not* include the Layer-2 CRC (4 bytes) at the end of the frame. -> goto 2.1.3;

else field is a *"Type"* and frame is a V2 (Version 2.0) frame. The type is really a protocol number that specifies what Layer-3 protocol header will be found immediately following the type. Another name for the type in a V2 frame is "EtherType". The EtherTypes are maintained by the IANA standards organization and are well-known values. -> goto 2.1.5;

2.1.3 Decode 802.2/LLC, 802.3/SNAP and Novell Raw Frame Types

Octets 14 - 15: DSAP/SSAP or Novell Field (16-bit, 2-octets)

if field = 0xFFFF
then frame is a Novell "Raw". The 0xFFFF is followed by an IPX Layer-3 routing header, i.e. this frame is for use only by the Novell IPX/SPX protocol stack. It *cannot* hold TCP/IP data. ->goto Novell Processing;

if field = 0xAAAA
then frame is 802.3/SNAP. Protocol information is follows. -> goto 2.1.4;

else frame is 802.2/LLC (Logical Link Control). The field's first octet is called the *Destination Service Access Point (DSAP)*, while the field's second octet is called the *Source Service Access Point (SSAP)*. These two values refer to the protocol type of the upcoming Layer-3 header. In addition, a 1 octet (8-bit) *Control Byte* follows the DSAP and SSAP fields (Octet 16). The value of the Control is set to 0x03 on Ethernet segments. -> goto 2.1.5;

2.1.4 Decode 802.3/SNAP Frame Type

Octets 16 - 21: Control Byte, OUI, and Type fields

The 802.3/SNAP frame has a 0xAAAA value for the *DSAP/SSAP* field and a 0x03 value for its *Control Byte* (Octet 16). Following the control byte, the *OUI (Organizationally Unique Identifier)* field is found (Octets: 17 thru 19). The OUI specifies which vendor's protocol stack will be used, e.g. Digital's DECnet, IBM's SNA, DoD's TCP/IP, Apple's AppleTalk, and so forth. The last two octets at 20 and 21 are the *Protocol Type*, specify which protocol will be used in the Layer-3 header of the frame. -> goto 2.1.5;

2.1.5 Layer-3 and Subsequent Layer Headers and Application Payload

The remaining data represents a series of protocol headers (also called *Protocol Data Units or PDUs*), starting with the Network Layer (Layer-3) and followed by Layers 4 through 7 (L4 = Transport, L5=Session, L6=Presentation, L7=Application). A header for each layer may not be present, depending on the protocol stack and the general characteristics of the frame. For example, the TCP/IP protocol stack does not use any Layer5 or Layer6 protocol headers, while the OSI protocol stack may use all of them.

The end of the frame is delineated by a CRC (Cyclic Redundancy Check), sometimes called a FCS (Frame Check Sequence). This 32-bit, 4-octet field is used to validate the integrity of the Ethernet frame upon reception. If the receiving node does not calculate the same CRC for the frame that the sender did, the frame is rejected as damaged.

2.2 IGMP Decode - Version 2

IGMP *(Internet Group Management Protocol)* is designed to allow end nodes and any immediately neighboring IP-multicast enabled routers (and Layer-3 switches) to converse about which users are interested in which multicast source traffic (i.e. "join" and "leave" operations). IGMP is used by routers and Layer-3 switches as a companion to the IP-Multicast routing protocol of choice, be it DVMRP, PIM-SM, PIM-DM, MOSPF, or other protocol. The IGMP protocol is an integral part of IP, as is the case with ICMP (Internet Control Management Protocol). IGMP's IP "protocol type" is of value two (2). The IGMP messages, encapsulated in IP datagrams, have a TTL (Time To Live) field value of one (1). In this way, the IGMP messages never move beyond the local Ethernet segment. In addition, the messages contain the IP *Router Alert Option* in their IP header. (Please reference RFC 2113 for more details on this IP header option.)

The logic for decoding of the IGMP protocol (using the Version 2 timing as defined above) is as follows:

2.2.1 Layer-2/3 Decode for IGMP

```
  if ( Layer-2 Data Link header "protocol type" field = 0x0800 and Layer-3 IP
header "protocol type" field = 0x02 )
    then the packet is a IGMP message packet.
```

2.2.2 IGMP Message Decode

The 64-bit (8-octet) *IGMP* Message is broken into these four (4) fields:

Bits	0 - 7	**Type**
Bits	8 - 15	**Maximum Response Time**
Bits	16 - 31	**Checksum**
Bits	**32 - 63**	**Group Address (IP-Multicast Source Address)**

"Type Field"

The IGMP "type" field can take on the following values:

```
0x11 = "Membership Query"
0x16 = "Membership Report, V2"
0x17 = "Leave Group"
```

with one additional type used for backwards compatibility with Version 1:

```
0x12 = Membership Report, V1"
```

Two subtypes of the Membership Query message are possible - the "General Query" used to learn which groups have members on an attached network, and the "Group-Specific Query" used to learn if a particular group has any members on an attached network. The Group Address field is used to differentiate the two subtypes (see below).

