

Multicast Session Address Allocation & Directory  
Architecture: A PhD Proposal

Piyush Harsh  
pharsh@cise.ufl.edu  
Computer and Information Science and Engineering  
University of Florida, Gainesville, FL 32603

March 24, 2008

## 0.1 PhD Proposal Document

### 0.1.1 Summary

IP Multicast holds great promise for the Internet in the near future. With the explosion in the multimedia content providers, very soon current Internet infrastructure will begin to feel the pressure due to higher bandwidth demands that may limit the content provider subscriber base and / or limit the quality of the stream. Multicast is best suited to address such concerns of scale and efficient network bandwidth utilization. One of the limiting factors that have stopped widespread Multicast deployment is the lack of global session's directory structure. Ubiquitous URLs along with the DNS infrastructure which has been a major factor for fast and increasing popularity of the Internet and the lack thereof for the Multicast points towards one of the last stumbling blocks for its widespread deployment.

### 0.1.2 Research Goals

- Proposal for a global and scalable Multicast session directory architecture
- Design of user friendly URLs for various Multicast Streams
- Efficient Multicast address allocation infrastructure
- Possible elimination of Globally Scoped Multicast addresses collision among sessions in various administrative domains

### 0.1.3 Research Motivation

IGMP v3 which is still in draft stage promises to simplify the Multicast protocol complexity significantly. It gives up on traditional ASM (Any Source Multicast) model in favor of simpler SSM (Source Specific Multicast) model. This new model greatly simplifies the network complexity by removing the source discovery responsibility from the routers but places them on the end hosts. It would be highly desirable and convenient if end users can query for Multicast sources in real time.

Although IPv6 solves IP Address scarcity issue but it still remains years away from full deployment. Multicast address in IPv4 traditionally defined as class D address is a scarce resource strictly managed by IANA based on IETF guidelines. Therefore efficient Multicast addresses distribution and reuse is highly desirable.

Proper URL scheme in order to map correctly to the Multicast resource should greatly improve the usability of the technology and therefore could result in faster and wider acceptance and deployment in the consumer networks.

Address collisions among different multicast sessions lead to significantly added burden on the end hosts' IP stack in order to filter out the garbage stream data. Therefore for efficient network resource and CPU cycle utilization it becomes extremely important to minimize address collisions.

# Contents

0.1	PhD Proposal Document . . . . .	1
0.1.1	Summary . . . . .	1
0.1.2	Research Goals . . . . .	1
0.1.3	Research Motivation . . . . .	1
<b>1</b>	<b>Introduction to IP Multicast</b>	<b>6</b>
1.1	IGMP (Internet Group Management Protocol) . . . . .	7
1.1.1	IGMP Version 1 . . . . .	8
1.1.2	IGMP Version 2 . . . . .	9
1.1.3	IGMP Version 3 . . . . .	10
1.2	PIM-SM (Protocol Independent Multicast - Sparse Mode) . . . . .	10
<b>2</b>	<b>IP Multicast Address Classifications</b>	<b>13</b>
<b>3</b>	<b>IP Multicast Address Allocation Problem</b>	<b>15</b>
3.1	Current Strategies and related work . . . . .	16
3.2	Our Proposed Solution (HOMA) . . . . .	17
3.2.1	HOMA Address Allocation Algorithm . . . . .	18
3.2.2	Time-Delay Analysis . . . . .	21
3.2.3	Advantages of HOMA . . . . .	21
<b>4</b>	<b>Multicast Session Discovery Architecture</b>	<b>24</b>
4.1	Need for Multicast Session Discovery . . . . .	24
4.2	Current Strategies and tools for session discovery . . . . .	25
4.3	mDNS: DNS-aware multicast session directory architecture . . . . .	27
4.3.1	mDNS hierarchy construction . . . . .	28
4.3.2	Session registration in mDNS . . . . .	31
4.3.3	mDNS search operation . . . . .	32
4.3.4	search example . . . . .	34
4.3.5	High level analysis of mDNS . . . . .	34
<b>5</b>	<b>What remains to be done ...</b>	<b>36</b>
5.1	System Simuations . . . . .	36
5.2	Simulation Parameter Space Selection . . . . .	37
5.3	Protocol Design . . . . .	37

<i>CONTENTS</i>	3
5.4 mDNS - Ongoing / Future Work (Time Permitting) . . . . .	37
5.5 Framework for HOMA and mDNS integration . . . . .	38
<b>Appendices</b>	<b>38</b>
<b>A List of publications</b>	<b>39</b>

# List of Figures

1.1	IGMP v 1 Message Format . . . . .	8
1.2	IGMP v 2 Message Format . . . . .	9
1.3	IGMP v 3 Message Format . . . . .	10
3.1	Global TLDs Overlay . . . . .	18
3.2	ISP Tree Rooted at Global TLD . . . . .	19
3.3	Peer n/w among sibling nodes . . . . .	19
3.4	A general scheme of HOMA nodes hierarchy . . . . .	22
4.1	a typical MSD hierarchy . . . . .	29
4.2	an example .edu hierarchy . . . . .	30
4.3	an general mDNS hierarchy . . . . .	35

# List of Tables

# Chapter 1

## Introduction to IP Multicast

IP Multicast [18] [62] [42] [22] [69] lies on the other end of the Internet delivery model. Today, the Internet supports three kinds of packet delivery mode, unicast, anycast [40] [39] [47] [56] and multicast. In unicast model, every IP packet has one source address and one destination address. In case of IP unicast, packet routing is based on the destination address and every unicast packet can take a path independent of the previous packet and there is no distribution structure in the core network.

Anycast model of packet delivery lies somewhat in between unicast and multicast model. It is an addressing and routing scheme where packets are delivered to any one of the multiple possible destinations. Almost always this delivery is made to the destination host which is nearest or best host as determined by the routing topology.

Currently, IP multicast is configured to support any source multicast (ASM). In IP multicast, data packets can be delivered to many different hosts. The data is forwarded along the multicast distribution tree to multiple receiver hosts. Routing decisions for a multicast packet is based on the source address (RPF [13] check) instead of the destination address for unicast and anycast. This distribution tree is setup by the participating multicast enabled routers in the core network and the consumer networks.

IGMP [27] [14] [69] (Internet Group Management Protocol) is an essential component that allows end hosts to join or leave a multicast group. IGMP version 3 which is still in the drafts committee promises to change the multicast landscape significantly. With the ASM model in place, the participating routers have to do a lot of processing and maintain lot of states. The routers have the responsibility of source discovery and maintaining the proper distribution tree. IETF committee and network operators believe this complexity in the core network has been delaying the true widespread deployment of IP multicast.

With IGMP v 3 [9] [22], the responsibility of discovering the multicast sources

move from the routers to the end hosts. End hosts with IGMP v 3 would have to specify the source along with the group address in order to join the Multicast group. This new model is now being referred to as SSM [7] [36] (Source Specific Multicast). Although ASM model is more flexible and may support wide variety of services and applications, IETF task force members believe SSM would be sufficient for service models suitable for large scale content distributors and the reduction in the network / core complexity would encourage faster and widespread deployment of the next generation multicast.

Over the last 2 decades many new multicast protocols have been introduced into the Internet. Different classes of protocols have been deployed to achieve different level of control and functionality in the network. For routing within a given network Intra-network multicast protocols like DVMRP [68] [66], PIM-DM [24], PIM-SM [23] [16], MOSPF [51], CBT [4] etc. are used. Similarly for routing among different autonomous systems (AS) Inter-network protocols such as MBGP [6] and M-ISIS are used.

Since in IP-multicast, packet forwarding decision is based on RPF [13] (reverse path forwarding) checks, it becomes necessary to maintain some kind of RPF Table within the multicast enabled routers. Some of the Intra-AS multicast routing protocol such as DVMRP include mechanisms within themselves to populate the RPF check table. Others such as PIM [15] [69] [22] depend on other protocol to set up this table. In fact this is one of the reasons for the increasing popularity of PIM-SM protocol for Intra-AS routing. In most cases such protocols use the unicast routing tables, which may be populated using routing protocols like RIP [46], OSPF [52] [12], IS-IS [2] etc., for RPF check as well.

Since currently PIM-SM and IGMP v 3 seems to be gaining grounds over other competing protocols [22], I will next describe these protocols in details.

## 1.1 IGMP (Internet Group Management Protocol)

IGMP (Internet Group Management Protocol) is the primary multicast control protocol that enables the end-hosts to contact the first hop router and express interest in a particular multicast session elsewhere in the Internet. Generally IGMP messages are not supposed to be forwarded beyond 1 hop router. If the edge router on the LAN where the host resides is not multicast capable, it may simply forward that IGMP Join request to the next upstream router that may be multicast capable, this is called IGMP proxy.

IGMP exists in three flavors namely - versions 1, 2 and 3. IGMP version 3 is currently in draft stages. Any Internet host interested in joining a multicast session must run some version of IGMP. Before any host could start receiving multicast packets, it must configure its layer 2 LAN card by mapping the corresponding Ethernet [61] [63] [62] address for the multicast channel address it is interested in.

### 1.1.1 IGMP Version 1

IGMP version 1 has been described in depth in RFC 1112. It was more or less a refinement of the original "Host Membership Protocol" defined as part of the Dr. Steve Deering's doctoral thesis. IGMP messages are encapsulated [71] within IP packet with protocol field set to 2. IGMP v 1 messages look like this:

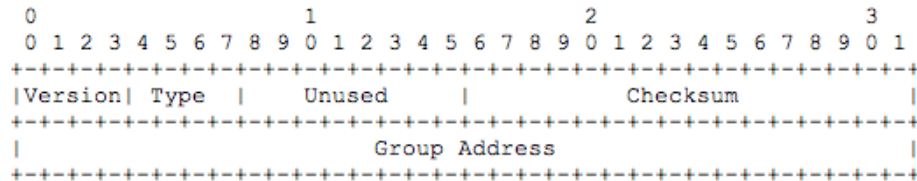


Figure 1.1: IGMP v 1 Message Format

*Version:* Set to 1 for IGMP version 1

*Type Field:* IGMP v 1 uses two type of messages namely "Membership Query" and "Membership Response"

*Checksum Field:* 16 bits, 1's complement of the 1's complement sum of IGMP message.

*Group Address Field:* It contains the group address when a membership report is being sent, it is normally zero when sent as membership query packet and is ignored by the hosts.

