DHT implementation on PIM-SM for an Inter-domain Multicast Routing

Nguyen Hung Long

Faculty of Environment and Information Studies

Keio University

5322 Endo Fujisawa Kanagawa 252-0882 JAPAN

Submitted in partial fulfillment of the requirements for the degree of Bachelor

Advisors:

Professor Hideyuki Tokuda

Professor Jun Murai

Associate Professor Hiroyuki Kusumoto

Professor Osamu Nakamura

Associate Professor Kazunori Takashio

Assistant Professor Noriyuki Shigechika

Assistant Professor Rodney D. Van Meter III

Associate Professor Keisuke Uehara

Associate Professor Jin Mitsugi

Lecturer Jin Nakazawa

Professor Keiji Takeda

Abstract of Bachelor Thesis

DHT Implementation on PIM-SM for an Inter-domain Multicast Routing

This thesis proposes a design and implementation using DHT (Distributed Hash Table) on PIM-SM (Protocol Independent Multicast - Sparse Mode) as a replacement to MSDP (Multicast Source Discovery Protocol) for inter-domain multicast routing. This implementation aims to solve multicast source discovery problem in inter-domain multicast as well as as reduce the scalability issue which lies in MSDP. With ability to scale extremely large number of nodes as well as provide efficient lookup service, instead of flooding the Source Active (SA) messages to inform the existence of Multicast sources between domains like MSDP, this research use DHT implementation as a sources-lookup operation which can lower network bandwidth consuming as well as improve scalability and robustness.

With comparison from calculation and evaluation result, we expect that with this implementation, the number of communicating messages between inter-domain routers would be reduced significantly comparing to MSDP protocol in large-scale network.

Keywords:
Multicast routing, Inter-domain, MSDP, DHT.

Nguyen Hung Long

Faculty of Environment and Information Studies
Keio University

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The development of Internet has enhanced many new services and applications as well as Internet users have been seen increasing in a very fast pace in only two decades. Nowadays, with the high growth of users in multimedia content communications like video conference or gaming online, who want to simultaneously share data and information, needs of applications which can provide data transfer with minimizing bandwidth requirements has been generated.

Those applications need a network-technique, that can send packets from one or many sources to a group of recipients in the most efficient strategy. IP Multicast is referred to be that technique. IP Multicast applications can be characterized into 3 general types for many purposes[1]:

One-to-Many (1toM): A single host sending to two or more (n) receivers. Examples include: push media content distribution of news, sports, weather; security monitoring; distribution of stock market prices, manufacturing process information, schedule announcements, keys, and network times; and file distribution and caching.

Many-to-Many (MtoM): Any number of hosts sending to the same multicast group address, as well as receiving from it. It can be applied for interactive distance learning, interactive multi-player games, jam sessions, multimedia teleconferencing (voice/video phones and whiteboards), chat groups, shared editing and collaboration tools, parallel computing, as well as distributed interactive simulations.

Many-to-One (Mto1): Any number of receivers sending data back to a (source) sender via unicast or multicast. Examples include polling and data collection, resource and service location discovery (Anycast), on-line auctions.

With an IP Multicast-enabled network available, some unique and powerful applications and application services are possible.

1.1.1 ASM and SSM

The original multicast, which is described in RFC 112, supported both many-to-many and one-to-many models. This is known as Any-Source Multicast (ASM) because ASM allowed one or many sources for a multicast group's traffic. An ASM network must be able to determine the locations of all sources, no matter where the sources might be located in the network.

However, due to the address allocation problem and lack of access control, Source-Specific Multicast (SSM) architecture has been designed. Ignoring the many-to-many model, SSM focus on the one-to-many source-specific multicast model, and multicast applications, such as television channel distribution over the Internet, might be brought to the Internet much more quickly and efficiently than if full ASM functionality were required of the network.

1.1.2 Multicast operation

Multicast network aims to provide efficient communication services for applications that send the same data to multiple recipients, without incurring network overloads. Hence, at each router, only one copy of an incoming multicast packet is sent per link, rather than sending one copy of the packet per number of receivers accessed via that link[3]. It's different from broadcast operation, that it only distributes the packet stream to recipient groups of hosts, not to all hosts. And it also differs from unicast operation that it delivers a single stream of information to a large group, with only one copy of the packet stream traveling over any individual link. This can reduce the traffic overload as well as minimize the usage of network bandwidth.

Figure 1.1 and 1.2 show the difference between unicast and multicast data-transfer from one data-source to receivers through routers.

In IP Multicast, multicast sources only need to transfer data on a single stream to a multicast group IP address, which represents a group of receivers. And receivers are to join the same multicast group addresses to receive that data. The routing between multicast sources and group of receivers are controlled by multicast routers, in which the data is only forwarded once on single through any individual link over the network.

1.1.3 Basic Concept of Multicast Routing

Stephen Deering described the standard multicast model for IP networks[4] as follows:

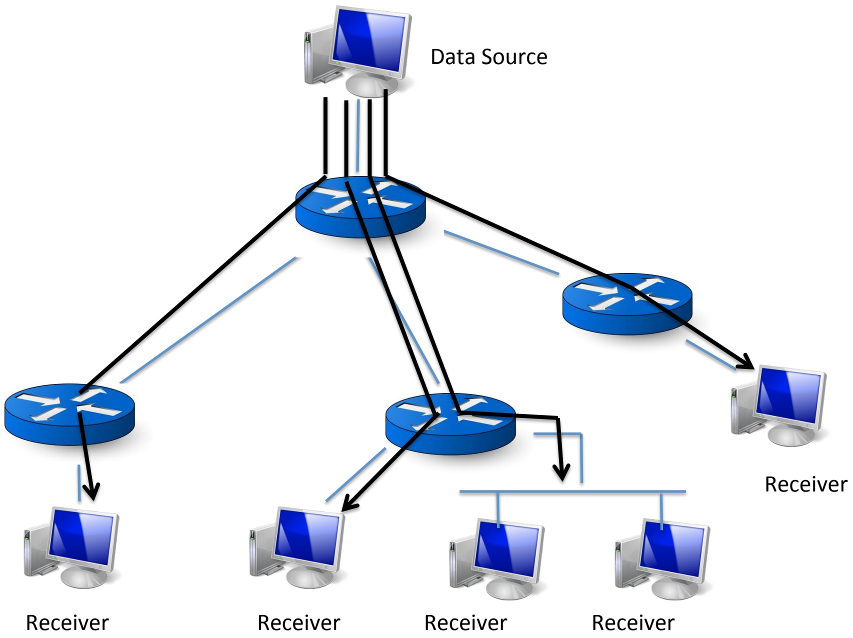


Figure 1.1: Unicast data-transfer

- IP-style semantics: A source can send multicast packets at any time, with no need to register or to schedule transmission. IP multicast is based on User Datagram Protocol (UDP) (not TCP), so packets are delivered using a best-effort policy.

- Open groups: Sources only need to know a multicast address. They do not need to know group membership, and they do not need to be a member of the multicast group to which they are sending. A group can have any number of sources.

- Dynamic groups: Multicast group members can join or leave a multicast group at will. There is no need to register, synchronize, or negotiate with a centralized group management entity. The standard IP multicast model is an end-system specification and does not discuss requirements for how the network should perform routing. The model also does not specify any mechanisms for providing quality of service, security, or address allocation.

1.2 Challenge and Research Goal

IP Multicast was first introduced in 1988, but after 22 years, Multicast network has still seen very slow deployment on the Internet compared to its expectation.

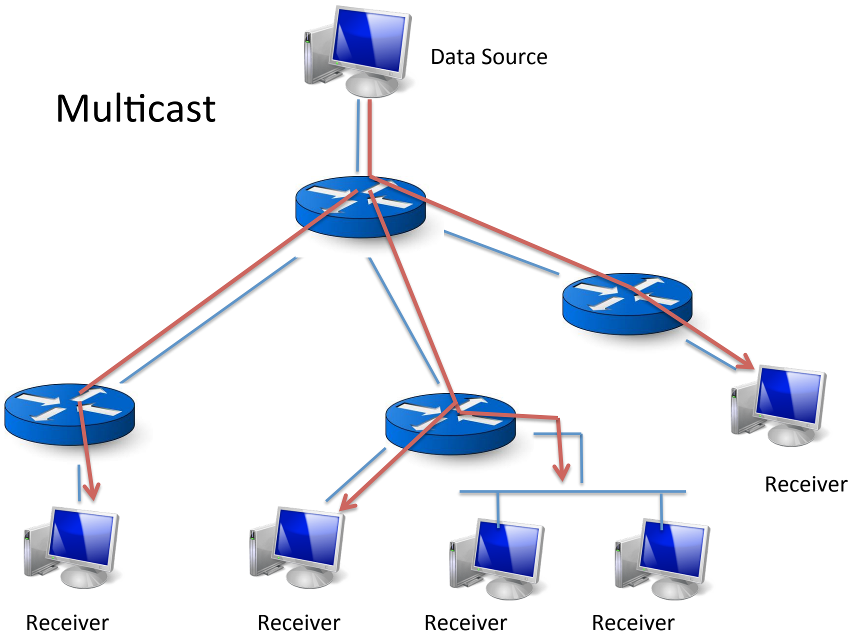


Figure 1.2: Multicast data transfer

One of the reasons is that multicast routing is quite complex and not totally completed yet. A multicast routing protocol should be efficient, scalable, robust, use minimal network overhead, consume minimal memory resources, inter-operate with other multicast routing protocols, and be easy to implement[5], which is not easy to achieve. Especially IP Multicast between domains will get more troubles because of difference policies between autonomous systems (ASes). Inter-domain routing involves the use of resources in autonomously administered domains, so the routing policy constraints of such domains need to be accommodated [6].