Special Note: A value of three (0x03) for the type provides the DVMRP routing protocol the ability to distribute source tree information between routers. (See below)

"Maximum Response Time" Field

The Maximum Response Time field is meaningful only in Membership Query messages and specifies the maximum allowed time (in units of 1/10 second) before sending a responding report. In all other messages, the field is set to zero (0) by the sender and ignored by receivers.

Varying this field's setting allows an IGMPv2 router to tune the "leave latency" - the time between the moment the last host leaves a group and when the routing protocol is notified that there are no more members). This field also allows tuning of the "burstiness" of IGMP traffic on a given subnet.

"Checksum" Field

The checksum field is the 16-bit one's complement of the one's complement sum of the whole IGMP message, i.e. the entire IP payload. For computing the checksum, the checksum field is first set to zero. When transmitting packets, the checksum *must* be computed and inserted into this field. When receiving packets, the checksum *must* be verified before processing a packet.

"Group Address" Field

The Group Address field is the "meat" of the IGMP message. In a Membership Query message, the group address field is set to zero (0) when sending a General Query and set to the group address being queried when sending a Group-Specific Query. In a Membership Report or Leave Group message, the group address field holds the IP multicast group address of the group being reported or left.

Other IGMP Fields

Please note that IGMP messages may be longer than eight (8) octets, especially future backwards compatible versions of IGMP. As long as the "type" is recognized, an IGMPv2 implementation *must* ignore anything past the first eight (8) octets while processing the packet. However, the IGMP checksum is always computed over the whole IP payload, not just over the first eight (8) octets.

2.2.3 Using the Decode to Implement IGMP-Snooping

By using the IGMP Layer 2/3 decode of 2.2.1, one may implement the "IGMP Snooping" technique described in Part 2. IGMP Snooping, if you recall, referred to the ability of a switch to be able to decode a multicast packet to determine if it was a control packet (IGMP) or an actual data-bearing packet. Thus, repeating the logic for section 2.2.1:

```
if ( Layer-2 DLH "MAC destination address" field is Multicast and
  Layer-2 DLH "protocol type" field = 0x0800 and
      Layer-3 IP header "protocol type" field = 0x02 )
  then the packet is a IGMP message packet
else the packet contains normal IP-Multicast dataNote:
```

The "IP protocol type" has a fixed location relative to the start of the Layer-3 IP header. It is the 10th octet (byte) in the Layer-3 IP header. This protocol value specifies what type of header follows the IP header.

2.3 PIM-SM/DM Decodes - Version 2

The Protocol Independent Multicast protocol (both for Sparse Mode and Dense Mode) uses control messages that have a IP header "protocol" field value of 103. PIM messages use either IP unicast addresses (e.g. the "Registers" and "Register-Stop" messages) or IP multicast addresses with the *ALL-PIM-ROUTERS* group value ("224.0.0.13") (e.g. the "Join/Prune" and "Asserts" messages).

Each of the nine PIM control messages is pre-pended with a "PIM Message Header" described in section 2.3.2 below.

The following logic may be used for decoding the PIM protocol:

2.3.1 Layer-2/3 Decode for PIM

```
if ( Layer-2 Data Link header "protocol type" field = 0x0800 and
  Layer-3 IP header "protocol type" field = 0x67 )
then the packet is a PIM message packet.
```

2.3.2 PIM Message Header Decode

The 32-bit (4-octet) PIM Message Header is broken into four (4) fields:

```
Bits  0 -  3: PIM Version
Bits  4 -  7: Type
```

```
Bits  8 - 15: Reserved
Bits 16 - 31: Checksum
```

"PIM Version" Field

The current version of PIM (SM and DM) is two (2).