In IGMP version 1 the query router periodically multicasts IGMP membership query to all hosts on the All-Hosts multicast group 224.0.0.1. Hosts interested in receiving multicast group packets must send back a membership report containing the interested group address in their report. Multiple hosts interested in the same group suppress their reports by randomly picking up a response time from an interval and cancelling their own report if they overhear some other host's report containing the same group address. This is IGMP report suppression mechanism. [69]

In IGMP version 1, there is no IGMP-querier election algorithm in case there are multiple multicast enabled routers in the subnet. IGMP version 1 depends on layer 3 protocol to decide a designated router for the subnet. In order to cut down on the join latency, if a host wishes to join a group, it can elect not to wait for the next membership query to come from the IGMP-querier, instead may generate IGMP membership report and send it on the All-Hosts group 224.0.0.1 indicating the group it is interested in. The process to leave a particular group is very simple in version 1. Hosts simply ignore the membership reports generated

by the IGMP-querier and after timeout (usually it is 3 times the query interval or 3 minutes) IGMP-querier stops forwarding multicast traffic for that group to its subnet.

### 1.1.2 IGMP Version 2

IGMP version 2 was accepted as a standard by the IETF in November 1997. RFC 2236 contains detailed description on version 2 and is intended as an update to RFC 1112.

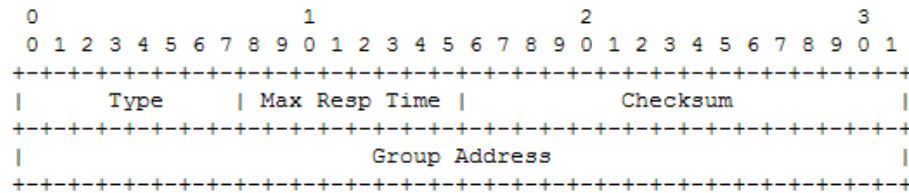


Figure 1.2: IGMP v 2 Message Format

The two major differences between IGMP version1 and version 2 Membership report messages are -

1. IGMP v2 query messages now fall into two categories, one General Queries which are essentially same in function as that in version 1 and second Group-Specific queries which are intended for a specific group related queries.
2. The Membership Reports have different IGMP type codes in version 2 compared to those in version 1.

Some of the new features that were introduced in IGMP version 2 include:

- Capability to elect IGMP Query Router among themselves, in IGMP version 1 this was left to layer 3 protocol.
- New field in the header namely "Max Resp Time" or Maximum Response Time was added to fine tune the burstiness in the query process and to control the leave latencies.
- Group-Specific Query messages now allows router to manage group membership for any specific group instead of every time resorting to general query messages.
- Now hosts could notify the query routers if they wish to leave any group, this resulted in much better leave latencies and better utilization of network resources.

IGMP version 2 was designed to be backward compatible with IGMP version 1 messages.

### 1.1.3 IGMP Version 3

IGMP version 3 brings many interesting feature and additions since IGMP version 2 was introduced. IGMP version 3 is yet to be made a standard but the process is near completion. IGMP version 3 is described in RFC3376. The message format for IGMP version 3 is shown below:

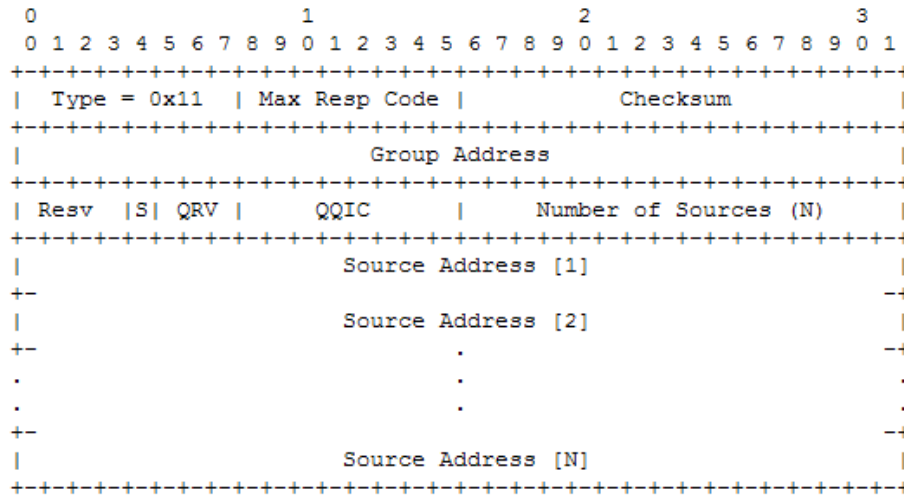


Figure 1.3: IGMP v 3 Message Format

IGMP version 3 has provisions to allow more control by session members over sources from where they wish to receive multicast traffic. It extends the join / leave process beyond multicast groups to allow joins and leaves to be requested for specific group sources using IGMP version 3 (S, G) Join/Leave messages.

## 1.2 PIM-SM (Protocol Independent Multicast - Sparse Mode)

Protocol Independent Multicast or (PIM) in short can be used in both dense mode and sparse mode. In past many years PIM-SM has clearly emerged as the protocol of choice for deployment as Intra-AS multicast protocol. As mentioned earlier, one of the strengths of PIM is that it makes use of preexisting routing table for RPF check. PIM exists in two versions namely version 1 and version 2. Version 2 has been defined extensively in RFC 2117 which has been made obsolete by RFC 2362.

PIM version 1 used hacked version of IGMP with IGMP version set to 1 and type field set to 4, different PIM messages were distinguished based on different IGMP Code field values. With version 2, PIM was assigned its own IP protocol number 103.

Multicast messages in PIM are transmitted from multicast sources either via RPT (Rendezvous Point tree) or SPT (Shortest Path tree) distribution trees. Before a source starts transmitting data or even a receiver starts receive multicast data, the designated router on their LAN must know about the RP (Rendezvous Point) for the multicast group in question. Multicast group to RP mapping can be achieved using three methods -

- Static group-to-RP mapping
- Cisco Systems auto-RP
- PIM bootstrap router [25] (BSR)

For these three, the first one is simplest but has a huge administrative burden in case the mapping changes later on. Cisco Systems auto-RP [26] and BSR are dynamic protocols that dynamically selects preferred RPs among several RP-candidate routers for a given multicast group.

The designated multicast router on any LAN is supposed to cache the RP announcement messages and maintain and update the group-to-RP mapping. Once the designated router received an IGMP (\*, G) Join message, it initiates tree graft process by forwarding the (\*, G) join towards the RP using the RPF table check against the candidate RP address. Once the distribution tree has been created / exists, multicast data can start flowing down this distribution tree from the RP node towards the interested host. In the discussion above (\*, G) denotes the specified multicast group irrespective of the transmitting source address (denoted appropriately by a \*).

Once the data reaches the host, the designated router discovers the source address from the payload and then initiates creation of SPT using the (S, G) Join on the uplink interface for that source 'S' which it determines through the RPF check. SPT distribution trees are much more efficient compared to RPTs and therefore a Cisco Networks and Juniper Network router usually immediately switches to SPTs.

A source transmitting data on a multicast group must send data to the RP. The designated router encapsulates the data inside IP packet and sends PIM Register message directly to the RP. If no receiver exists for that group at that RP, it sends back Register-STOP message to the designated router for the source's LAN which puts that router in a periodic wait state. If RPT distribution exists implying that there are receives for that multicast group in the network, the RP decapsulates the payload and sends it down the tree. It also initiates its own SPT Join towards that sender in order to start receiving the multicast data natively.

The above description is just the simple case description of what happens within the PIM-SM framework, the algorithm behaves slightly differently for various situations but the essence remains the same. For the sake of completeness, listed below are the various PIM version 2 message types:

0. Hello

1. Register (used in PIM-SM only)
2. Register-Stop (used in PIM-SM only)
3. Join / Prune
4. Bootstrap
5. Assert
6. Graft (used in PIM-DM only)
7. Graft-Ack (used in PIM-DM only)
8. Candidate RP-Advertisement (used in PIM-SM only)

## Chapter 2

# IP Multicast Address Classifications

In IP version 4, multicast addresses are a scarce resource whose allocation and use is strictly governed by IANA which follows IETF recommendation on address allocations. Address availability is not an issue in IP version 6, but since IPv6 [17] [30] is many years away from significant deployment, it becomes important to understand the limitations and restrictions in the usage of multicast addresses. IANA generally does not assign static multicast addresses. Static addresses assigned by IANA are generally allotted and reserved for specific network control algorithms.

IP Multicast addresses have been assigned to the old class D address space. These addresses have first 4 prefix bits fixed at 1110; and hence IP Multicast addresses range between 224.0.0.0 to 239.255.255.255. Below are some of the static addresses that have been allocated by the IANA mainly for network control purposes.

Link-Local Multicast Addresses - 224.0.0.0 to 224.0.0.255, these have been allocated to be used for network control messages on a LAN segment. Regardless of TTL values in the IP packets with these addresses, LAN routers do not forward these packets. Some of the popular examples of Multicast Addresses that belong to this range are -

- 224.0.0.1 - All Hosts
- 224.0.0.2 - All Multicast Routers
- 224.0.0.12 - DHCP [21] Server/Relay Agent

Specifically Allocated Multicast Addresses - 224.0.1.xxx, IANA sometimes assigns addresses from this range for some specific network protocol or applications that justified technical merit to have their own multicast address. Some of the more well known addresses from this address range include -

- 224.0.1.21 - Mtrace [10]

- 224.0.1.39 - Cisco-RP-Announce [26]
- 224.0.1.40 - Cisco-RP-Discovery [26]

Administratively Scoped Multicast Addresses - 239.0.0.0 to 239.255.255.255, this range has been reserved by IANA for private multicast networks. Their unicast counterpart would be the range 10.0.0.0/8. This address range is free for use within a multicast domain as long as the border routers can filter incoming / outgoing multicast packets that belong in this range. This helps conserve the limited addresses because it promotes address reuse within different multicast domains.

Some of the other popular range allocations by IANA are listed below -

- 224.2.0.0/16 - Session Announcement Protocol (SAP) / Session Description Protocol (SDP [33]) range
- 232.0.0.0/8 - The Source Specific Multicast (SSM) range
- 233.0.0.0/8 - The AS-encoded, statically assigned GLOP [48] range

These above three ranges are global in nature. The multicast packets belonging to these ranges need not be filtered out by the boundary routers and hence can traverse the whole of the Internet. Out of above three ranges, the SAP [34]/SDP [33] range and the SSM range are dynamic in nature. That is any application is free to pick addresses belonging in these ranges and may start transmitting data on that channel.

Some Internet applications, for example a globally scoped stock exchange ticker application, may require a statically assigned multicast address. And since IANA usually is very reluctant to assign static addresses unless there is a technical sound reasoning behind the proposal, the only other way out for service provider may be to turn to their ISPs for static allocation in the GLOP address range.