This thesis describes scalability problem, which lies in Multicast Source Discovery Protocol (MSDP) - one of Inter-domain multicast routing protocols, and proposes a solution using implementation of DHT network to this problem.

1.3 Structure of Thesis

The rest of this thesis is organized as follows.

Chapter 2 describes background materials in the area of multicast routing which include intra-domain and inter-domain multicast routing protocols as some related works and also discusses the problem statement which lies in MSDP's scalability. Chapter 3 describes about this thesis's approach which use DHT network as solution. In Chapter 4, we present the system design using DHT and Chapter 5 is implementation with DHT. Chapter 6 discusses results and evaluation from experimental. Finally in Chapter 7, we present conclusions and future work.

Chapter 2

Multicast Routing

2.1 Intra-domain multicast routing protocols

This section provides a brief description of the intra-domain multicast routing protocols. Based on routing algorithms, intra-domain multicast routing protocols can be classified into two categories: source-based and core based multicast tree.

The source-based multicast core tree approach: A tree rooted at a source node is constructed and connected to every member in the multicast group. Data packets originating from the source node are sent to all the destination nodes via the links of a multicast tree[7]. Source-based multicast core tree approach may include The Distance Vector Multicast Routing Protocol (DVMRP), The Multicast Open Shortest Path First protocol (MOSPF), The Protocol Independent Multicasting-Dense Mode (PIM-DM).

- DVMRP[8]- a distance vector routing protocol. It was derived from the Routing Information Protocol (RIP), which was designed for unicast routing. DVMRP constructs source-based multicast trees using the Reverse-Path Multicast (RPM) algorithm.

- MOSPF[9]- depends on OSPF[10] to construct unicast routing table. In OSPF, each router within a routing domain keeps topological and state information of this domain. This is achieved through link-state advertisement (LSA) flooding. An MOSPF router makes use of this feature, uses IGMP to monitor multicast group membership on directly attached sub networks and floods group-membership LSA to all the other routers. An MOSPF router builds a shortest-path tree rooted at the source using Dijkstra algorithm.

- PIM-DM[11]- , RFC 3973, was designed to be used for groups with a large number of members (dense mode). As in DVMRP, "Flood and Prune" Reverse Path Forwarding (RPF) is used in PIM-DM. The difference is that while DVMRP maintains its own routing table, PIM-DM uses the unicast routing table to perform RPF check.

The group-shared tree approach: One node for each group is selected as the core (or termed as a rendezvous point, RP) for the group. A tree rooted at the core is then constructed and span all the group members [7].

CBT[12]- The Core Based Tree routing protocol, is an attempt to improve the scalability of DVMRP and MOSPF. CBT uses the basic sparse mode paradigm to create a single shared tree used by all sources. The tree is rooted at a core. All sources send their data to the core, and all receivers send explicit join messages to the core.

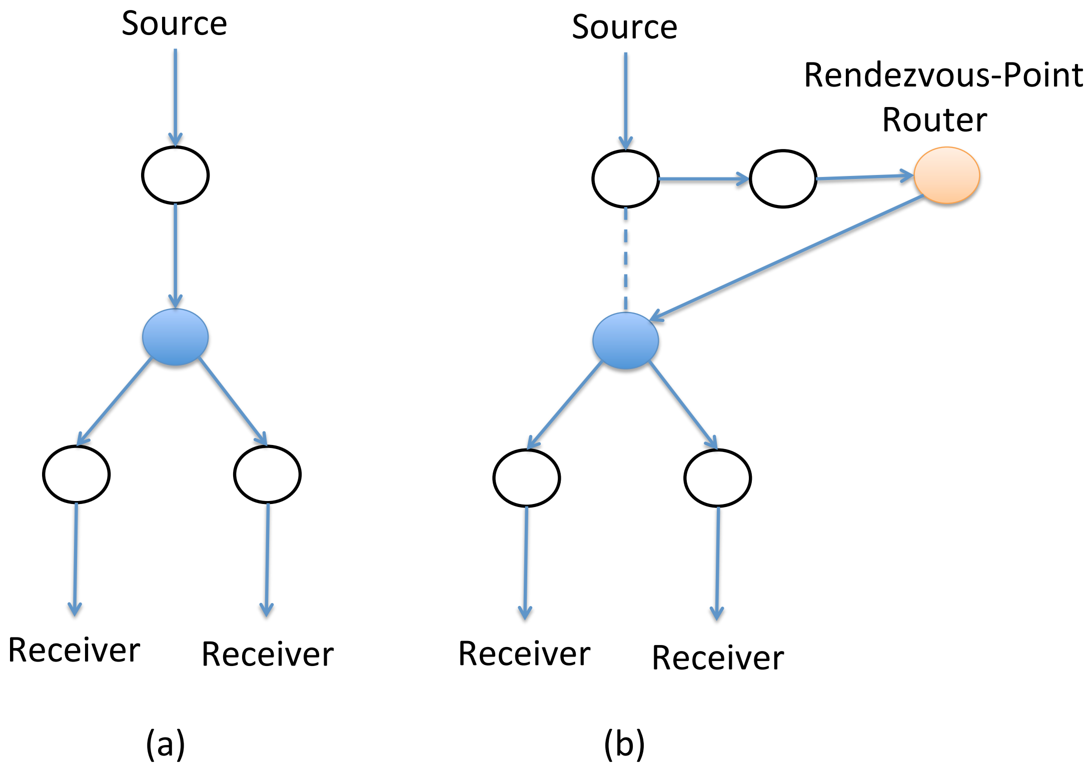


Figure 2.1: Two methods of constructing multicast algorithms

(a) dense mode, using a source-based tree; (b) sparse-mode, using a shared tree.

PIM-SM[13]- The Protocol Independent Multicasting-Sparse Mode, design for groups where members are sparsely distributed over the routing domain. Difference to almost others intra-domain multicast routing protocols, PIM-SM uses both source-based tree and group-based tree.

PIM-SM is much more widely used than CBT. It is similar to PIM-DM in that routing decisions are based on unicast routing table, but the tree construction mechanism is different. This protocol basically based on the multicast tree's rendezvous point (RP), where sources can transfer data to receivers through. PIM-SM operation include these steps:

- Local hosts joining a group: when a host want to join a multicast group, it sends IGMP messages to its upstream router, which means the routers can join the multicast group. The upstream router sends "Join message" toward the RP (Rendezvous point).
- Establishing the RP-rooted shared tree: The "join message" is processed by all the routers between the receiver and the RP, then the new branch of multicast tree for the new member is set-up.
- Hosts sending to a group: when a source join a multicast group, it send the packets to its DR. Then DR encapsulates encapsulated data packets in a "Register message" toward the RP. The RP decapsulates each Register message and forwards the enclosed data packet natively to downstream members on the shared RP-tree.
- Switching from shared tree (RP-tree) to shortest path tree (SP-tree): The router can switch to a source's shortest path tree (SP-tree) after receiving packets from that source over the shared RP-tree.

  PIM-SM with its operation is considered to be robust, flexible, and scalable; and it is now the most widely used protocols in large networks.

2.2 Inter-domain multicast routing protocols

2.2.1 Autonomous System

In the last section, we have discussed about intra-domain multicast routing. In this section, first we talk about domains - autonomous system (AS), as a background for inter-domain multicast routing protocol.

Autonomous System is a set of routers under single technical administration, using an interior gateway protocol, and common metrics to route packet within the AS, and using an exterior gateway protocol to route packets to other ASes. Number of ASes all over the Internet has increased dramatically over years (Figure 2.2). Nowadays, there are more than 35000 ASes over the Internet.

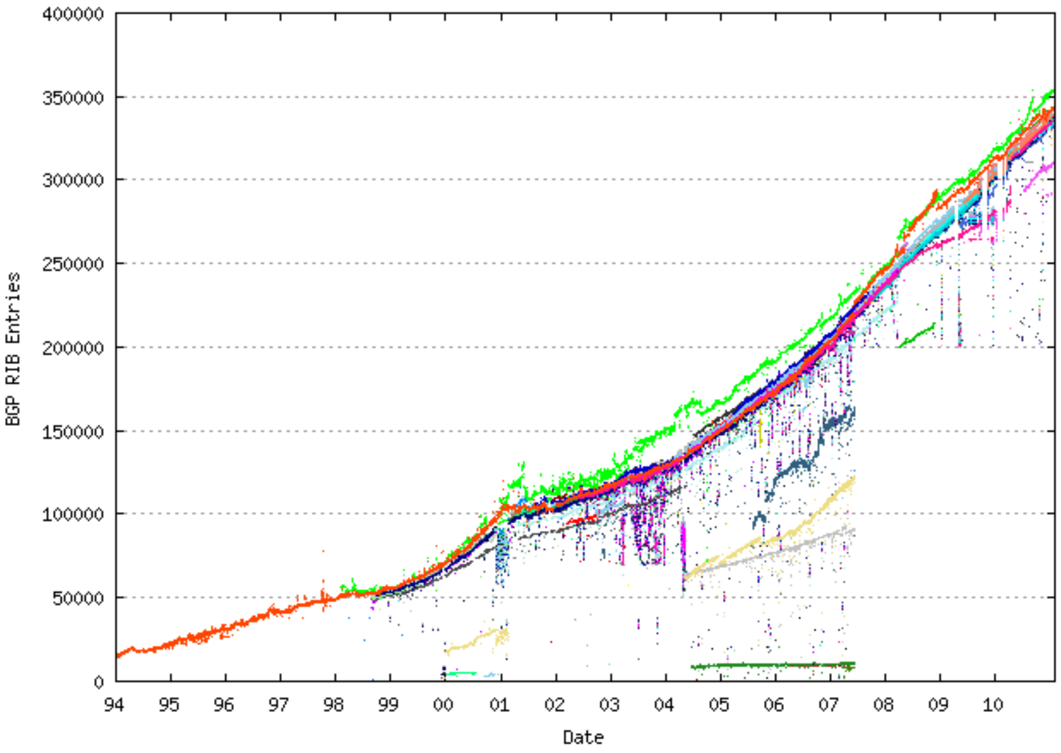


Figure 2.2: Growth of BGP - AS numbers

Sources: http://bgp.potaroo.net

As the growth of ASes number and independence policy in each AS, the key requirements for inter-domain multicast routing concern scalability, stability, policy, and intra-domain multicast routing protocol independence.