"Type" Field

The PIM Message Header "type" field can take on the following values: (*DR*="Designated Router", *RP*="Rendezvous Point", *MC*="Multicast Traffic", *BSR*="Bootstrap Router", *IR*="Intermediate Router", *DMO*=Dense Mode Only)

```
0x00 = "Hello" // sent periodically on all router interfaces
0x01 = "Register" // sent by DR to the RP for MC traffic
0x02 = "Register-Stop" // sent by DR to the RP to stop MC traffic
0x03 = "Join/Prune" // sent by routers to upstream sources & RPs 0x04 =
"Bootstrap" // originated at the BSR and sent to IRs
0x05 = "Assert" // used to avoid duplicate paths to receiver
0x06 = "Graft (DMO)" // re-instate a pruned branch
0x07 = "Graft-Ack (DMO)" // response to a "Graft"
0x08 = "Candidate-RP-Advertisement" // sent periodically from each candidate-
RP to the BSR
```

"Reserved" and "Checksum" Fields

The "Reserved" field is ignored by receivers and is always set to zero (0) by senders. The "Checksum" field is the 16-bit one's complement of the one's complement sum of the entire PIM message, excluding the data portion in the "Register" message). The "Checksum" field is first zeroed before computing the checksum.

2.3.3 PIM Message Address Formats

Within each PIMv2 message, addresses are encoded in one of three ways - a Unicast Address, a Group (IP-Multicast) Address, and a Source Address. Each has its own 32-bit or 64-bit format within a PIMv2 message.

Unicast Address Encoding

```
Bits  0 - 7:  Address Family
Bits  8 - 15: Encoding Type
Bits 16 - 15: Unicast Address
```

Address Family is one of the following IANA-assigned values:

```
0x00 = "Reserved"
0x01 = "IPv4" // ** most common setting **
0x02 = "IPv6"
0x03 = "NSAP"
0x04 = "HDLC (8-bit multi-drop)"
0x05 = "BBN 1822"
0x06 = "802 - includes all 802 media plus Ethernet canonical format"
0x07 = "E.163"
0x08 = "E.164 (SMDS, Frame Relay, ATM)"
0x09 = "F.69 (Telex)"
0x0A = "X.121 (X.25, Frame Relay)" 0x0B = "IPX"
0x0C = "Appletalk"
0x0D = "Decnet IV"
0x0E = "Banyan Vines"
0x0F = "E.164 with NSAP format sub-address"
```

Encoding Type is always set to zero (0)

Unicast Address as represented by the given Address Family and Encoding Type

Group Address Encoding

```
Bits  0 -  7: Address Family
Bits  8 - 15: Encoding Type
Bits 16 - 23: Reserved
```

```
Bits 24 - 31: Mask Len
Bits 32 - 63: Group Multicast Address
```

Address Family is as described above.

Encoding Type is as described above.

Reserved is transmitted as zero (0). Ignored upon receipt.

Mask Len is set to 32 for IPv4 native encoding and 128 for IPv6 native Encoding.

Group Multicast Address contains the group address.

Source Address Encoding

```
Bits  0 -  7: Address Family field
Bits  8 - 15: Encoding Type field
Bits 16 - 20: Reserved field
Bits 21 - 23: S, WC, RPT field
Bits 24 - 31: Mask Len field
Bits 32 - 63: Source Address field
```

Address Family is as described above.

Encoding Type is as described above.

Reserved is transmitted as zero (0). Ignored upon receipt.

S, WC, RPT is a set of three bits with the following special meanings:

"S-bit": the *Sparse bit* is set to one (1) for PIM-SM and is used for PIM Version 1 compatibility only.

"WC-bit": the WC bit is set to one (1) if the join or prune applies to the (*,G) or (*,*,RP) entry. It is set to zero (0) if the join or prune applies to the (S,G) entry where S is Source Address. Joins and prunes sent toward the RP must have this bit set."

RPT-bit": the RPT is set to one (1,) if the information about (S,G) is sent towards the RP. If set to zero (0), the information must be sent toward S, where S is the Source Address.

Mask Len is set to 32 for IPv4 native encoding and 128 for IPv6 native Encoding.

Source Address is the address sourcing the multicast traffic.

Of all the PIM message types mentioned above, the most important message type for keeping an IP-Multicast CAM table up-to-date is the "Join/Prune" message - type 0x03.

A Join/Prune PIM message utilizes the following format:

```
Bits  0 -  3: PIM Version field
Bits  4 -  7: Type field
Bits  8 - 15: Reserved field
Bits 16 - 31: Checksum field
Bits 32 - 63: Upstream Neighbor Address field (Encoded-Unicast)
Bits 64 - 71: Reserved field
Bits 72 - 79: Num Groups field
Bits 80 - 95: Holdtime field
Bits 96 and beyond: a series of Multicast Group Blocks
----- end of Join/Prune message -----
```

Each Multicast Group Block contains:

```
Bits  0 - 31: Group Address #1 field (Encoded-Multicast)
Bits 32 - 47: Number of Joined Sources field
Bits 48 - 63: Number of Pruned Sources field
Bits 64 and beyond: a series of Joined Source Addresses, and then a series of
Pruned Source Addresses
```

Each Joined Source Address is 32-bits and is Encoded-Source. Each Pruned Source Address is also 32-bits and is Encoded-Source

The first four fields of the Join/Prune message comprise the PIM *message header* (described in 2.3.2 above). The type field is set to 0x03 for the Join/Prune type.