Under GLOP scheme, organizations which have AS number reserved from IANA can allocate 255 statically assigned multicast addresses to applications per AS number. These addresses can be assigned from multicast address range which is constructed simply as -

*233.[First byte of the AS number].[Second byte of the AS number].0/24*

Interestingly enough GLOP does not stand for any acronym but was chosen as appropriate name for this allotment scheme.

## Chapter 3

# IP Multicast Address Allocation Problem

As already mentioned earlier, IP Multicast addresses are shared resource. Applications and application writers are free to choose multicast addresses to be used by their application. This has the potential to create conflicts in the shared address space in the absence of well defined address allocation and maintenance mechanism. Although original class D address range was thought to be sufficient when the IP multicast was in its infancy, with the growing popularity of multimedia data streams and emergence of high speed IP networks all over the world, address range sufficiency for current and future applications has really become an emergent and challenging issue. With IP v4 address space, random address allocation collision probability is no more negligible. Imaging a situation where multiple multicast sessions that may be operating completely independently of one another, somehow pick the same multicast channel address, this would then result in significant cross-talk among these applications necessitating application designers to provision of filtering out garbage data. This would make applications more complex, would result in wasted network resources and wasted CPU cycles at the end hosts.

Deployment of IP v6, IGMP v 3 and deployment of SSM homogenously across the Internet is the obvious solution to this problem. But the ground realities are not that promising, at least not in the foreseeable future. ISP's reluctance to upgrade hardware and high deployment of ASM may push the changeover by many years. In the face of ground realities, it seems only reasonable to research into ways to better manage the limited IP v4 multicast address space with the goal to reduce address collision among different group sessions and additional goals of optimal address space utilization and timely reclamation and less fragmentation.

Keeping these goals in mind we will next provide details on some of the existing research that has been done in this field and our proposed solution to the above stated problem and justification for our proposal.

### 3.1 Current Strategies and related work

MBONE tool sdr is still in use by some applications for address allocation for a newly created multicast session. For a globally scoped session, sdr allocates address randomly selected from the SAP/SDP range 224.2.0.0/16. While random allocation scheme is simple and easy to implement, it does not scale well as number of sessions increase. There are bound to be address clashes in truly random allocation schemes.

'sdr' alleviates some of the allocation woes by using informed random multicast allocation or IRMA [38]. This introduces an additional problem of global session state information which must be maintained by the sdr tool. This scheme might work for small number of sessions in a smaller multicast scope. And the effectiveness of such a scheme is heavily dependent on the session announcement message delays and packet loss rates on the Internet. And on the global scale, maintaining individual session states is truly impractical.

IPRMA [31] or Informed Partitioned random Multicast Address Allocation scheme which was proposed by Van Jacobson was a partial improvement on reducing address collision while allocating session addresses locally. The author shows that depending on the number of partitions in IPRMA, the address collision rate varied in between  $O(\sqrt[3]{n})$  and  $O(n)$  where 'n' is the number of addresses available for allocation. The optimal rate of  $O(n)$  was achieved in the case where no two TTL values fell in the same partition. Ideally this would suggest having as many partitions as there could be different TTL scopes for various multicast sessions. This introduced effective utilization problems where one of the higher demand partitions would become full while other partitions remaining underutilized.

In another approach [58], it was suggested to pair multicast address and port numbers in order to extend the multicast address space as well as to achieve port resolution. The authors proposed a distributed multimedia address management in which the scheme could generate at least 16 multicast addresses per node. The address generation scheme was tightly integrated with the network class address. But the way they generated multicast addresses necessitated internetwide multicast enabled router upgrade. The scheme also left portions of multicast address space unused. It also required routers to look into layer 4 port number field while making routing decisions and had the potential to generate duplicate multicast traffic. They also compared their scheme with IBM's Heidelberg Multicast Protocol (HeiMAP) [70] which proposed a multicast address allocation scheme but only in the context of LANs. Under HeiMAP scheme, each host maintained table of all multicast addresses in use and the address allocation scheme was a two phase negotiation protocol. The major drawback in this scheme was the broadcast nature of the protocol.

MASC / BGMP [41] architecture for hierarchical and dynamic multicast address allocation has been proposed. MASC proposal has lots of nice features such as global scalability. Its hierarchical address prefix allocation scheme gels well with CIDRized [5] philosophy on network address assignments. Their scheme also results in compact routing table and less third party dependence for

efficient multicast routing. One nice feature is the multicast tree being rooted in the domain owning the multicast prefix chunk.

MASC protocol wait period of almost 48 hours before claiming a set of addresses could result in potential collision related instability on the global scale. Also threshold based address claim mechanism seems defensive algorithm at best. Because of 48 hour wait period before claiming an address set, there could be instances where in MASC the MAAS servers must resort of random address allocation to requesting sessions even though there might still be available free addresses in the parent's address set.

Daniel Zappala et al. [73] [45] presented a very comprehensive analysis of the multicast address allocation problem. They compared simulation results of various allocation algorithms including MASC [41], Cyclic [44] and MaxQ [73] and found that surprisingly prefix based allocation schemes did equally well compared to contiguous allocation schemes including the scheme proposed by Paul E Tsuchiya [67]. Their simulation also pointed that allowing just 2 address chunks to be owned by sub domains in MASC protocol was too restrictive and in fact with 4 chunks allowed the overall allocation performance to improve significantly.

## 3.2 Our Proposed Solution (HOMA)

Our proposed multicast address allocator scheme tries to overcome some of the shortcomings of MASC proposal by incorporating some recommendations by researchers such as Daniel Zappala and team and making use of a hybrid hierarchical overlay network [20] of address allocator servers on the similar lines of MASC proposal. In addition we augment the architecture with sub-domain level node peering using dedicated multicast channels at each level in the hierarchy. We conjecture that our proposed architectural modification should result in better address space utilization while trying to minimize routing flux at the global level at the cost of slightly higher routing flux at the lower level routers. Our proposal also tries to retain the global address allocation on the lines of unicast CIDRized scheme as much as possible on the similar lines of MASC. But we try to improve on the latency by forgoing claim-collide scheme for request-reply model.

In our design, IANA initially assigns the globally scoped multicast addresses among global TLDs (see figure 3.1). This division might take into consideration global statistics on multicast session's usage pattern and address demand. IANA involvement in our scheme is only limited to this initial address allocation to each of the TLD.

Each global TLD serves as the root level domain for the regional and enterprise domains under its jurisdiction.

In order to maximally utilize the multicast addresses, each sibling domain at any level also forms a dedicated peer network which could be an IP overlay or using a multicast channel. Necessary information for forming the overlay peering could be transmitted to each of the siblings at the next layer by the

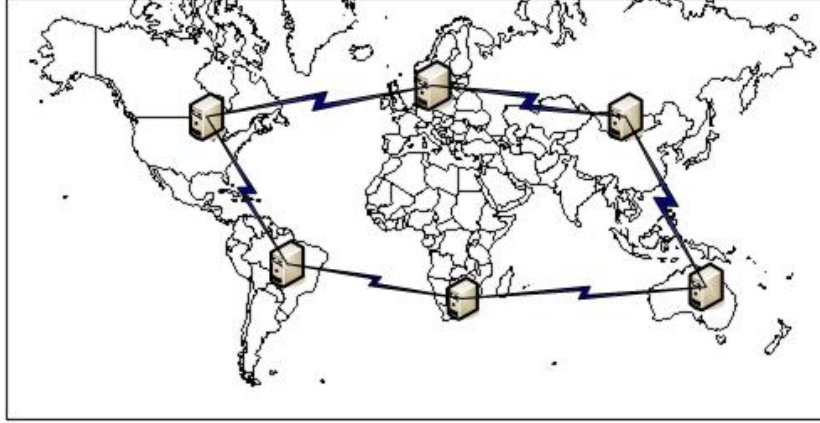


Figure 3.1: Global TLDs Overlay

parent node. For instance in the example tree hierarchy (see figure 3.3), ATT, Sprint and MCI forms a peer network among themselves. This peerage network is constructed at each level among the sibling nodes at that level in the tree hierarchy.

### 3.2.1 HOMA Address Allocation Algorithm

Each of the nodes in HOMA [35] framework maintains two parameters  $\alpha$  and  $\beta$  from the time an address block is allocated to it from the parent node until the time when the block lease expires. The values of  $\alpha$  and  $\beta$  are updated every 5 minutes duration.

Let  $\lambda$  be the number of new address requests within current 5 minute time slice. Let  $\mu$  be the number of address release by multicast applications within the same time frame. Then -

$$\begin{aligned}\alpha_{new} &= \lambda \cdot p + \alpha_{old} \cdot (1 - p) \\ \beta_{new} &= \mu \cdot p' + \beta_{old} \cdot (1 - p')\end{aligned}$$

where parameters  $p$  and  $p'$  are experimentally determined. The parameters  $\alpha$  and  $\beta$  are used as an estimate of future rate of new address requests and release of old addresses respectively.

Also let  $\gamma$  denote the address utilization factor at each node that when reaches the predetermined threshold value, would trigger the additional address request protocol within the HOMA node. The additional address requirement can be computed as follows -

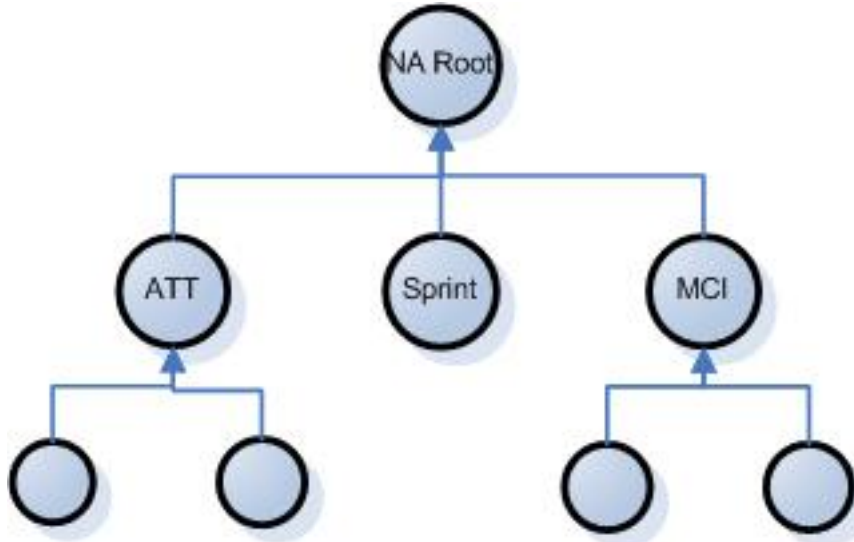


Figure 3.2: ISP Tree Rooted at Global TLD

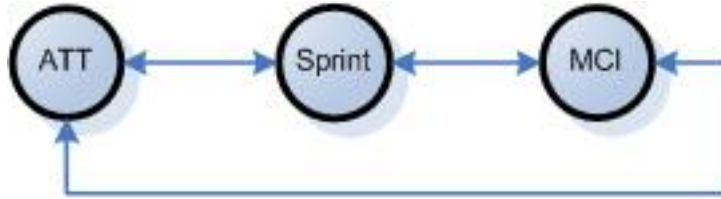


Figure 3.3: Peer n/w among sibling nodes

$$\text{Let } N = \lceil \text{lease time} - \text{current time} \rceil \div 5$$

Here  $N$  represents the number of 5 minute slots until the current address set allotted to this HOMA node expires. Then additional addresses anticipated  $\delta$  is given by

$$\delta = \lceil (\alpha - \beta) \cdot N \rceil - \#free\_addresses\_remaining$$

Let us assume that first time a HOMA node is brought online it directly contacts the parent node for a chunk of multicast address, it gets the sibling peerage details from the parent node and joins the sibling peer network. All this can be considered part of the HOMA bootstrapping process.