2.2.2 Scalability in Inter-domain Multicast Routing

Nowadays, scalability is typically viewed as one of the biggest challenges in inter-domain multicast routing. Understanding that the number of users through the global network is very large and dynamic, multicast routing must be very efficient to be able to achieve scalability state.

Scalability in inter-domain multicast routing is described with two options as below[6]:

Multicast forwarding state: The amount of state, which must be distributed to permit global multicast forwarding should be minimal and scale well as the Internet expands. Where there are no receivers or senders, state for the group should be minimized.

Address allocation: The address allocation scheme should scale well as the number of groups increases. The probability of address collision, as well as the delay in obtaining an address to assign to a group, should be small, consistent with best- effort architectural principles. An application-or session-level protocol should be able to detect and drop packets that it receives due to infrequent collisions to the extent required by that application.

2.2.3 Multicast Source Discovery Problem

In this subsection, we would like to discuss one of inter-domain multicast routing protocol: multicast source discovery. Given that in each domain, PIM-SM is used as an intra-domain multicast routing, the problem is basically how to inform an RP in one domain about the existence sources in other domains.

For a domain, there is only one RP. Multicast source discovery problem arises when multicast group members are spread in multiple domains. There is no mechanism to connect multiple intra-domain tree together. While the traffic from all the sources for a particular group within a particular domain will reach the group's receivers, any sources outside the domain will remain disjoint. The reason is that within a domain, receivers send join messages toward one RP, and sources send register messages to the same RP. However, there is no way for an RP in one domain to find out about sources in other domains using different RPs. There is no mechanism for RPs to communicate with each other when one receives a source register message.

Figure 2.3 below describes Multicast Source Discovery Problem:

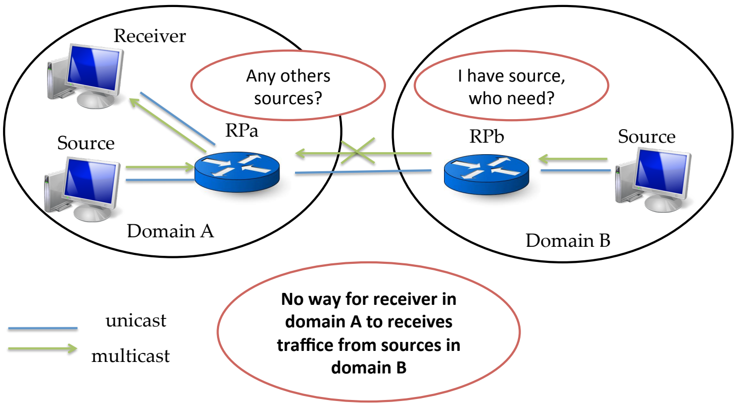


Figure 2.3: Multicast Source Discovery Problem

2.2.4 MSDP protocol

Many protocols have been already proposed for inter-domain multicast routing: MSDP, BGMP, EXPRESS, SIMLPEM, HIP.

However, with the complex and difficulties in implementation as well as incomplete in algorithm and operation, many protocols are considered as far-term solution and would only be deployed in far future (Table 2.1).

In which, MSDP is considered the short - term solution for inter-domain multicast routing.

FEATURES	MSDP/PIM-SM	BGMP	EXPRESS	SIMPLEM	HIP
Development Stage	Not deployed yet	Standardization Process	Research only	Research only	Research only
Scalability	Low	High	Low	High	High
Easy to Implement	Yes	No	No	Yes	No

Table 2.1: Inter-domain protocols

Multicast Source Discovery Protocol (MSDP) is a mechanism to connect multiple routing domains, generally PIM-SM domains, as well as to solve multicast source discovery problem. MSDP allows sources for a multicast group to be known to all of the rendezvous points (RPs) in different domains.

MSDP's operation[14]:

1. When a new source for a group becomes active it will register with the domain's RP.

2. The MSDP peer in the domain will detect the existence of the new source and send a Source Active (SA) messages to all directly connected MSDP peers.

3. MSDP message flooding:

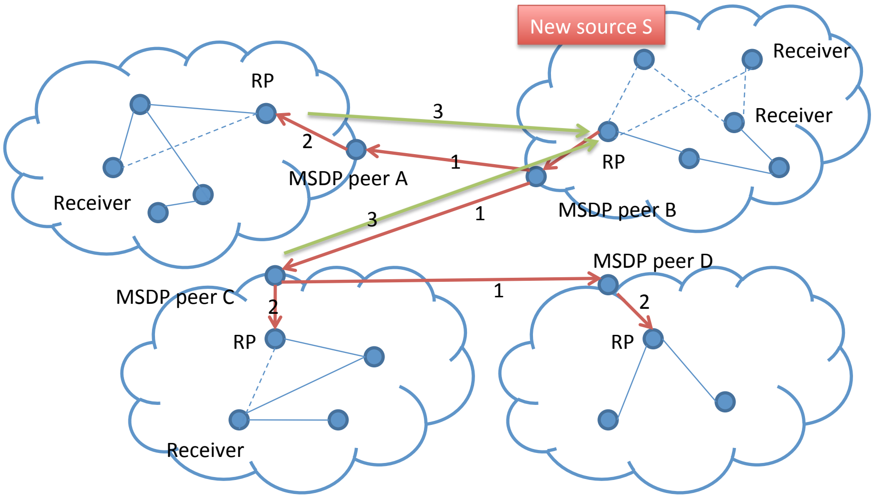
-MSDP peers that receive an SA message will perform a peer-RPF check. The MSDP peer that received the SA message will check to see if the MSDP peer that sent the message is along the correct MSDP-peer path. These peer-RPF checks are necessary to prevent SA message looping.

-If an MSDP peer receives an SA message on the correct interface, the message is forwarded to all MSDP peers except the one from which the message was received. This is called as peer-flooding.

4. Within a domain, an MSDP Peer (also the RP) will check to see if it has state for any group members in the domain. If state does exist, the RP will send a PIM join message to the source address advertised in the SA message.

5. If data is contained in the message, the RP then forwards it on the multicast tree. Once group members receive data, they may choose to switch to a shortest path tree using PIM-SM conventions.

6. Steps 3-5 are repeated until all MSDP peers have received the SA message and all group members are receiving data from the source.



1:MSDP Peer flood SA messages

2: RP check if got receiver or not

3: RP send join message to the new source RP

Figure 2.4: MSDP operation

One of the advantages of using MSDP is that there is no shared tree built across domains. Therefore, each domain can depend solely on its own RP. SA state is not stored at all the MSDP peers, but the one that originated the SA. Data could already be encapsulated in SA messages for low-rate bursty sources. MSDP peers could cache SA messages and if they do, MSDP peers can get MSDP state sooner and reduce join latency for new joiners[5].

2.3 Problem Statement

Scalability issue is a big concern when we consider MSDP. Because of the way MSDP operates, every time a new source appears, SA messages must be flooded all over domains to inform other RPs. If multicast were popular on the Internet, then numbers of sources and receivers may reach very high as well as the number of SA messages (plus data) being flooded around the network could become very large. Over-flood of SA messages can be easily predicted if MSDP is used over global-Internet applications.

This thesis focuses on the scalability issue on MSDP protocol, and discusses an approach to replace MSDP protocol with the DHT implementation on PIM-SM which would be explained more details in next chapter.

Chapter 3

DHT Network for Multicast Source Discovery

3.1 Distributed Hash Table

3.1.1 Introduction of Distributed Hash Table

A distributed hash table (DHT) is a class of decentralized distributed system that provides a lookup service similar to a hash table; (key, value) pairs are stored in a DHT, and any participating node can efficiently retrieve the value associated with a given key. The responsibility for maintaining the mapping from keys to values is distributed among the nodes, in such a way that a change in the set of participants causes a minimal amount of disruption. This allows a DHT to scale to extremely large numbers of nodes and to handle continual node arrivals, departures, and failures.

DHTs form an infrastructure that can be used to build more complex services, such as distributed file systems, peer-to-peer file sharing and content distribution systems, cooperative web caching, multicast, anycast, domain name services, and instant messaging.

The key characteristics of DHT are:

- Decentralization: the nodes collectively form the system without any central coordination.
- Scalability: the system should function efficiently even with thousands or millions of nodes.
- Fault tolerance: the system should be reliable (in some sense) even with nodes continuously joining, leaving, and failing.