The *Upstream Neighbor Address* is the IP unicast address of the RPF or upstream neighbor. This is the router that is being targeted with the Join/Prune message. The *Reserved* field is transmitted as zero and ignored upon receipt.

The *Num Groups* field specifies the number of *Multicast Group Blocks* that follow the *Holdtime* field. Each *MGB* specifies a series of Join and Prune source addresses for a given IP-Multicast group.

The Holdtime field specifies the amount of time a receiver must keep the Join/Prune state alive, in seconds. If this field is set to 0xFFFF, the receiver of this message never times out (useful for ISDN lines). If this field is set to zero (0), the information is timed out immediately.

Within the MGB, we start with the *Group Address* field which specifies the IP-Multicast group that will have a series of joins and prunes, which is followed by the *Number of Joined Sources* and *Number of Pruned Sources* fields that specify the number of each. The *Joined Source Addresses* are then listed, followed by the *Pruned Source Addresses*.

The Control Plane must use these Join/Prune messages to add and remove entries to the IP-Multicast CAM, referred to as the Multicast Forwarding Table (MFT)

Conclusion

Over the past few years, organizations have dramatically increased their use of the Internet to communicate and collaborate. Many have revised their business practices in order to take advantage of the Internet as a facilitator of communication. Over the next several years, this trend should continue with IP Multicasting becoming an increasingly important technology for delivering information to large, geographically disparate audiences.

Great strides in refining and improving IP Multicasting have been made and continue to be made today. This set of articles should provide developers with a basic understanding of this enabling technology as they begin to implement IP Multicasting in their own companies or organizations.

IP-Multicasting Glossary of Terms

Broadcast

One-to-all transmission where the source sends one copy of the message to all nodes, whether they wish to receive it or not. See Unicast, Multicast.

Class D IP addresses

Used to specify multicast host groups. In Internet standard "dotted decimal" notation, host group addresses range from 224.0.0.0 to 239.255.255.255.

Core Based Trees (CBT) Routing Protocol

Unlike DVMRP or MOSPF, which construct spanning trees for each source/group pair, CBT protocol constructs a single tree that is shared by all members of the group. Multicast traffic for the entire group is sent and received over the same tree, regardless of the source. A CBT shared tree has a small number of core routers (called cores) that are used to construct the tree. Other routers may join the tree by sending a join message to the core. CBT trees are bi-directional.

Dense-mode multicast routing protocols

Dense-mode routing protocols assume that the multicast group members are densely distributed throughout the network and bandwidth is plentiful, i.e., almost all hosts on the network belong to the group. Dense-mode routing protocols Distance Vector Multicast Routing Protocol (DVMRP), Multicast Open Shortest Path First (MOSPF), and Protocol-Independent Multicast - Dense Mode (PIM-DM). See also Sparse-mode Routing Protocols.

Distance Vector Multicast Routing Protocol (DVMRP)

The first protocol that was developed to support multicast routing is called Distance Vector Multicast Routing Protocol (DVMRP), described in RFC 1075. It is used extensively on the MBONE. The approach used by DVMRP is to assume initially that every host on the network is part of the multicast group. Multicast messages are transmitted over every possible router interface as they proceed across the network, forming a spanning tree to all possible members of the multicast group. DVMRP maintains a current image of the network topology using a distance-vector routing protocol such as the Routing Information Protocol RIP. The distance metric used for RIP and DVMRP is the number of hops in the path.

Host Group

All hosts belonging to a multicast session. The membership of a host group is dynamic; that is, hosts may join and leave groups at any time. There is no restriction on the location or number of members in a host group. A host may be a member of more than one group at a time.

Internet Group Management Protocol (IGMP)

IGMP is used by multicast routers to learn the existence of host group members on their directly attached subnets. See RFC 1112.

IP Multicast

A one-to-many transmission, in contrast to Unicast, Broadcast.