---

Pseudo-code for address allocator module

---

```

if incoming request is for a new channel address by a multicast application
then
  if free channel address available then
    negotiate address lease
    allocate address to requesting application
    update  $\gamma$  and  $\delta$ 
  else if free channel address is not available then
    allocate a channel address randomly from the parent's address space
    update  $\delta$ 
  end if
end if
if incoming request is to release one of the already allotted addresses by a
multicast application then
  if the address belongs to the set owned by this HOMA node then
    add it to the free address list
    update  $\gamma$  and  $\mu$ 
  else if address does not belong to the address set owned by the HOMA
node then
    do not add to free address list
    update  $\mu$ 
  end if
end if
  At every 5 minutes interval -
  recompute  $\alpha$  and  $\beta$ 
  set  $\lambda = \mu = 0$ 
  After every address allocation / de-allocation check the value of updated  $\gamma$ 
if  $\gamma < \text{threshold}$  then
  do nothing
else if  $\gamma \geq \text{threshold}$  then
  Compute the anticipated additional address required  $\delta$ 
  if  $\delta > 0$  then
    initiate a request for  $\delta$  number of addresses on the sibling peer network
    wait for 2 minutes for responses
    if response comes then
      add addresses to the free address pool
      track the lease associated with these addresses
    else if no response comes then
      initiate additional address request to parent HOMA node
    end if
  end if
end if
if additional address request is received on the sibling peer network then
  Compute possible disposable address count  $\theta$  using the following relation

```

```

         $\theta = \#free\_addresses\_remaining - [(\alpha - \beta) \cdot N]$ 
if  $\theta > 0$  then
    indicate willingness to allocate  $\theta$  set of addresses to the sibling node
    treat this allocation just like any other address allocation
else if  $\theta \leq 0$  then
    do nothing
end if
end if

```

---

This pseudo-code is implemented at each HOMA node and each node executes this pseudo-code independently of one another. There is no centralized component in the above pseudo-code.

### 3.2.2 Time-Delay Analysis

For purpose of doing time delay analysis suppose that that with probability  $\pi$  the additional address demand is satisfied from one or more sibling nodes. In the worse case any node must wait for a duration of 2 minutes before sending additional address request to it's parent node, we can define a recursive equation for the overall delay in terms of tree depth'd'.

$$Delay = 2 \cdot \pi + (2 + \Lambda_d) \cdot (1 - \pi)$$

where  $\Lambda_d$  is the delay if the request must be made to one's parent node.

$$\Lambda_d = 2 \cdot \pi + (2 + \Lambda_{d-1}) \cdot (1 - \pi)$$

Here in the above equation,  $\Lambda$  must also account for time delay in locating a possible chunk of address in ones internal free addresses list.

The value of  $\pi$  remains to be experimentally determined. It can be calculated by tracking the fraction of cases during any simulation run where the additional address demand was satisfied by sibling nodes. We conjecture that this delay behavior is more suitable for a dynamic session scenario than delay behavior of claim-collide mechanism in MASC proposal.

### 3.2.3 Advantages of HOMA

Since HOMA distributed algorithm can be implemented in software, there is no need for ISPs to update their routing hardware. Ability to exist in current deployed environment is one of the greatest strengths of proposed algorithm. Lack of any centralized components in the proposed algorithm is in line with accepted trend in the Internet management protocol design community. It also makes the algorithm robust against localized failures.

The fact that the global TLDs are well known in our design; it could be used effectively to design simple DDoS [29] prevention strategy. The child nodes of first few level deep parent nodes could also be assumed well known thereby enabling the parent node to filter out protocol messages from downstream clients

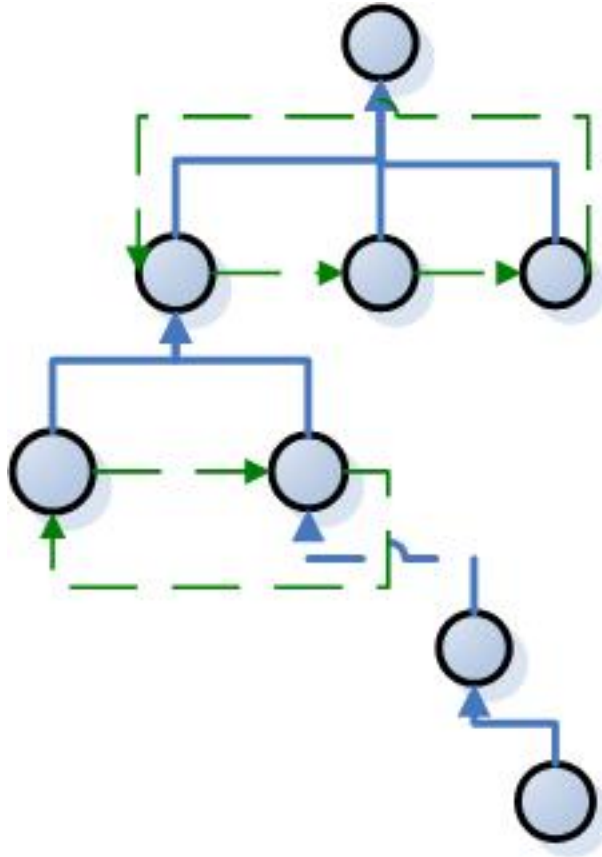


Figure 3.4: A general scheme of HOMA nodes hierarchy

more effectively [28] [55] [37]. Resilience against DDoS attacks in the Internet management architecture [29] is becoming more and more important.

Another important feature in HOMA design is minimization of routing flux. Since the address allocation is hierarchical in our proposal, any address sub lease and exchanges among the sibling nodes would result in routing table entry changes in the sibling HOMA nodes' parent domain's multicast router and no higher. Routing stability is paramount for any globally scalable Internet service architecture.

Since the address allocation scheme of HOMA proposal is more responsive and real time by design, there is no need for using a very conservative threshold setting as proposed in MASC paper. We conjecture that using non conservative threshold in HOMA algorithm would result in better address space utilization. One of the reasons for this is improved address reuse among siblings and HOMA being open to both chunks as well as individual address allocations to child / sibling nodes.

## Chapter 4

# Multicast Session Discovery Architecture

### 4.1 Need for Multicast Session Discovery

One of the last few missing pieces in the multicast wide scale deployment is the global session discovery infrastructure and protocol framework. One of the major factors behind fast popularity of the IP unicast mode of the Internet was the much more usable and ease of recall of the URLs and the global DNS [50] hierarchical structure availability. Imagine what would be the net like if we still had to refer to websites using the IP's dotted decimal format [72]. Of course, the role of other dominant factors such as global access to information and seamless inter-operability among heterogeneous networks can not be emphasized less.

Some groups argue that IP Multicast lacks killer applications like email for the Internet for the lack of popularity. But we disagree; there already exist lots of killer applications that have potential for far reaching impact globally. Multimedia content delivery and large scale group interaction capability using IP multicast are just a couple of them. What is conspicuously missing from the multicast is the DNS like architecture. There do exist multicast sessions in the Internet today but to become a group recipient, you must find out the multicast session address beforehand. Most of the times, these popular session addresses are made available using mass emailing, IRC sessions [54] and bulletin board postings. There does not exist a uniform session naming scheme. We believe with global infrastructure support towards this goal would make IP multicast much more user friendly and would help generate lot more widespread consumer demand just like DNS did for the Internet.

With future deployment of IGMP v3 and ISPs switching to SSM mode of IP multicast from current ASM model, source discovery which currently is responsibility of IP layer 3 routers (and is a major reason for multicast protocol complexity) will shift to end hosts at the consumer networks. Global session directory will naturally gear towards meeting this demand. Somehow this research

has slipped from the network researchers' radar worldwide and very little work has been done to date to address this concern. All these reasons are a major motivating factor behind my dissertation work.

In the following sections I would highlight some of the current strategies that have been proposed followed by our proposal for the globally scalable multicast session directory architecture along with a universal sessions naming scheme for aiding better recall by people instead of the native dotted-decimal notation [72]. Another goal of my research is to enable location based session discovery. Let me justify this last goal with a simple motivation - Users in the city of Gainesville while searching for multicast channel about latest pizza deals in town would like to discover a session that relays content specific to this geographic region. In my research I would add capabilities in my multicast sessions directory architecture that will make possible geographically relevant searches.

## 4.2 Current Strategies and tools for session discovery

'sdr' [32] - Session Directory tool has been very popular among Mbone [3] enthusiasts. It has been used as a session management tool and is primarily based on LBL's Session Directory tool - 'sd'. It makes use of SDP - Session Description Protocol to announce the critical characteristics of multicast sessions such as channel address, port numbers, timing and resource information for remote hosts to join the conference. It uses a well known multicast channel address for propagating this information on the Internet. It also maintains a cache of other multicast sessions advertised elsewhere on the Internet through sdr. Based on the cache information maintained locally by each 'sdr' client, it tries to assign a new channel address to requesting multicast sessions in such a way as to reduce the address collisions among different sessions. It is the general consensus of the research community that even though 'sd' / 'sdr' was a great technology demonstrator; it is not suited to scale globally. All 'sdr' clients essentially maintain a flat hierarchy on the Internet which makes information dissemination among different 'sdr' clients a challenging prospect.