A key technique used to achieve these goals is that any one node needs to coordinate with only

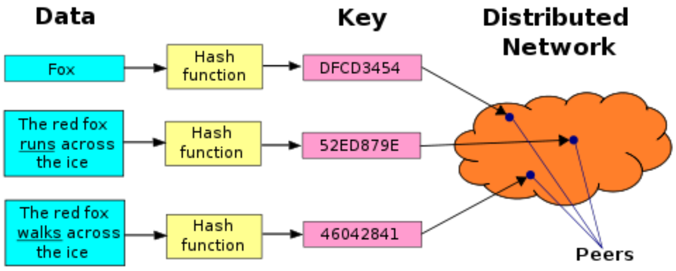


Figure 3.1: Simple example of DHT

Sources: http://en.wikipedia.org/wiki/Distributed_hash_table

a few other nodes in the most common systems, O(log(n)) of the n participants, so that only a limited amount of work needs to be done for each change in membership.

3.1.2 Distributed Hash Table in P2P

DHT is now mostly being used in P2P system as the infrastructure for file sharing. A host with a keyword for filename can find others hosts who get the file in DHT network with much better resources and network bandwidth uses compared to normally searching method. DHT only use O(log(n)) in order to find file data information.

DHT's operation in P2P can be described as follows:

- To store a file with given filename and data in the DHT, the SHA-1 hash of filename is generated, producing a key k, and a message $put(k, data)$ is sent to any node participating in the DHT.
- The message is forwarded from node to node through the overlay network until it reaches the single node responsible for key k. That node then stores the key and the data.
- Any other client can then retrieve the contents of the file by again hashing filename to produce k and asking any DHT node to find the data associated with k with a message $get(k)$.

- The message will again be routed through the overlay to the node responsible for k, which will reply with the stored data.

3.1.3 DHT maintenance in overlay network

Each node maintains a set of links to other nodes (its neighbors or routing table). Together these links form the overlay network. A node picks its neighbors according to a certain structure, called the network's topology.

All DHT topologies share some variant of the most essential property: for any key k, each node either has a node ID which owns k or has a link to a node whose node ID is closer to k, in terms of the keyspace distance. It is then easy to route a message to the owner of any key k using the following greedy algorithm: at each step, forward the message to the neighbor whose ID is closest to k. When there is no such neighbor, then we must have arrived at the closest node, which is the owner of k. This style of routing is sometimes called key-based routing.

Two important points on the topology are to guarantee that the maximum number of hops in any route (route length) is low, so that requests complete quickly; and that the maximum number of neighbors of any node (maximum node degree) is low, so that maintenance overhead is not excessive. Having shorter routes requires higher maximum degree. Some common choices for maximum degree and route length are as follows, where n is the number of nodes in the DHT, using Big O notation:

- Degree O(1), route length O(n)

- Degree O(log(n)), route length O(log(n) / log(log(n)))

- Degree O(log(n)), route length O(log(n))

- Degree , route length O(1)

The third choice is the most common, even though it is not optimal in terms of degree/route length trade off, because such topologies typically allow more flexibility in choice of neighbors. Many DHTs use that flexibility to pick neighbors, which are close in terms of latency in the physical underlying network.

3.2 DHT implementation operation

Instead of sending SA messages through all the domains and ASes like MSDP protocol, we can use DHT as infrastructure network module for RPs in order to search for multicast sources for receivers in the domains. With this solution we could reduce the scalability problem in MSDP protocol.

The detailed source-lookup operation can be described as below:

- When a new source for a group becomes active it will register with the domains RP.
- For every 1 minute, RP in each domain detects the existence of source in the domain and hash source's information (Source address, Multicast group address) into hash file.
- Hash file will be stored in DHT network nodes with keyword is multicast group address, value is Source address.
- Within a domain, RP checks the multicast group interested in the domains, and searches for sources with same multicast group in other domains with keyword is multicast group address.
- After receiving the sources information, RP triggers PIM (S,G) join event towards the data sources and create branch of source-tree from sources to domain . After this step, data packets from sources can arrive RP via this tree branch.

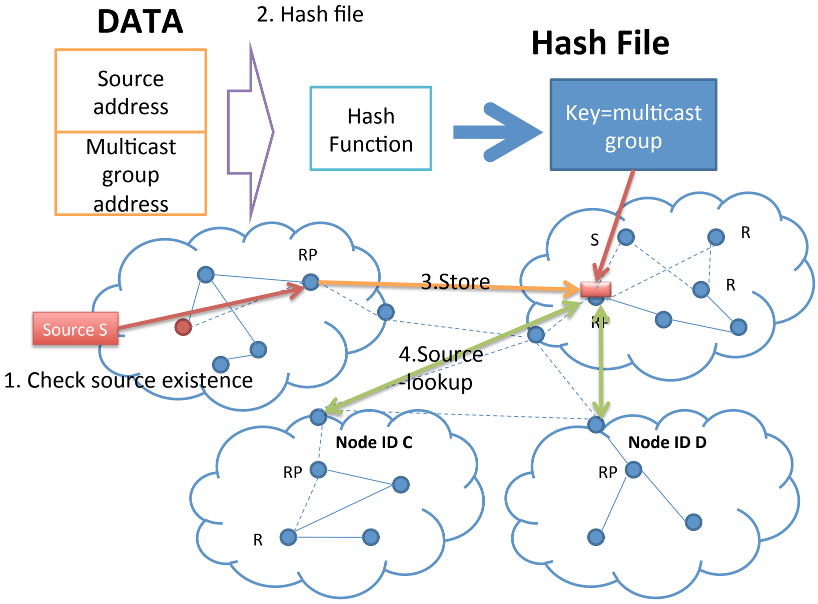


Figure 3.2: Approach

With the approach described above, there are points need to be considered:

1. Source-lookup success rate must be 100%. Or else there may be a chance that some source's existence can never be known to other RPs.

2. One multicast group address may have many sources. Therefore, source-lookup must support multiple values searching which mean 1 same keyword (multicast group address) must return all different values.

3. DHT data integrity: DHT storage data must always be fresh in order to keep the DHT storage and lower the network bandwidth. If a source stop sending data to a multicast group, then information of that source must be deleted from the DHT nodes.

3.3 DHT and MSDP comparison for multicast source discovery

In this section, we do a comparison of number of messages sending between MSDP protocol and DHT implementation by algorithms with considering parameters:

- Number of domains - RPs: n.
- Number of sources: S.
- Number of multicast group: G.

With MSDP protocol, with one source's existence RP needs to send n messages to other domains to inform others all others RPs. So in N domains, number of messages sending would be:

(MSDP) Number of message $= S * n$**.**

With DHT implementation, RP in one domain with $O(log(n))$ effort can put a source hash file to other DHT node as well as to get sources list for one multicast group.

So in n domain:

- Number of messages to put source: $S * O(log(n))$.
- Number of messages to get source list: $G * O(log(n))$.
- Number of message for DHT maintenance: $O(log(n))$.

Total number of messages by DHT implementation:

(DHT Implementation) number of messages $= (S + G + 1) * O(log(n))$**.**

As the number of domains n increases, or the average number of sources in each domain is bigger compared to number of multicast group, then the DHT implementation number of messages would be much less if comparing to MSDP protocol.

If number of domains as well as n is big number then number of messages:

(MSDP) $S * n >> (S + G + 1) * O(log(n))$ **(DHT).**

As we have discussed in previous chapter, the number of ASes over Internet today is over 35000. If multicast is successfully over the Internet, users for multicast applications would increase significantly. Therefore, number of messages transferring by DHT implementation operation would be reduced multiple times over MSDP protocol operations.

Chapter 4

System Design

4.1 System Environment

As we discussed in the previous chapter about operation, DHT implementation is designed to run in domains RP routers only. Any other routers beside RPs can run normal PIM-SM routing daemon, with no relation with DHT network.

Figure 4.1 shows the overview environment for this DHT implementation system design.

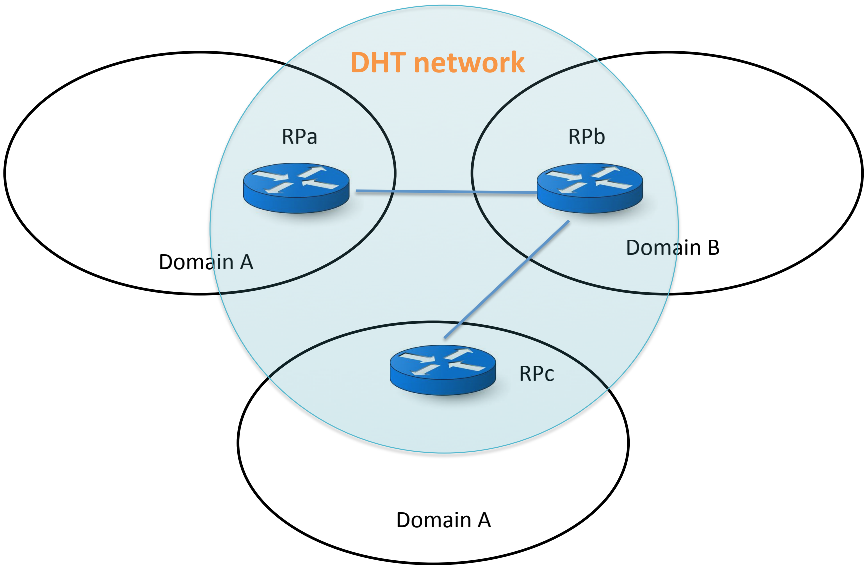


Figure 4.1: System Environment

4.2 System Architecture

Figure 4.2 illustrates overview design of the system.