An extension to the standard IP network-level protocol. RFC 1112, Host Extensions for IP Multicasting, authored by Steve Deering in 1989, laid the groundwork for IP Multicasting. The RFC describes IP Multicasting as: "the transmission of an IP datagram to a 'host group', a set of zero or more hosts identified by a single IP destination address. A multicast datagram is delivered to all members of its destination host group with the same 'best-efforts' reliability as

regular unicast IP datagrams. ... The membership of a host group is dynamic; that is, hosts may join and leave groups at any time. There is no restriction on the location or number of members in a host group. A host may be a member of more than one group at a time."

IP Multicast Datagram

A datagram delivered to all members of the multicast host group with the same 'best-efforts' reliability as regular unicast IP datagrams.

IP Multicast Router

A router supporting IGMP and one or more routing protocols, including Distance Vector Multicast Routing Protocol (DVMRP), Multicast Open Shortest Path First (MOSPF), and Protocol-Independent Multicast - Dense Mode (PIM-DM), Core Based Trees (CBT) and Protocol-Independent Multicast - Sparse Mode (PIM-SM).

MBONE

The MBONE (Multicast Backbone) is a virtual network layered on top of the physical Internet to support routing of IP Multicast packets. It has been in existence for about 5 years. See www.mbone.com for more information.

Multicast Open Shortest Path First (MOSPF)

The MOSPF routing protocol is the IP Multicast extension of the Open Shortest Path First (OSPF) unicast routing protocol. It is defined in RFC 1584. OSPF routes messages along least-cost paths, where cost is expressed in terms of a link-state metric, as opposed to hops, used by RIP and DVMRP. Each router can calculate a spanning tree with the multicast source at the root and the group members as leaves. This tree is the path that is used to route multicast traffic from the source to each of the group members.

Multicast

To transmit information to a group of recipients via a single transmission by the source, in contrast to Unicast, Broadcast. See also IP Multicast.

Protocol-Independent Multicast (PIM) Routing Protocols

The Protocol Independent Multicast (PIM) routing protocol is currently under development by an IETF working group. The objective of the designers of PIM is to develop a standard multicast routing protocol that can provide scalable inter-domain multicast routing across the Internet that is not dependent on the mechanisms provided by any particular unicast routing protocol. In contrast, DVMRP is based on RIP and MOSPF on OSPF. PIM has two modes, dense and sparse, discussed below.

PIM- Dense Mode (PIM-DM) Routing Protocol

PIM also defines a new dense-mode protocol for "dense" groups, instead of relying on existing dense-mode protocols such as DVMRP and MOSPF. See the entry for dense-mode multicast routing protocols. PIM-DM control message processing and data packet forwarding is integrated with PIM-SM operation so that a single router can run different modes for different groups.

PIM- Sparse Mode (PIM-SM) Routing Protocol

PIM-SM differs from existing dense-mode multicast algorithms in two essential ways. Routers with directly attached or downstream members are required to join a sparse mode distribution tree by transmitting explicit join messages. If a router does not become part of the pre-defined distribution tree, it will not receive multicast traffic addressed to the group.

Real-Time Streaming Protocol (RTSP)

RTSP is an application-level protocol for control over the delivery of data with real-time properties to enable controlled, on-demand delivery of real-time data, such as audio and video.

Real-Time Transport Protocol (RTP)

RTP provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services.

Reliable multicast protocols

Reliable multicast protocols overcome the limitations of unreliable multicast datagram delivery and expand the uses of IP Multicast.

ReSerVation Protocol (RSVP)

RSVP, the ReSerVation Protocol, enhances the current Internet architecture with support requests for a specific quality of service (QoS) from the network for particular data streams or flows.

Sparse-mode multicast routing protocols

A selective one-to-many transmission, in contrast to Unicast, Broadcast. See also IP Multicast.

Sparse-mode routing protocols assume that the multicast group members are sparsely distributed throughout the network and bandwidth is not necessarily widely available, for example across many regions of the Internet. It is important to note that sparse-mode does not imply that the group has a few members, just that they are widely dispersed. Sparse-mode routing protocols include Core Based Trees (CBT) and Protocol-Independent Multicast - Sparse Mode (PIM-SM).

Spanning Tree

For efficient transmission, multicast routers construct a spanning tree from the multicast source at the root of the tree to all the multicast receivers as leaves of the tree. A spanning tree has just enough connectivity so that there is only one path between every pair of LAN's, and it is loop-free.

Tunneling

An interim deployment strategy to connect islands of multicast routers separated by links which do not support IP Multicast. Tunneling is used extensively in the MBONE. Tunneling is discussed in the IP Multicast Initiative white paper Introduction to IP Multicast Routing.

Unicast

Point-to-point transmission requiring the source to send an individual copy of a message to each requester.