Andrew Swan and team [65] at Berkeley developed a completely decentralized sessions directory which they incorporated as part of their Light-Weight multimedia sessions framework. In this architecture they advertised the multimedia session's bindings at well known bootstrap address. They made use of Sessions Announcement Protocol for this purpose. To overcome the latency issue that plagues the multicast session directory architecture based on LBL's 'sd' application and SAP announcement bandwidth limitations, they proposed a tired announcement rate approach. The announcements agent under local scope announces session advertisements at a much higher frequency than traditional SAP clients. They also proposed splitting the traditional SAP client into two parts, one persistent server that runs SAP and caches all the network SAP announcements heard over a long period of time, and another ephemeral client

that contacts this persistent server for the cached list of available sessions.

In [60] Joaquim and team analyze the use / misuse of SDP as a session directory tool to advertise multimedia sessions. They argue that the session directory information that is embedded inside SDP fields is not standardized. Had it been standardized, these fields such as "media", "repeat time", "time active" etc. may be used for aggregating sessions which could then be used later to query for sessions. They propose that user should not be burdened with the task of browsing a flat structure; instead the task could be assigned to a server. This information could be presented to the user using either a well known multicast channel, or using several multicast channels or maybe using some specific server database. The article did not specify to what extent any of these goals were already implemented and what was the current status of their work.

Another attempt at making a distributed information discovery system in the Internet was Harvest [8]. The system was built using subsystems such as gatherers that were placed at the resource site, brokers which collected data from gatherers and incorporated in their resource index, brokers further consisted of index/search subsystems that were optimized for space and/or search time. The system also made use of replicators to replicate the data over multiple site in the Internet and object caches at critical sites to minimize the communication overload in the Internet. Instead of harvest being truly hierarchical, we believe it was replicated, Internet wide cache and unsuitable for multicast sessions discovery issue. It would be difficult to incorporate scoping and session lifetime requirements within their proposed framework. Also the dynamic nature of many sessions would result in cache instability.

Researchers at UCLA have proposed a scalable multicast information discovery graph (IDG) [64] based on the semantic description of stored as well as real time multimedia content over the Internet. This work is being done under Sematic Multicast project [1]. Even though their proposal provides for a hierarchical semantic directory, its not truly distributed. They have proposed making use of caching and soft state state refreshes to make their system more scalable and robust. In their approach, new multicast user may have to start at a well known root directory server which may create a bottleneck scenario as the number of sessions and users grow and even due to stale caches and periodic cache refreshes. The semantic hierarchy in their proposal currently is coarsely defined and may require significant rework in order to account for variety of multimedia sources uploaded online these days. The architecture does allow for much better search time compared to LBL's 'sd' tool but the bandwidth requirement grows linearly with the number of data sources. This is clearly a source of concern. Also multicast scoping requirements have been overlooked in the published work.

In [43], the authors proposed building an anycast SDP Proxy in order to give end-users immediate access to session announcements. They proposed an architecture along with HTTP style protocol format to access session information as well as create session entry at the remote SDP Proxy. This approach still suffers from the traditional 'sdr' issues of non-scalability in the face of large

number of sessions. Another issue with their approach is deploy-ability over SSM only networks.

In [53] the authors deal specifically with multicast session announcements over SSM networks. They propose a 2 tier hierarchy of dedicated session announcement servers (SAS). They propose to reduce the SAP related delay by increasing the allowed bandwidth limit of 4kbps to 50kbps for intra-domain announcements. SAS servers located in the backbone network of various ISP networks (level 2 SAS servers) act as relays and do not cache any announcement. Just increasing the intra-domain bandwidth seems to be a nearsighted solution at best. They state that whenever a new SAS level 2 server is added in the ISP's backbone network, the ISP finds out the address of other level 2 SAS servers and this information is configured into the new server. They failed to mention how this is achieved and whether the configuration is manual or automatic?

### 4.3 mDNS: DNS-aware multicast session directory architecture

We have tried to use well known network design principles in our proposed architecture. Notably among them are -

- **soft state** Maintaining soft states [59] in network components instead of hard coded parameters allows the network architecture to be more resilient against intermittent failures and also allows for self healing to occur once the failed components become online at some later point in time.
- **application layer framing** Application layer framing [11] allows for quick deployment of new and possibly novel network protocol using existing protocol stack as containers. The network core hardware does not usually require major upgrade.
- **auto configuration** Network Administrators are generally overworked people. If a network architecture is designed keeping auto configuration goal in mind, may help lessen the burden on network administrator. It also makes the architecture less prone to configuration errors and helps in keeping the network reasonably manageable with increasing network complexity.
- **hierarchical** Flat network architectures usually do not scale well with increasing network complexity. Network architectures are generally preferred to be hierarchical compared to flat structure.

We have proposed a globally scalable multicast session directory architecture which has been designed on the similar principle of domain name server (DNS) hierarchy. Here are the list of terminology that we will use in the rest of this chapter -

- $MSD_x^y$  - Multicast Session Directory (MSD) server 'y' in domain 'x'
- $MSD_x^d$  - Designated MSD Server in domain 'x'
- $DNS_x$  - Domain Name Server for domain 'x'
- $URS_x$  - URL registration server in domain 'x'

We will assume that each domain knows its DNS server address and DNS servers know about their parent DNS server's address which is a reasonable assumption to make. Also for global discover-ability of multicast sessions, we will assume that at least one MSD server coexists with the DNS server at each domain level. Failure of doing so may result in disconnected islands of sessions discovery zone in the global Internet (which may be desired sometimes in some cases). We propose to make an additional entry into DNS server's record table. We call it MCAST record and it contains such details as 'anycast IP address' [47] for MSD servers in that domain, globally scoped multicast channel details for establishing multicast group with designated MSD server of a particular domain and designated MSD servers in the children subnets and the globally scoped multicast channel details for establishing multicast group with designated MSD server of that particular domain and designated MSD server in the parent's domain. Additionally it may contain address of URS server in its domain. An example DNS MCAST record entry may look like -

```
@MCAST{
  ANYCAST=a.b.c.d
  CMCAST=233.[ASN Byte1].[ASN Byte2].XXX
  PMCAST=233.[ASN Byte1].[ASN Byte2].???
  PORT=pqrs
  URS=x.y.z.w
}
```

Note that the above example is just for illustrative purpose and actual DNS entry must follow the correct DNS entry standards. Notice that we have suggested the use of globally scoped addresses from the GLOP [48] address range. These address are assigned by the ISPs and the domain owners / administrators must apply for these address from their ISPs. Also ASN denotes the AS (Autonomous System) Number. The system has been proposed to co-exist with our HOMA [35] multicast address allocation and management scheme.

### 4.3.1 mDNS hierarchy construction

We will describe the hierarchy of our scheme through two example networks, namely .edu hierarchy network and a general hypothetical ISP network. Our scheme will work with any network organization as long as our initial DNS server and MSD server assumptions remain valid. A typical example is shown in figure 4.1.

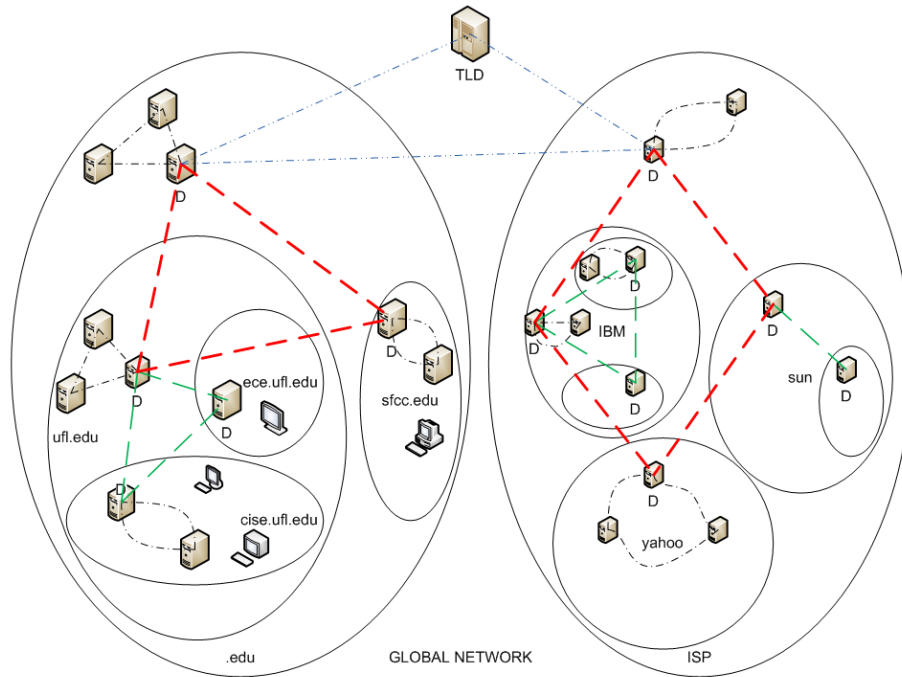


Figure 4.1: a typical MSD hierarchy

Now we try to explain the proposed mDNS hierarchical architecture in some detail now. Let us concentrate on just the .edu hierarchy. Under the .edu domain we have two university networks. Under UF network we have 2 sub-domain namely CISE and ECE, both of these are in themselves independent administratively scoped multicast domains. Further UF is also administratively scoped domain. CISE and ECE maintain their own DNS server whose parent DNS server will be the UF DNS server. UF DNS server is a child node of the .edu TLD DNS Server. UF maintains multiple MSD servers, all of which subscribe to a fixed (possibly IANA assigned) administratively scoped multicast channel. From here on we will refer to this channel as MSD-LOCAL-MCAST channel. CISE and ECE also maintain their own sets of MSD servers, again those subscribe to MSD-LOCAL-MCAST channel. Since this channel is administratively scoped channel, if the edge routers are properly configured then there should be no cross-talk among these channels.