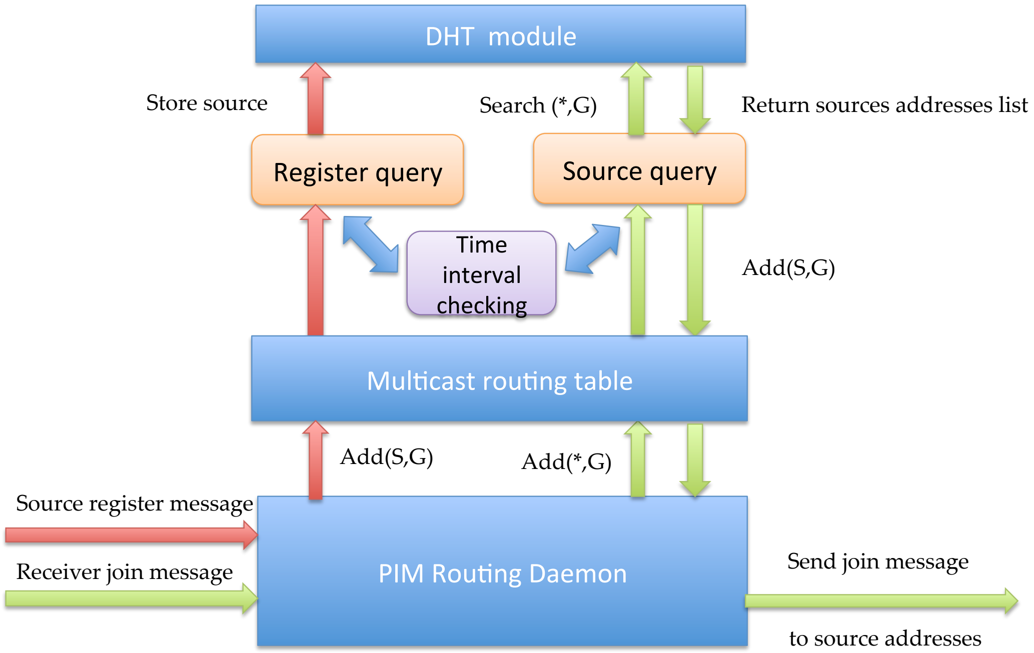


Figure 4.2: System architecture

This system get two types of messages as the input:

- Source-register message.
- PIM-Join message.

When the system receives the input, PIM-SM daemon will do the first work, get the information from two messages, add to routing table, and does the PIM-SM routing control. Information of sources addresses and multicast group addresses are then transferred between PIM-SM and DHT module by the function of two main components: *RegisterQuery* and *SourceQuery*. In this system, PIM -SM daemon and DHT network module is running parallel, and the two system's main components: *RegisterQuery* and *SourceQuery* are acting as bridging functions between them.

RegisterQuery is described as a function with task to get the domain's sources information (source address and multicast address) from PIM- SM daemon (S,G) then put them inside DHT network.

SourceQuery gets domain's multicast group interest from PIM-SM (*,G) routing table and does the sources-lookup, which is to get the sources list (with the same multicast group address) using DHT module.

The system uses Time-interval checking function calling to the two main components periodically to keep DHT database over the network integrity.

PIM-SM daemon after receiving the sources list will send the PIM Join towards every source as the output of this system.

4.3 Registering Source

When the PIM routing daemon interface receives a Register message - new source (S,G) appears, PIM-SM daemon will add the information of the source which include source address and multicast group address (S.A/M.A information) from the Register message to routing table. Then, Register Query function is called, gets (S.A/M.A information). Using DHT hash module, S.A/M.A will be hashed to a hash file with key - value pair is key: multicast address - value: source address. After that the key-value pair hash file will be stored inside the DHT network nodes (using DHT *put*() interface).

4.4 Discovering Sources

When the PIM routing daemon interface receives a PIM-Join message or after a set of time (using Time-interval checking module to check) Source Query component is called.

Source query's function can be described with these steps:

1. Get receiver's multicast group addresses (M.A) through PIM Routing daemon routing table.

2. Use DHT module to search sources addresses (using DHT *get*() interface) with keyword = multicast group address (M.A).

3. Inform PIM routing daemon sources addresses list and multicast group (S.A/M.A) to routing table.

4.5 DHT storage integrity

Figure 4.4 describes solution to keep DHT storage integrity:

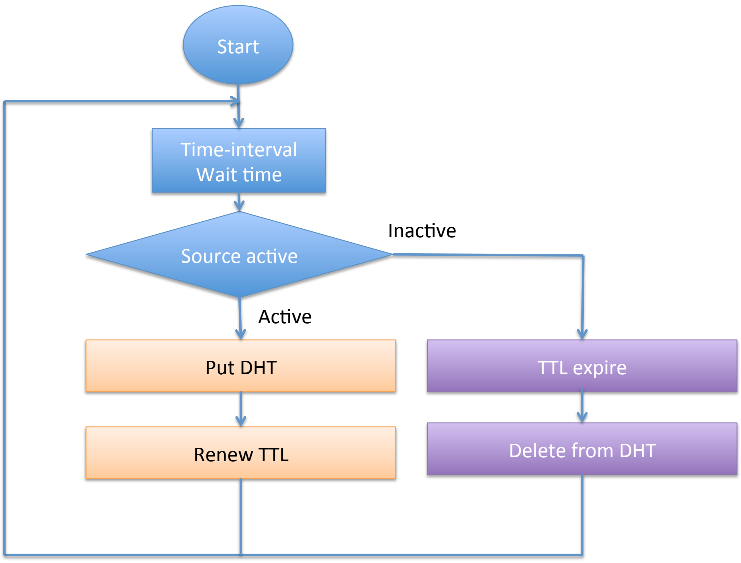


Figure 4.3: TTL and interval-time function to keep DHT storage integrity

This system must keep DHT storage integrity. This requirement means that ’active source’ hash file information must always be kept in DHT network storage, while ’inactive source’ (source which stop sending traffic) hash file information must be deleted from DHT network storage. This requirement is to lower the network bandwidth as well as to limit the storage.

The solution is to use time-to-live (TTL) and time-interval check function. As the total storage in the system for DHT is limited, DHT treats stored values as ”soft state” by expiring them after a TTL interval. The node in DHT which puts a key-value pair specifies the pair’s TTL. There is also no notification sent to node when the TTL expires, so nodes must manage their storage to achieve the degree of persistence they desire. When a node wishes to cause its key-value pair to persist beyond its original TTL, it may refresh the pair by issuing another put. The TTL in the latest put will take effect for that value.

Therefore, whenever a multicast source become ’inactive’, its RP would stop putting hash file to DHT storage, and as a result source’s information existence kept in DHT will be automatically deleted as its TTL expires.

To make sure ’active source’ information would always stored in DHT, this system use time-interval check function to call Register Query components to put ’active source’ regularly to DHT network before its TTL expires. Time interval-check number must be smaller than TTL.

Figure 4.4 describes solution to keep DHT storage integrity.

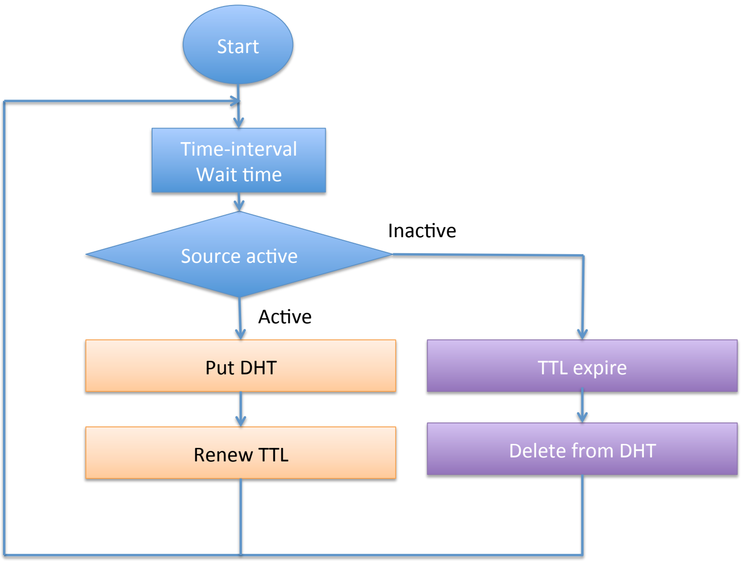


Figure 4.4: TTL and interval-time function to keep DHT storage integrity

To recognize new multicast sources in other domains, time-interval check function also calls Source Query components periodically to check for new sources.

There are many factors to optimize appropriate TTL as well as time-interval. For example, if TTL and time-interval is low then latency time is better, however it takes more bandwidth; in contrast, TTL and time-interval high: bandwidth consumption lower but higher latency. We would like to define appropriate TTL and time-interval as future works.

4.6 Summary

In this chapter, we proposed the design of DHT Network Implementation for PIM-SM system. The system design includes Register Query and Source Query as the main components to communicate between PIM-SM daemon and DHT network module; while time-interval checking module calling two components periodically to keep DHT storage data integrity.

Chapter 5

Implementation

5.1 Environment

Implementation environment of DHT on PIM is shown in Table 5.1 below:

Operation System	Ubuntu 10.10
PIM-SM Daemon	PIMD
g++ version	4.4.5
DHT	Bamboo DHT
Java JDK version	1.6.0 build 21

Table 5.1: Implementation Environment

Bamboo DHT is a good choice to this implementation, with these following reasons:

- Source look up success rate is high and stable: the router running bamboo DHT in light churn returns consistent results 99.99% of the time, with a fast response time.
- Bamboo supports to multiple value search .
- Bamboo has been tested for a long time in Open-DHT server.