If there are multiple MSD servers maintained under a domain, a designated MSD server is chosen based on some leader election algorithm [49] [19]. In figure 4.2, these are marked with the letter 'D' next to them. The designated MSD servers joins two globally scoped multicast channels namely those specified by CMCAST and PMCAST entries in the MCAST record of the DNS server in their domain. If any of these two entry is NULL that particular channel is not

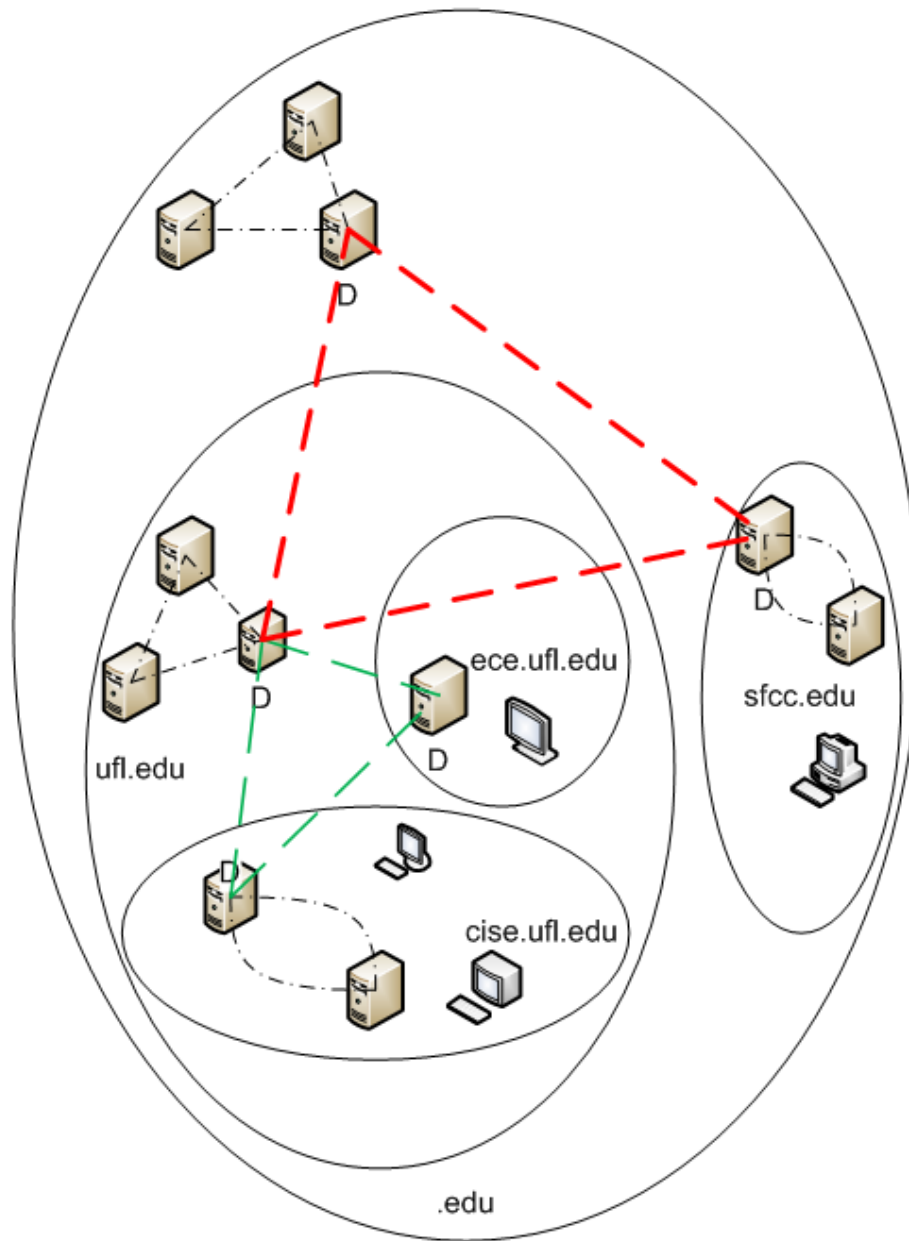


Figure 4.2: an example .edu hierarchy

subscribed to. It is important to note that all the MSD servers in any particular domain (excluding children subdomains) are anycasted at the anycast IP address specified in the DNS server's MCAST record.

---

**MSD Server's Base Algorithm**

---

```

join MSD-LOCAL-MCAST channel
initiate leader election on this channel
if elected leader then
    query local DNS server for PMCAST and CMCAST
    join PMCAST and CMCAST channels
end if

```

---

This is how MSD servers' hierarchical structure is established in mDNS architecture. It is quite easy to see that the hierarchy will exist as long as the initial two assumptions are satisfied in any network domain hierarchy. We chose to present our example for a .edu domain just for illustration purposes.

### 4.3.2 Session registration in mDNS

mDNS architecture has been designed to co-exists with our HOMA [35] proposal. We would assume that an application in any domain before transmitting multicast traffic (in any scope) has an appropriate multicast channel address allotted to it. In this paper we have not provided details on the internal database maintained at MSD servers but let us assume further that MSD servers are capable of registering channel keywords along with channel details on behalf of multicast applications. It is the responsibility of the session creator to provide upto maximum of 10 keywords correctly describing the session. These keywords will aid in session discovery process described later.

An application under mDNS architecture, after it has acquired a valid channel address, will execute the following pseudocode -

---

**Session registration pseudocode**

---

```

contact local DNS Server to find URS address
if URS server exist then
    request URS server to register a channel descriptive 'keyword'
    if keyword registration at URS successful then
        done.
    else
        pick another keyword
        try again
    end if
end if
initiate channel registration request on the MSD-LOCAL-MCAST
register the channel details with MSD server
provide list of keywords (max upto 10)
provide session duration and operating times
provide URS registered 'keyword' if any

```

---

The MSD server will correctly identify the scope of the multicast channel based on the channel address being registered with it.

### 4.3.3 mDNS search operation

Multicast sessions in mDNS can be searched using session keywords. Sessions can also be accessed directly if the session creator successfully registered a valid 'keyword' with the domain's URS server. We have proposed a simple URL scheme in order to facilitate multicast channel details access in order for the remote host to subscribe to that particular channel on the Internet under mDNS architecture. We believe that having an URL scheme will greatly enhance the usability of IP-multicast and would alleviate it to its fullest potential rapidly.

mDNS URL is constructed using the following syntax -

`<protocol>://<domain URL>/<URS Keyword>`

In the above URL scheme, protocol could be the method the remote user will use to communicate with the MSD server defined in the DNS MCAST record. It could be HTTP or similar protocol. The domain URL helps resolve the MSD server located in the multicast session creator domain. It must begin with 'mcast' to specify the MSD server. For example, let us assume that under cise.ufl.edu sub-domain, we have created a globally scoped multicast channel that multicasts information about gators. Further assume we have successfully registered the keyword 'gators' with the URS server located in CISE domain. Someone else can then directly access the multicast session details to subscribe to the gator channel using this URL -

`http://mcast.cise.ufl.edu/gators`

The search is done in a very similar fashion, under mDNS scheme, user can do domain specific search or general global search. If the user wishes to do domain specific search, he can specify the specific domain to search for a particular keyword in the same fashion as specified above but instead now using the qualifiers "search" and "keyword" in the URL. For example, if the user wishes to find sessions with keyword 'gators' under cise.ufl.edu domain, they can do so by using the string -

`mcast.cise.ufl.edu/search=all&keyword=gators`

The end users' multicast search application would resolve the mcast.cise.ufl.edu anycast address and would connect to one of the MSD servers located in the cise.ufl.edu domain. MSD servers perform database search against the keyword "gators" and every content type. Content type could be audio, video, whiteboard, etc. to name a few possible types. The search is performed in a top-down hierarchy starting from the domain specified and percolating down to all sub-domains (if any). In our present scheme, the search results are returned from MSD servers at each level directly to the requesting client. There are few other delivery schemes we are considering that we will describe in future research section. The MSD servers are smart in the sense that they recognize whether the query comes from a host inside its domain or outside the domain. If the querist is located outside the MSD domain, the MSD returns only those search results

that are globally scoped channels. Administratively scoped sessions are only returned as part of query if the querist resides in the same domain.

Additionally any user can choose to perform a global keyword search. In order to do this, the client must contact the local MSD server co-located at the same zone as its DNS server. Global search is propagated by the MSD servers on both PMCAST and CMCAST channels and its own MSD-LOCAL-MCAST channel thereby spreading the search to both child domains as well as parent domain. The propagated search request contains uniquely identifying string that allows the originator MSD to kill the search if the same query id comes again to the same MSD server. Also the search query based on the identifying sting is never re-propagated on the same channel on which it was received. It is easy to see that in the proposed mDNS architecture, this uniquely identifying search id is generated by the designated MSD server at each level and only when the search query was received on the MSD-LOCAL-MCAST channel.

Here is the pseudocode for search which is executed at each MSD server -

---

MSD pseudocode for search

---

```

received search query on subscribed channels
search sessions internal database for search match
if multicast sessions found then
  if querist resides in another domain then
    return only globally scoped session details (if any)
  else
    return every result found directly to the querist
  end if
end if
if MSD server is a designated MSD server then
  if search unique ID is missing then
    generate unique search ID
  end if
  if query request was not received on CMCAST channel then
    propagate search on CMCAST channel
  end if
  if search is global in nature then
    if query request was not received on PMCAST channel then
      propagate search on PMCAST channel
    end if
  end if
  if search query has self generated previous ID then
    drop search request
  end if
end if

```

---

#### 4.3.4 search example

Lets take an example hierarchy to see how search operation is performed in mDNS architecture (see figure 4.3).

Assume the search is initiated by an end host at the only subnet in the SUN network. Also assume that the search is global in nature. The search goes to the only MSD server in the subnet which is also the designated MSD server. The MSD server searches its local sessions database and returns all the hits to the requesting host's application.

The designated MSD server generates a unique search query ID and propagates the search to both CMCAST and PMCAST channels. In this example network, CMCAST channel does not exist for the SUN internal subnet, so the search is propagated only on PMCAST channel (green dashed line). It reaches the SUN network wide MSD server, this outer SUN MSD server searches its internal sessions database. Since the search came from a subnet host and not from a coexisting host, it only returns those results that are global in scope. It propagates the search on the PMCAST channel (red dashed line) for the SUN outer MSD server.

Now the search query reaches the MSD servers in the IBM, Yahoo and the ISP's TLD MSD server. This is how the query eventually propagates to all MSD servers in the mDNS architecture.

#### 4.3.5 High level analysis of mDNS

We believe that mDNS has features to take IP-Multicast to the next level of deployment in the general consumer network. Together with HOMA [35] architecture, mDNS has true potential to make multicast session discovery and deployment seamless and consumer friendly. mDNS URL scheme presented in section 3 would make bookmarking of popular multicast sessions feasible and equally userfriendly as webpage bookmarks in the current Internet.

But mDNS still being in early development stages has lots of drawbacks in its current form. It is specially vulnerable to DoS attacks. Also each global query has the potential to activate every MSD server deployed under mDNS scheme. Further more direct query results transmission to the querist from MSD servers creates the possibility of DDoS [29] attack on some remote host on the Internet using IP spoofing attacks.

There are several immediate benefits of the mDNS scheme. First among many is the database space savings. In LBL's sd and later sdr software implementation, possibly every sdr client may cache the sessions detail in the software's local cache. This makes the sdr scheme particularly not scalable. In mDNS scheme only the leaf node MSD server in the hierarchy maintains the multicast session database. Also the receiver application does not have to wait 10s of minutes to discover a session because of SDP global bandwidth usage restriction. We conjecture that session search and discovery in mDNS scheme should be many orders of degree faster than current schemes. Benefits of URLs have already been discussed earlier.

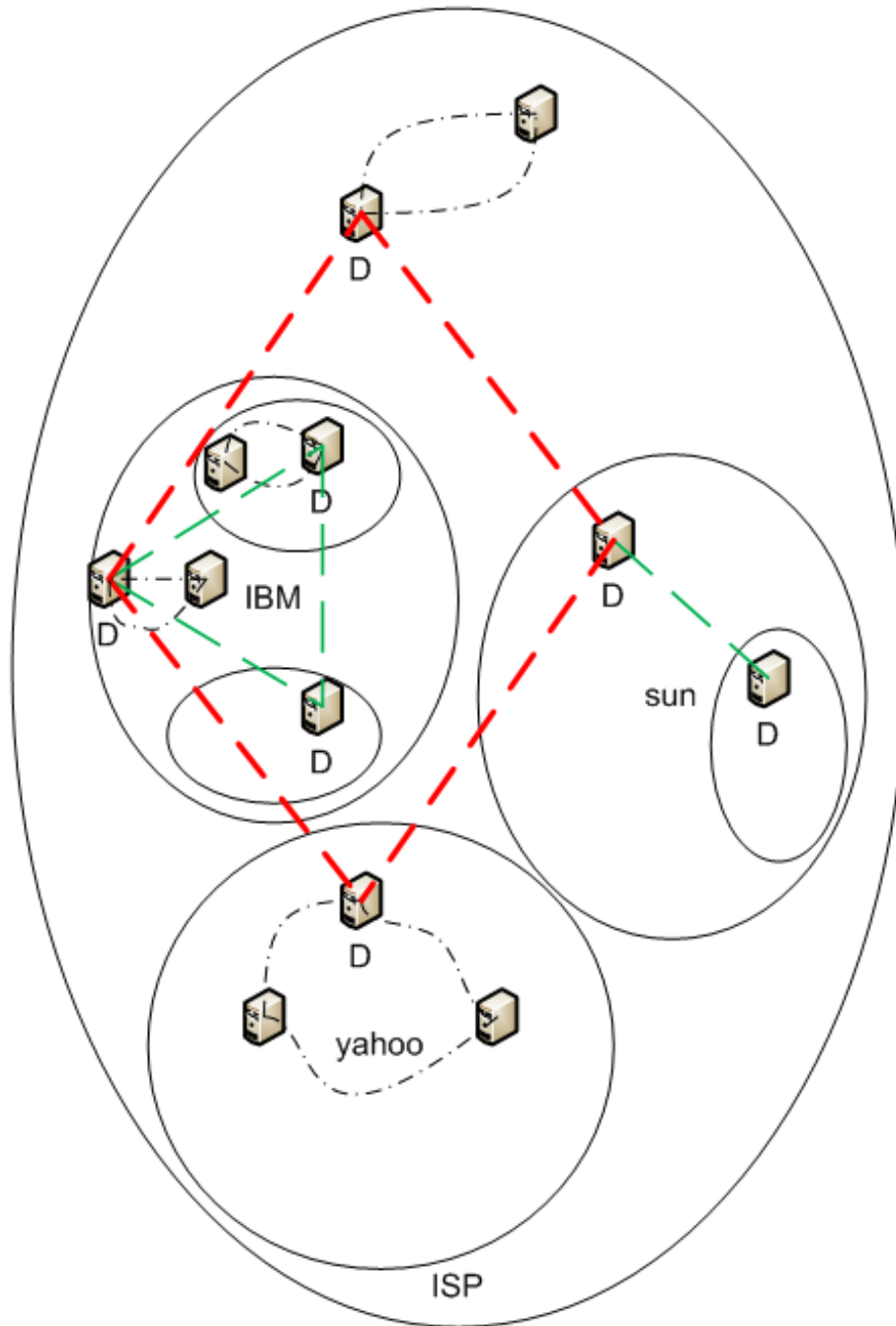


Figure 4.3: an general mDNS hierarchy

## Chapter 5

# What remains to be done ...

### 5.1 System Simuations

I am currently looking into a valid Internet simulation models to implement. Paxon and Floyd [57] presented arguments on why is it very difficult to simulate the Internet. They proposed that simulation models based on system invariants such as network user "session" arrivals and session durations would be better models that would remain valid for relatively longer period of time in future.

Network sessions arrivals could be described effectively using Poisson processes. In our case this would correspond to multicast applications trying to get a suitable channel address from HOMA servers or session queries arrival in the mDNS proposal in earlier chapters. It was suggested that in order to better simulate the behavior in the Internet, one should periodically change the Poisson arrival rate preferably at an hourly rate. This rate adjustments on an hourly rate further depends on another invariant which is the daily and the weekly internet traffic patterns.

They further suggested that the Internet sessions duration could be characterized as log-normal. In our work this would correspond to the session addresses leasing period which the multicast sessions would negotiate with the HOMA servers. Whether the lease negotiated would be approved by the HOMA nodes is entirely different issue that is strongly effected by the address chunk validity period at that node.

I am currently looking into other models such as ON-OFF network model, network activity models based on heavy-tail distributions (Pareto distribution with shape parameter  $\alpha < 2$ ). Once the simulation model has been finalized I propose to compare my results with few other models that have been proposed including MASC [41], sdr [32].

## 5.2 Simulation Parameter Space Selection

I have yet to decide on simulation parameter space to observe. Doing a simulation based on any one parameter would be a very short sighted approach for the rapidly evolving and extremely heterogenous Internet today. An extensive parameter space study which are relevant to my simulation has to be done.

As mentioned in HOMA proposal, the algorithm parameters such as  $p$ ,  $p'$  and  $\pi$  have to be determined. It would be somewhat interesting to find the stability range for these parameters with respect to HOMA design. I would also vary the algorithm cycle time for HOMA and try to find out how does this parameter variation effects the stability of HOMA distributed algorithm (if at all). Address space utilization factor in HOMA scheme compared to other proposed schemes is also to be studied.

For mDNS architecture, search effectiveness and speed, network stress factor, control overhead would be some of the interesting parameters to look into during the simulation phase of my work. In addition I will do database space comparisons between mDNS and flat-structured 'sdr'.

## 5.3 Protocol Design

Network protocol design for both HOMA and mDNS proposals have to be done. Future readiness of protocol fields structure design will be paramount in my efforts. I would provide as part of my final dissertation a complete protocol structure for both HOMA inter-node communication and mDNS search queries and results dissemination framework. I would also provide complete protocol for inter-node control and maintenance efforts.

## 5.4 mDNS - Ongoing / Future Work (Time Permitting)

Time permitting we will consider subscription based approach where each MSD client in addition to maintaining locally residing session source database, also maintains keywords subscription level by hosts in its subnet / domain. We will try to come up with a threshold based system to propagate the keyword subscription up the mDNS hierarchy.

One of the major concerns we have is activation of every MSD server in mDNS scheme for every global search. This could prove to be very taxing on MSD servers. We will try to look into intelligent cache placements along with smart caching techniques in order to reduce the possible workload on MSD servers.

Security remains a major concern in any distributed deployment in the Internet today. mDNS in its current form is vulnerable to all kinds of attacks. We may also study into the possible nature of threat to mDNS and possible solution to thwart those threats.

We may also look into the design of the mDNS database record structure.

## **5.5 Framework for HOMA and mDNS integration**

The design of HOMA and mDNS architectures have been done keeping their integration in the mind since the very beginning. We strongly believe that they can be co-located on the same servers. Nevertheless I will study their integration issue in some details and would incorporate my findings in the final dissertation.

# Appendix A

## List of publications

- Piyush Harsh and Richard Newman - "mDNS - A Proposal for Hierarchical Multicast Session Directory Architecture", submitted to ICOMP 2008.
- Piyush Harsh and Richard Newman - "An overlay solution to IP-Multicast address collision prevention", proceeding of IASTED EuroIMSA 2008, March 2008.
- Piyush Harsh and Richard Newman - "Usability and Acceptance of UF-IBA, an Image-Based Authentication System", proceedings of IEEE ICCST 2007, October 2007.
- Richard E. Newman, Piyush Harsh and Prashant Jayaraman - "Security Analysis of and Proposal for Image-based authentication", proceedings of IEEE ICCST 2005, p. 141, October 2005.

# Bibliography

- [1] Semantic multicast project. <http://www.wins.hrl.com/projects/semcast/>.
- [2] *OSI IS-IS Intra-domain Routing Protocol*, 1990. Internet Engineering Task Force, RFC 1142.
- [3] ALMEROTH, K. The evolution of multicast: from the mbone to interdomain multicast to internet2 deployment. *Network, IEEE 14*, 1 (Jan/Feb 2000), 10–20.
- [4] BALLARDIE, T. Core based tree (cbt) multicast – architectural overview and specification, 1995.
- [5] BANDEL, D. A. Cidr. *Linux J.*, 2.
- [6] BATES, T., CHANDRA, R., KATZ, D., AND REKHTER, Y. Multiprotocol extensions for bgp, 1998.
- [7] BHATTACHARYYA, S. *An Overview of Source-Specific Multicast (SSM)*, July 2003. Internet Engineering Task Force, RFC 3569.
- [8] BOWMAN, C. M., DANZIG, P. B., HARDY, D. R., MANBER, U., AND SCHWARTZ, M. F. The harvest information discovery and access system. *Computer Networks and ISDN Systems 28*, 1–2 (December 1995), 119–125.
- [9] CAIN, B., DEERING, S., KOUVELAS, I., FENNER, B., AND THYAGARAJAN, A. *Internet Group Management Protocol, Version 3*, 2002. Internet Engineering Task Force, RFC 3376.
- [10] CASNER, S., AND THYAGARAJAN, A. *mtrace(8): Tool to Print Multicast Path Form a Source to a Receiver*. UNIX manual command.
- [11] CLARK, D. D., AND TENNENHOUSE, D. L. Architectural considerations for a new generation of protocols. *SIGCOMM Comput. Commun. Rev.* 20, 4 (1990), 200–208.
- [12] COLTUN, R., FERGUSON, D., AND MOY, J. *OSPF for IPv6*, 1999. Internet Engineering Task Force, RFC 2740.