5.2 Inter-process communication

5.2.1 Bamboo System call

Since PIMD uses C while Bamboo-DHT uses Java, to get Bamboo-DHT put/get module running with PIMD, we need an inter-process communication between two of them. In this case we use C compiler system-call to call Bamboo-DHT application and Procedure Call Function (RPC) to run Put/Get interface.

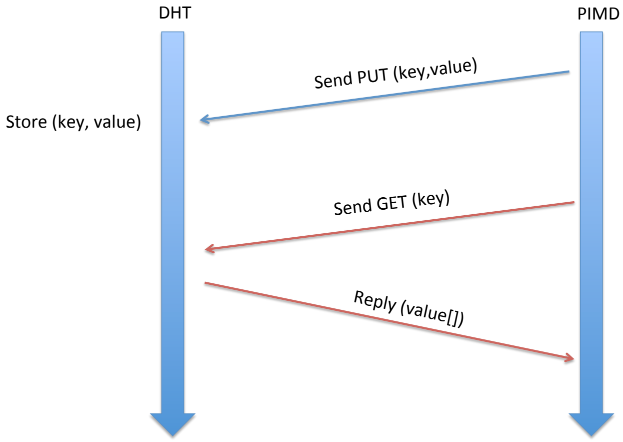


Figure 5.1: DHT - PIMD inter process communication

Bamboo-DHT uses configuration file to run the network. Before joining Bamboo-DHT, a node must define its own nodeID, gateway connect, as well as opening a port for local RPC call. as Below is the compositions for a node which need to be defined in the configure file:

```
string NodeID; // Each Bamboo Node get Unique NodeID

string Gateway; // Gateway to connect to other node through Bamboo-DHT network

int port; // Port to use RPC

string store; // Storage Database file name

string file_cfg; // Config file name
```

Figure 5.2: Bamboo config class

5.2.2 Put/Get Interface

This implementation uses RPC to call the Put/Get module in Bamboo-DHT from the local host. The required components include Put/Get key - the multicast group address, Put value_val: Source-address, Get-value_val: List of sources addresses, Put TTL: Source's information storage time.

```
struct bamboo_put{
  char key[20]; // put-get key
  char *value_val; // put value
  u_int ttl_sec; //time to live to second
};
struct bamboo_get{
  char key[20]; // put-get key
  char *value_val; // get values
  u_int value_len; // number of values
};
```

Figure 5.3: Bamboo put-get data structure

5.3 Implementation in PIMD

PIMD is a complete routing daemon. In order to implement DHT Put/ Get interface in PIMD, which would take and change the information in the routing table of PIMD, we need to use a safe solution as not to do conflict between PIMD and DHT implement. In this implementation, we use thread, one to run normal PIMD and one to run DHT implement. Thread running DHT implement would include functions:

- Get source addresses and group addresses from PIMD routing table.
- If (*,G) (group interest) then Get Source lists from DHT Figure 5.4.
- If (S,G) (source active) then Put to DHT Figure 5.5.

```
for (g = grplist->next; g != (grpentry_t *) NULL; g = g-> next)  // grouplist
  if ((r = g->grp_route) != (mrtentry_t *) NULL) {              // point to (*,G)
    strcpy (get_args.key, inet_fmt(g->group, s2, sizeof(s2)))   // get key = group
    get_result = do_get(get_args);                       // source list store in
        get_result
  }
```

Figure 5.4: Get source list

```
for (g = grplist->next; g != (grpentry_t *) NULL; g = g-> next) { // grouplist
  for (r = g->mrtlink) ; r!= (mrtentry_t *) NULL; r = r->grpnext) { // sources
      put_args.key = r->group;
      put_args.val = r->source->address
      do_put(put_args);
  }}
```

Figure 5.5: Get source list

5.4 Trigger (S,G) join

After receiving a source address list from Get interface of DHT, we need to trigger the (S,G) join towards sources. There are 2 steps to trigger:

- Create Source - group (S,G) entry in the routing table.
- Send (S,G) join message toward source.

However, there is one problem occur after we create Source-Group entry in PIMD multicast routing table: PIMD cannot differ the source inside its own domains and the sources in other domains. So after Source-Group entry is created, PIMD also PUT(S,G) of the source information which just received to DHT network. This could create two problems:

- Duplicate PUT: many RPs in different domains do PUT for 1 source information. It would increase number of messages as well as make bandwidth consumption much higher.
- 'Inactive Source' information is not deleted: There's a possibility that the source information stored in DHT network is never deleted, even though it becomes 'inactive'. For example, there are two different RPs in two domains receive data from a source outside their domains. They get information of the source through DHT and create (S,G) entry in its multicast routing

table. Periodically, they do PUT/GET source information to/from DHT. As a result, when source is inactive, even though its own RP stop PUT its information to DHT, because the other two RPs continuously PUT and GET source information to each other through DHT network, there is a chance that source existence is stored forever in DHT storage.

The solution here is to create a link list of source address, which include sources addresses from other domains, so that before PUT any sources information to DHT, system must check if the source is inside domain or not. Only if source is inside its own RPs domain then we do PUT.

Link list functions include:

- Search-Source: to check if a source is in link list or not, if source is in link list that means that source is from other domains.
- Add-Source: if get source address from other domain then add source address to link list.
- Delete-Source: if a source from other domain is inactive, the delete it from link list.

Chapter 6

Evaluation

6.1 Evaluation Overview

We evaluate this implementation in terms of scalability characteristics in two points:

- Number of messages:

  First, we evaluate this implementation in terms of the total number of messages in all network. This is the most important point to be considered when evaluate this implementation, as the purpose of this research and evaluation is to solve the scalability issue in MSDP by reducing total number of messages all over the network. As we have calculated and proved in the Chapter 3, number of messages used by DHT implement should be less than numbers of messages used by MSDP. Therefore, in this evaluation, we compare total number of messages between the actual evaluated result and the expected result (by calculation). If there is no difference in the number of messages between actual evaluated result and the expected result, it is a proof that this implementation works, and DHT implementation to replace MSDP would be a succeed.

- Bandwidth consumption:

  Bandwidth consumption is also an very important point to evaluate scalability characteristics of any systems. In this evaluation, besides total number of messages, we also evaluate the bandwidth consumption over network links.

6.2 Scenario

6.2.1 Topology

We assume that one independent RP with its network is a PIM-SM domain. In each domain, RP has two interfaces, one to connect to its own intra-domain multicast source-receiver link, one to connect to the other domains RPs. Multicast sources and receivers can connect directly to its own RP.

In this evaluation topology, we use 3 domains: domain A, domain B, and domain C. Both domain A and domain C have connection to domain B. Inside domain A and domain C, hosts are sources, while inside domain B, hosts are receivers.

In this topology, it must be assured that receivers in domain B get the data from sources in domain A and domain C. Figure 6.1 below show the topology of this evaluation:

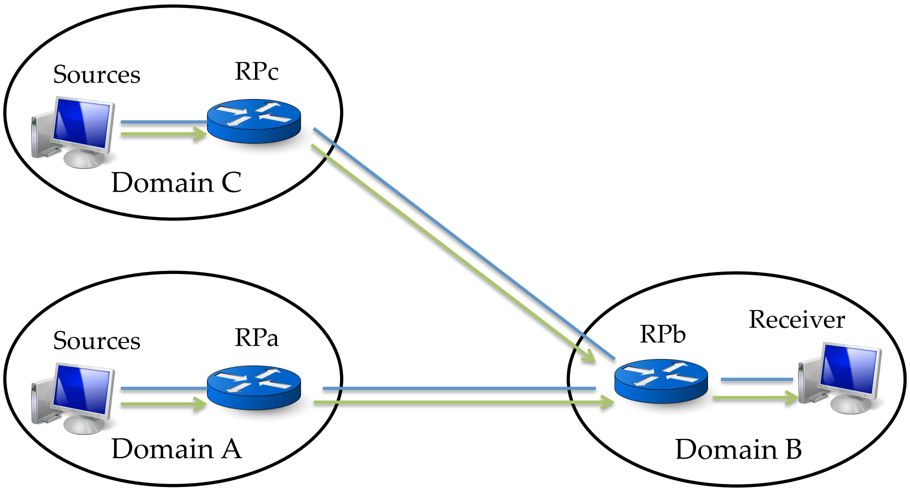


Figure 6.1: Evaluation Topology

6.2.2 Scenario operation

To be able to evaluate in terms of scalability, we use a scenario with high numbers of sources and receivers. Evaluation scenario is described as follows:

- First, we evaluate the network when there is no receiver or source active. In this time, we can see the DHT maintenance network bandwidth consumption.
- Then, we increase the number of receivers in domain B from 0 to 50, 100, and 200 consecutively (use mcfirst tool). Each receiver joins a different multicast group to trigger the number of multicast groups interested in domain B to a high number. After this operation, we can see the change in the network links with the effect of DHT (get) function (get function is called to get multicast sources addresses list when there are interest multicast groups).
- At last, we increase the number of sources in domain A and C from 0 to 200, 400, and 600 consecutively (use ping), with the concept of two sources join one multicast group. From here, we can see the changes in the network with the effect of DHT (put) function (Put function is called to store the sources in DHT network).

Table 6.1 below shows scenario operation of increasing number of users (sources and receivers) over the network:

Time	Domain A	Domain B	Domain C	Total
T1	0	$0->50->100->200$	0	200
T2	$0->200->400->600$	200	0	800
T3	600	200	$0->200->400->600$	1400

Table 6.1: Evaluation Scenario

In order to evaluate the number of messages and bandwidth consumption in network topology, data taken and analyzed is number of packets and packet length in two interfaces of all 3 RP routers.

6.3 Evaluation Result

6.3.1 DHT maintenance network

We evaluate this implementation when the sources and receivers numbers are still 0 to check out for DHT maintenance network bandwidth consumption. The result is shown in Figure 6.2 :

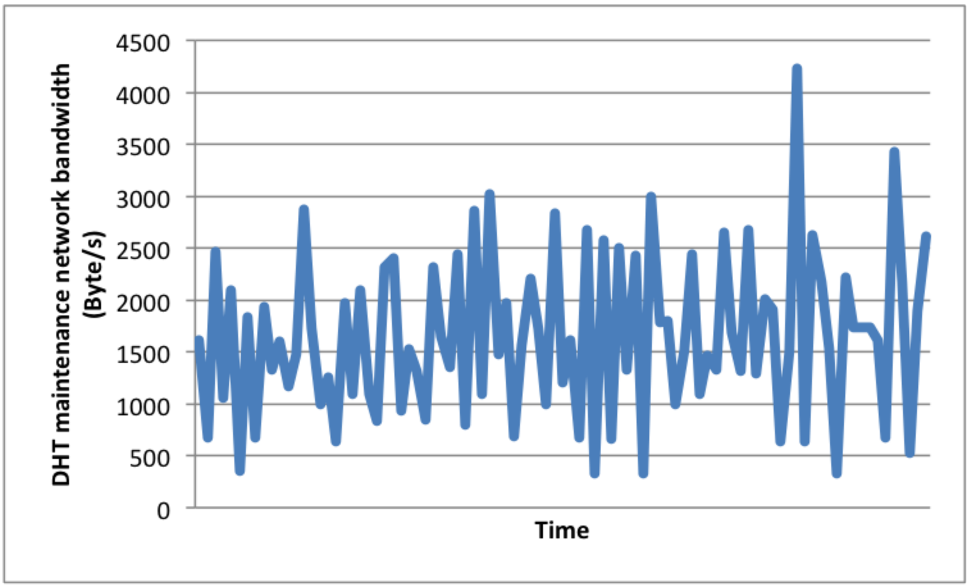


Figure 6.2: DHT maintenance network bandwidth consumption over each link

From the result figure, we can see that DHT maintenance network consumption overhead is quite low. Maximum network bandwidth consumption only reaches to around 4(KB/s), while the average network bandwidth consumption is around 1.8(KB/s). Even though the network topology is small (number of nodes in DHT network is only 3), but with the result of low network bandwidth consumption as evaluated, as well as the network consumption of DHT network is defined as O(log(n)), we believe that DHT maintenance is not a big concern when we consider about scalability issue in large-scale network for this implementation.

6.3.2 Change in number of messages over scenario operation

This subsection will describe change in number of messages over the network when evaluation scenario operation is processed.

GET effect

Figure 6.3 shows the change in the network link between RPb and its own receivers when receivers in domain B increase from 0 to 50, 100, and 200 consecutively using mcfirst tool. Figure 6.3 proves that receivers sent the IGMP join messages to RPb to join for multicast groups.

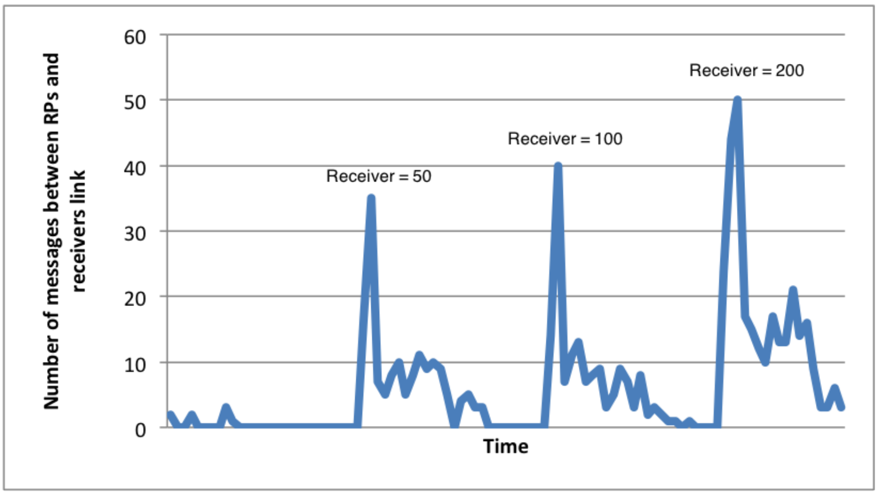


Figure 6.3: IGMP messages from receivers

However, as Figure 6.4 shows number of messages between RP routers, we can see that the number of messages did not have any significant change at all even though GET function is called. Number of messages is just like DHT maintenance. The reason is that in this evaluation, *RPb* is the node to store the information. When the GET function is called, it only searched inside of node *RPb*, and as a result there is no change in the number of messages over the network.

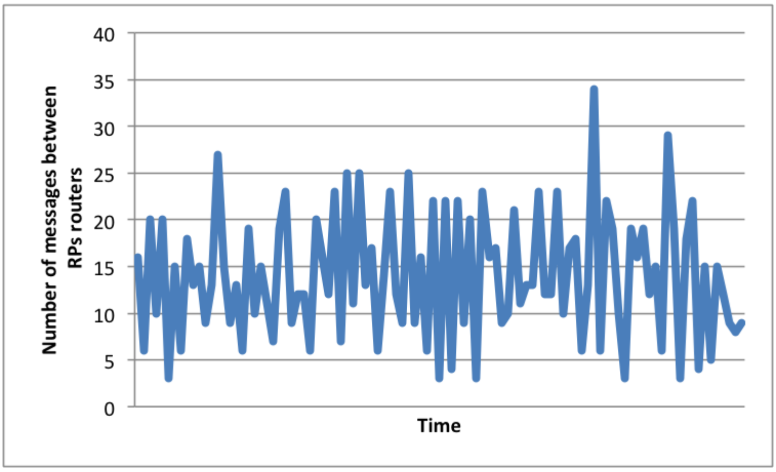


Figure 6.4: Number of get messages

PUT Effect

Figure 6.5 shows the change in number of messages between domains RPs router when the number of sources in domain A increases from 0 → 200 → 400 → 600. From the figure, number of messages over the network link increases significantly as the number of sources start to increases.

However, even though the maximum number of messages over one second does not change much, because of the time to proceed (time to send messages) by RPs router is longer, number of messages different is also bigger.

For example: As the number of sources doubles from 200 → 400, maximum number of messages over one second is almost not changed (only from 425 → 525). However, because of time to proceed doubles, number of messages when the sources change from 200 → 400 is more than double.

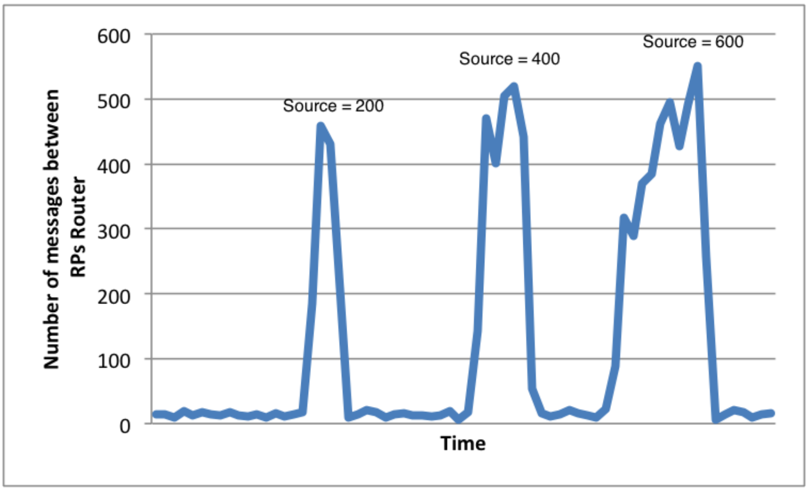


Figure 6.5: Number of put messages

After putting, PIM join/prune messages are sent from routers. Figure 6.6 shows the tcpdump snapshot as the proof of (S,G) join:

```
03:02:16.932838 IP 192.168.2.2 > 192.168.1.1: PIMv2, Join / Prune, length 634
03:02:16.933445 IP 192.168.2.2 > 192.168.1.2: PIMv2, Join / Prune, length 634
03:02:16.934234 IP 192.168.2.2 > 192.168.1.3: PIMv2, Join / Prune, length 634
03:02:16.935321 IP 192.168.2.2 > 192.168.1.4: PIMv2, Join / Prune, length 634
```

Figure 6.6: Send PIM Join/Prune

6.3.3 Total number of messages

Figure 6.7 shows the change in total numbers of messages between RPs when number of users (sources + receivers) increases, as well as compare it with expected result from calculation. The blue line is actual evaluated result, while the red line is expected result. Expected result is calculated: (Number of users) * (Number of network links) * 2.