- [13] DALAL, Y. K., AND METCALFE, R. M. Reverse path forwarding of broadcast packets. *Commun. ACM* 21, 12 (1978), 1040–1048.
- [14] DEERING, S. *Host Extensions for IP Multicasting*, August 1989. Internet Engineering Task Force, RFC 1112, STD 5.
- [15] DEERING, S., ESTRIN, D. L., FARINACCI, D., JACOBSON, V., LIU, C.-G., AND WEI, L. The PIM architecture for wide-area multicast routing. *IEEE/ACM Transactions on Networking* 4, 2 (1996), 153–162.
- [16] DEERING, S., FENNER, B., ESTRIN, D., HELMY, A., FARINACCI, D., WEI, L., HANDLEY, M., JACOBSON, V., AND THALER, D. Hierarchical pim-sm architecture for inter-domain multicast routing, 1995.
- [17] DEERING, S., AND HINDEN, R. *Internet Protocol, Version 6 (IPv6) Specification*, 1995. Internet Engineering Task Force, RFC 1883.
- [18] DEERING, S. E. Multicast routing in internetworks and extended lans. In *SIGCOMM '88: Symposium proceedings on Communications architectures and protocols* (New York, NY, USA, 1988), ACM, pp. 55–64.
- [19] DOLEV, S., ISRAELI, A., AND MORAN, S. Uniform dynamic self-stabilizing leader election. *IEEE Trans. Parallel Distrib. Syst.* 8, 4 (1997), 424–440.
- [20] DOVAL, D., AND O'MAHONY, D. Overlay networks: A scalable alternative for p2p. *Internet Computing, IEEE* 7, 4 (July-Aug. 2003), 79–82.
- [21] DROMS, R. Automated configuration of tcp/ip with dhcp. *Internet Computing, IEEE* 3, 4 (Jul/Aug 1999), 45–53.
- [22] EDWARDS, B. M., AND WRIGHT, B. *Interdomain Multicast Routing: Practical Juniper Networks and Cisco Systems Solutions*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002. Foreword By-John W. Stewart.
- [23] ESTRIN, D. Protocol independent multicast-sparse mode (PIM-SM): Motivation and architecture. draft-ietf-idmr-pim-arch-01.ps, Internet Draft.
- [24] ESTRIN, D., FARINACCI, D., HELMY, A., JACOBSON, V., AND WEI, L. Protocol independent multicast (pim) dense mode protocol specification, 1996.
- [25] ESTRIN, D., HANDLEY, M., HELMY, A., HUANG, P., AND THALER, D. A dynamic bootstrap mechanism for rendezvous-based multicast routing. *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* 3 (21-25 Mar 1999), 1090–1098 vol.3.
- [26] FARINACCI, D., AND WEI, L. *Auto-RP: Automatic discovery of Group-to-RP mappings for IP multicast*. CISCO Press, Sept 9, 1998.

- [27] FENNER, W. *Internet Group Management Protocol, Version 2*, 1997. Internet Engineering Task Force, RFC 2236.
- [28] FERGUSON, P., AND SENIE, D. *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*, 1998. Internet Engineering Task Force, RFC 2267.
- [29] GARBER, L. Denial-of-service attacks rip the internet. *Computer* 33, 4 (Apr 2000), 12–17.
- [30] GRAHAM, B. *TCP/IP Addressing: Designing and Optimizing Your IP Addressing Scheme, Second Edition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.
- [31] HANDLEY, M. Session directories and scalable internet multicast address allocation. *SIGCOMM Comput. Commun. Rev.* 28, 4 (1998), 105–116.
- [32] HANDLEY, M. The sdr session directory: An mbone conference scheduling and booking system.
- [33] HANDLEY, M., AND JACOBSON, V. Sdp: Session description protocol, 1997.
- [34] HANDLEY, M., PERKINS, C., AND WHELAN, E. *Session Announcement Protocol*, 2000. Internet Engineering Task Force, RFC 2974.
- [35] HARSH, P., AND NEWMAN, R. E. An overlay solution to ip-multicast address collision prevention. *IASTED EuroIMSA Conference Proceedings* (Mar 17–19 2008).
- [36] HOLBROOK, H., AND CAIN, B. *Source-Specific Multicast for IP*, October 2003. Internet Engineering Task Force, Work in Progress.
- [37] HUANG, Y., AND PULLEN, J. Countering denial-of-service attacks using congestion triggered packet sampling and filtering. *Computer Communications and Networks, 2001. Proceedings. Tenth International Conference on* (2001), 490–494.
- [38] JACOBSON, V. Multimedia conferencing on the internet. *SIGCOMM* (Aug 1994).
- [39] KATABI, D. The use of ip-anycast for building efficient multicast trees. *Global Telecommunications Conference, 1999. GLOBECOM '99 3* (1999), 1679–1688 vol.3.
- [40] KATABI, D., AND WROCLAWSKI, J. A framework for scalable global ip-anycast (gia). *SIGCOMM Comput. Commun. Rev.* 31, 2 supplement (2001), 186–219.

- [41] KUMAR, S., RADOSLAVOV, P., THALER, D., ALAETTINOĞLU, C., ESTRIN, D., AND HANDLEY, M. The masc/bgmp architecture for inter-domain multicast routing. *SIGCOMM Comput. Commun. Rev.* 28, 4 (1998), 93–104.
- [42] KUROSE, J. F., AND ROSS, K. W. *Computer Networking: A Top-Down Approach Featuring the Internet*. Pearson Benjamin Cummings, 2004.
- [43] LIEFOOGHE, P., AND GOOSENS, M. The next generation ip multicast session directory. *SCI, Orlando FL* (July 2003).
- [44] LIVINGSTON, M., LO, V. M., ZAPPALA, D., AND WINDISCH, K. J. Cyclic block allocation: A new scheme for hierarchical multicast address allocation. In *Networked Group Communication* (1999), pp. 216–234.
- [45] LO, V., ZAPPALA, D., AND GAUTHIERDICKY, C. A theoretical framework for multicast address allocation, 2002.
- [46] MALKIN, G. *RIP Version 2 - Carrying Additional Information*, 1994. Internet Engineering Task Force, RFC 1723.
- [47] METZ, C. Ip anycast point-to-(any) point communication. *Internet Computing, IEEE* 6, 2 (Mar/Apr 2002), 94–98.
- [48] MEYER, D., AND LOTHBERG, P. *GLOP Addressing in 233/8*, 2001. Internet Engineering Task Force, RFC 3180.
- [49] MIRAKHORLI, M., SHARIFLOO, A. A., AND ABBASPOUR, M. A novel method for leader election algorithm. In *CIT '07: Proceedings of the 7th IEEE International Conference on Computer and Information Technology* (Washington, DC, USA, 2007), IEEE Computer Society, pp. 452–456.
- [50] MOCKAPETRIS, P., AND DUNLAP, K. J. Development of the domain name system. *SIGCOMM Comput. Commun. Rev.* 18, 4 (1988), 123–133.
- [51] MOY, J. Multicast extensions to OSPF. Tech. rep., 1991.
- [52] MOY, J. *OSPF Version 2*, 1998. Internet Engineering Task Force, RFC 2328.
- [53] NAMBURI, P., AND SARAC, K. Multicast session announcements on top of ssm. *Communications, 2004 IEEE International Conference on* 3 (20-24 June 2004), 1446–1450 Vol.3.
- [54] OIKARINEN, J., AND REED, D. *Internet Relay Chat Protocol*, 1993. Internet Engineering Task Force, RFC 1459.
- [55] PARK, K., AND LEE, H. On the effectiveness of route-based packet filtering for distributed dos attack prevention in power-law internets. *SIGCOMM Comput. Commun. Rev.* 31, 4 (2001), 15–26.

- [56] PARTRIDGE, C., MENDEZ, T., AND MILLIKEN, W. *Host Anycasting Service*, 1993. Internet Engineering Task Force, RFC 1546.
- [57] PAXSON, V., AND FLOYD, S. Why we don't know how to simulate the internet. *Simulation Conference, 1997., Proceedings of the 1997 Winter (7-10 Dec 1997)*, 1037–1044.
- [58] PEJHAN, S., ELEFThERiADiS, A., AND ANASTASSIOU, D. Distributed multicast address management in the global internet. *Selected Areas in Communications, IEEE Journal on 13*, 8 (Oct 1995), 1445–1456.
- [59] RAMAN, S., AND MCCANNE, S. A model, analysis, and protocol framework for soft state-based communication. *SIGCOMM Comput. Commun. Rev. 29*, 4 (1999), 15–25.
- [60] SANTOS, A., MACEDO, J., AND FREITAS, V. Towards multicast session directory services.
- [61] SHOCH, J. F., DALAL, Y. K., REDELL, D. D., AND CRANE, R. C. The ethernet. In *Proceedings on Local Area Networks: An Advanced Course* (London, UK, 1985), Springer-Verlag, pp. 1–35.
- [62] STALLINGS, W. *Data and Computer Communications*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.
- [63] STALLINGS, W. *High Speed Networks and Internets: Performance and Quality of Service*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.
- [64] STURTEVANT, N., TANG, N., AND ZHANG, L. The information discovery graph: towards a scalable multimedia resource directory. *Internet Applications, 1999. IEEE Workshop on* (Aug 1999), 72–79.
- [65] SWAN, A., MCCANNE, S., AND ROWE, L. A. Layered transmission and caching for the multicast session directory service. In *ACM Multimedia* (1998), pp. 119–128.
- [66] THYAGARAJAN, A. S., AND DEERING, S. E. Hierarchical distance-vector multicast routing for the mbone. In *SIGCOMM '95: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication* (New York, NY, USA, 1995), ACM, pp. 60–66.
- [67] TSUCHIYA, P. Efficient utilization of two-level hierarchical addresses. *Global Telecommunications Conference, 1992. Conference Record., GLOBECOM '92. Communication for Global Users., IEEE* (6-9 Dec 1992), 1016–1021 vol.2.
- [68] WAITZMAN, D., PARTRIDGE, C., AND DEERING, S. E. *Distance Vector Multicast Routing Protocol*, 1988. Internet Engineering Task Force, RFC1075.

- [69] WILLIAMSON, B. *Developing IP Multicast Networks*. Cisco Press, 1999.
- [70] WOLF, L. C., AND HERRTWICH, R. G. The system architecture of the heidelberg transport system. *SIGOPS Oper. Syst. Rev.* 28, 2 (1994), 51–64.
- [71] WOODBURN, R. A., AND MILLS, D. L. *Scheme for an internet encapsulation protocol: Version 1*, 1991. Internet Engineering Task Force, RFC 1241.
- [72] WRIGHT, R. *IP Routing Primer*. Macmillan Technical Publishing, 1998.
- [73] ZAPPALA, D., LO, V., AND GAUTHIERDICKEY, C. The multicast address allocation problem: Theory and practice. *Special Issue of Computer Networks* (2004).