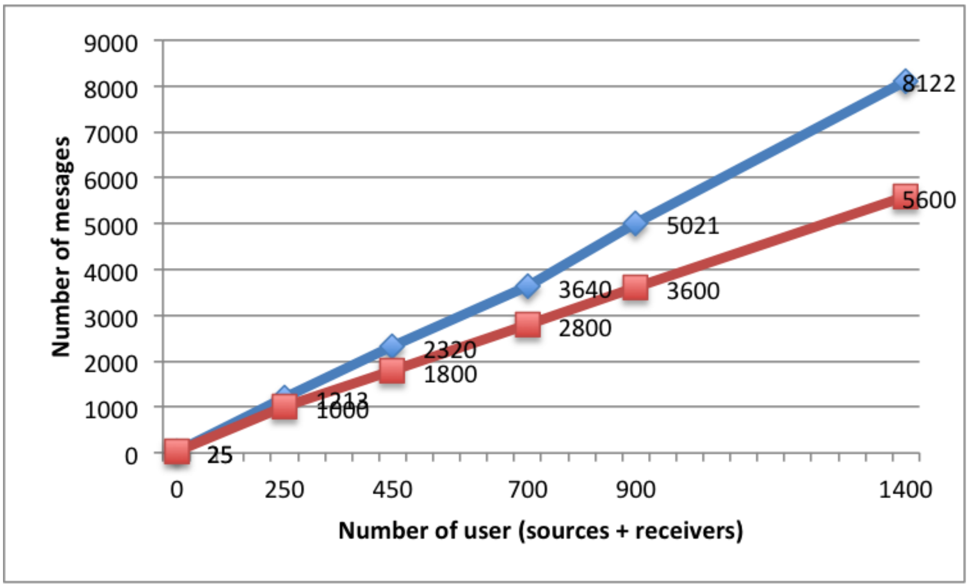


Figure 6.7: Total number of messages over number of users

This figure shows that:

- Total number of messages increases almost linearly as total number of users increases.
- Although there is difference between actual evaluated result and expected result from calculation especially when the number of users increases; however the difference gap is not big. This would be a proof that this implementation works.

  For example, when the number of users reachs highest of 1400, total number messages of actual evaluated result is 8122, while the expected result is 5600, the difference is around 1.45 times. This difference is acceptable as we consider between DHT implementation and MSDP, difference is n and O(log(n)).
- From this result we believe that this implementation would be a success to replace MSDP in terms of reducing number of messages in large-scale network.

6.3.4 Bandwidth consumption

Figure 6.8 shows the change in bandwidth consumption as number of users increases:

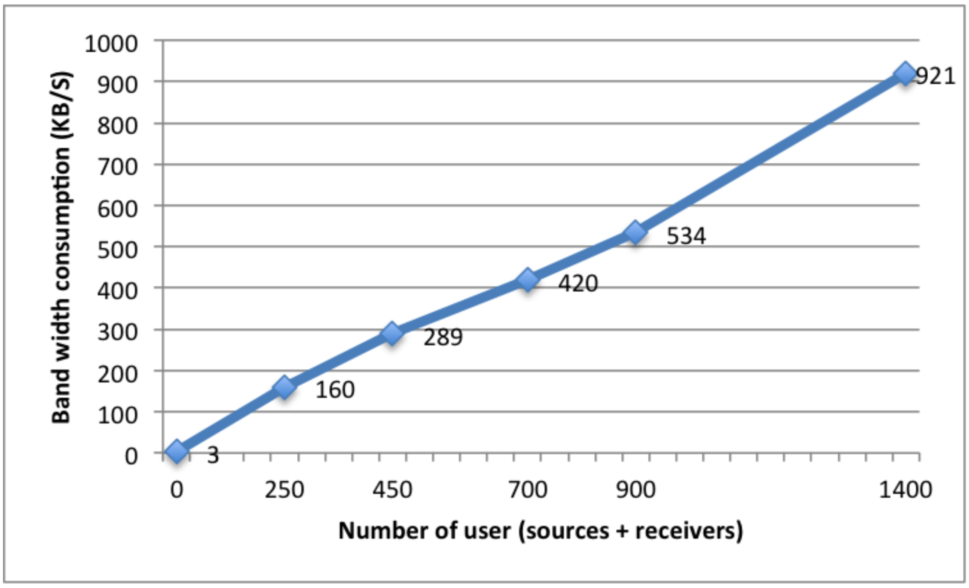


Figure 6.8: Bandwidth consumption over network

From this figure:

- Bandwidth consumption is a big problem when we consider this implementation. When number of users increase to 1400, total bandwidth consumption between two RPs network links is 921(KB/s), as average bandwidth consumption over each link is very high: around 460(KB/s). If this implementation is used in large-scale network, then bandwidth consumption would be much higher. This is the fail of this implementation.

- The reason expected would be because of the way DHT default encapsulate hash file packet and transfer it through the network. Even though the contents of the hash file would only include multicast group address and source address, packet length is still around 100 KB, which makes the bandwidth consumption very high.

- Solution to this high bandwidth consumption would be change in Bamboo-DHT packet encapsulation method, or change of other DHT implementations.

6.4 Summary

In this chapter, we discussed about evaluation of this implementation. The evaluation results show that:

- DHT maintenance network bandwidth consumption is low, and would not be a concern about scalability issue of this implementation.
- This implementation would be a success to replace MSDP in terms of reducing number of messages
- However, bandwidth consumption is a big problem to consider, as this implementation bandwidth consumption is too high compared with actual data transferred.

Chapter 7

Conclusion

7.1 Summary

This thesis focused on scalability issue on inter-domain multicast routing, which is a big challenge to multicast development nowadays. In which, MSDP, as a inter-domain multicast source discovery solution, has serious scalability issue because of its message-flooding operation.

In this thesis, we have designed DHT implementation system on PIM-SM over inter-domain network as a solution to scalability issue in MSDP protocol. In details, instead of flooding sources information to all MSDP peer router in different domains, this implementation does the source-lookup operation with the help of DHT network module, as the communication between RPs. The focusing operation of this implementation would include:

- Store sources information inside DHT network.
- Search appropriate/needed sources information from DHT network.

Through the evaluation results, we proved that DHT implementation could reduce the scalability issue in MSDP by reducing number of messages over large-scale network. However, this implementation has other problem in bandwidth consumption, which needs to be solved before deployment.

We expect that this research would give some views and new idea proposal would be a solution to scalability issue, which can help to the development of inter-domain multicast deployment.

7.2 Future work

There are at least three issues that should be discussed in the future development:

Firstly, as we discussed in Chapter 4 System Design, appropriate TTL and time-interval number must be decided properly to balance between latency and network bandwidth.

Secondly, we need to reduce high bandwidth consumption of this implementation. Suggestion would be to change the way Bamboo-DHT encapsulate packet or we try and use another DHT implementation with better encapsulating method.

Last, even though this implementation tried to solve the scalability problem in inter-domain multicast routing, we only evaluated in small-scale topology, which would give many differences in results as well as bugs when we try in a more scale-network. Future works would be evaluation in large-scale network after solving the bandwidth consumption problem.

Acknowledgement

First and foremost, I am sincerely grateful to faculty members in Murai and Tokuda Laboratory, Professor Hideyuki Tokuda, Professor Jun Murai, Associate Professor Hiroyuki Kusumoto, Professor Osamu Nakamura, Associate Professor Kazunori Takashio, Assistant Professor Noriyuki Shigechika, Assistant Professor Rodney D. Van Meter III, Associate Professor Keisuke Uehara, Associate Professor Jin Mitsugi, Lecturer Jin Nakazawa.

I would be extremely thankful to Lecturer Achmad Husni Thamrin and Dr. Kotaro Kataoka for their endless help and valuable comments to my thesis. Without their help, I cannot even think of writing this thesis.

I would also like to thank all Bianco group members, Shige-kun, Umou-kun, Shouma-kun, Nish-kun, Eden-kun, Shori-senpai, Tsune-chan, who have given me numerous support and always cheer me up in my hard times.

I own my special thank to Doctor Abazh and Dikshie for their sincerely help and many advices since the time I first came to the lab.

I would like to thank my friends Van-chan and Leorio-kun for their supports in many ways.

I would like to thank to members in Murai and Tokuda Laboratory who have been so friendly and kindly supporting me with my research.

Bibliography

[1] S. Deering. Multicast routing in internetworks and extended lans,. pages 55– 64, 1988.

[2] S. Bhattacharyya. *An Overview of Source-Specific Multicast*, 2003. RFC 3618.

[3] B. Quinn, Celox Networks, K. Almeroth, and UC-Santa Barbara. *IP Multicast Applications: Challenges and Solutions*, 2001. RFC 2458.

[4] S. Deering. Inter-domain multicast routing. 1988.

[5] Maira Ramalho and Alcatel Corporate Research Centre. Multicast routing protocols: A survey and taxonomy. *IEEE*, 2009.

[6] Satish, Kumart, and Radoslavov. the masc / bgmp architecture for inter-domain multicast routing. *IEEE*, 1998.

[7] Bin Wang Hou. Multicast routing and its qos extension: problems, algorithms, and protocols. *IEEE*, 2002.

[8] D. Waitzman, C. Partridge, BBN STC, and S. Deering. *Distance Vector Multicast Routing Protocol*, 1998. RFC 1075.

[9] J. Moy. *MOSPF: Analysis and Experience*, 1994. RFC 1585.

[10] J. Moy. *OSPF Version 2*, 1998. RFC 2178.

[11] A. Adams, J. Nicholas, and W. Siadak. *Protocol Independent Multicast - Dense Mode*, 2005. RFC 3973.

[12] A. Ballardie. *Core Based Trees (CBT version 2) Multicast Routing*, 1997. RFC 2189.

[13] D. Estrin, D. Farinacci, D. Thaler A. Helmy, S. Deering M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. *RFC 2362: Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification*, 1998. RFC 2362.

[14] B. Fenner and D. Meyer. *RFC 3618 - Multicast Source Discovery Protocol (MSDP)*, 2003. RFC 3618.