MADSH: A NEW SOLUTION FOR IP MULTICAST ADDRESS ALLOCATION

Krisztián Kiss
Ph.D. Student
Mobile Communications Laboratory, Department of Telecommunications
Budapest University of Technology and Economics, Hungary
H-1117 Budapest Pázmány P. sétány 1/D.
e-mail: krkiss@hit.hit.bme.hu

ABSTRACT

The current Internet-wide multicast routing infrastructure faces some problems. One of these limitations is that the current scheme used to assign multicast addresses to groups is only an elementary solution and does not scale well. Hence, a need has been recognized for a hierarchical multicast address allocation scheme for the Internet. This paper presents two different solutions for multicast address allocation: the first is the Multicast Address Allocation Architecture (MAAA) proposed by IETF, and the second is the Multicast Address Distribution Servers' Hierarchy (MADSH), which is planned to be the competitor of the MAAA architecture.

1. WHY TO USE IP MULTICAST?

With the increasing need of transmitting multimedia applications, such as multimedia teleconferencing, distance learning, data replication and network games, through the Internet, multicasting became a hot key topic by now. It saves bandwidth if the same data has to be transferred simultaneously to several destinations by sending only one copy of the data stream from the source, and duplicating it only at the nodes of the network, where it is really necessary: where paths to different destinations fork. IP Multicast provides efficient many-to-many data distribution in an Internet environment and also provides the functionality to logically group a set of hosts/routers.

Multicasting in IP is already in an experimental phase: there are different working multicast routing protocols, like:

- Distance Vector Multicast Routing Protocol (DVMRP) [6]
- Multicast Extension to Open Shortest Path First (MOSPF) [7]
- Protocol Independent Multicast - Sparse Mode (PIM-SM) [8]

2. MULTICAST ADDRESS ALLOCATION IN THE INTERNET

Senders to the group use the multicast address as the destination of packets to reach all the members of the group. Nowadays a multicast group initiator typically contacts an address allocation application and an address is randomly assigned from those not known to be in use. The assigned address is unique with high probability when the number of addresses in use is small, but the probability of address collisions increases steeply when the percentage of addresses in use crosses a certain threshold and as the time to notify other allocators grows. Hence, a need has been recognized for a hierarchical multicast address allocation scheme for the Internet.

2.1. Requirements for the Multicast Address Allocation Mechanisms

The important properties of the multicast address allocation mechanisms are defined in [1]. These are the robustness, timeliness, low probability of clashing allocations, and good address space utilization in situations where space is scare. These are detailed in the following paragraphs:

- Robustness/Availability: the robustness requirement is that an application requiring the allocation of an address should always be able to obtain one, even in the presence of other network failures.

- Timeliness: from a timeliness point of view, a short delay of up to a few seconds is probably acceptable before the client is given an address with reasonable confidence in its uniqueness. If the session is defined in advance, the address should be allocated as soon as possible, and should not wait until just before the session starts. It is in some cases acceptable to change the multicast addresses used by the session up until the time when the session actually starts, but this should only be done when it averts a significant problem such as an address clash that was discovered after initial session definition.

- Low Probability of Clashes: a multicast address allocation scheme should always be able to allocate an address that can be guaranteed not to clash with that of another session. A top-down partitioning of the address space would be required to completely guarantee that no clashes would occur.

- Address Space Packing in Scarcity Situations: in situations where address space is scarce, simply partitioning the address space would result in significant fragmentation of the address space. This is because one would need enough spare space in each address space partition to give a reasonable degree of assurance that addresses could still be allocated for a significant time in the event of a network partition. In addition, providing backup allocation servers in such a hierarchy, so that fail-over (including partitioning of a server and its backup from each other) does not cause collisions would add further to the address space fragmentation.

Since guaranteeing no clashes in a robust manner requires partitioning the address space, providing a hard guarantee leads to inefficient address space usage. Hence, when address space is scarce, it is difficult to achieve constant availability and timeliness, guarantee no clashes, and achieve good address space usage.

2.2. Using dynamic multicast addresses

For most purposes, the correct way to use multicast is to obtain a dynamic multicast address. These addresses are provided on demand and have a specified lifetime. An application should request an address only for as long as it expects to need the address. Under some circumstances, an address will be granted for a period of time that is less than the time that was requested. This will occur rarely if the request is for a reasonable amount of time. Applications should be prepared to cope with this when it occurs.

At any time during the lifetime of an existing address, applications may also request an extension of the lifetime, and such extensions will be granted when possible. When the address extension is not granted, the application is expected to request a new address to take over from the old address when it expires, and to be able to cope with this situation gracefully. As with unicast addresses, no guarantee of reachability of an address is provided by the network once the lifetime expires. These restrictions on address lifetime are necessary to allow the address allocation architecture to be organized around address usage patterns in a manner that ensures addresses are aggregable and multicast routing is reasonably close to optimal. In contrast, statically allocated addresses may be given sub-optimal routing.

3. MAAA: MULTICAST ADDRESS ALLOCATION ARCHITECTURE

The Internet Engineering Task Force (IETF) has established a new working group (MALLOC - Multicast Address aLLOCation) for elaborating the multicast address allocation protocols. The development of the protocols is in progress, the first Internet Draft documents were born in 1998. The proposed MAAA architecture [1] is three layered, comprising a Client-Server protocol (MADCAP), an intra-domain protocol (Multicast AAP) and an inter-domain protocol (MASC). The architecture can be seen in Figure 1., the protocols are detailed in the next three paragraphs.

- Multicast Address Dynamic Client Allocation Protocol (MADCAP) [2] is a protocol that allows hosts to request multicast address allocation services from Multicast Address Allocation Servers (MAAS). MAAS servers can allocate individual multicast addresses to groups initiated in their domain. MADCAP is built on a client-server model, where hosts request address allocation services from address allocation servers. When a MADCAP client wishes to request a service, it unicasts or multicasts a message to one or more MADCAP servers, each of which optionally responds with a message unicast to the client.

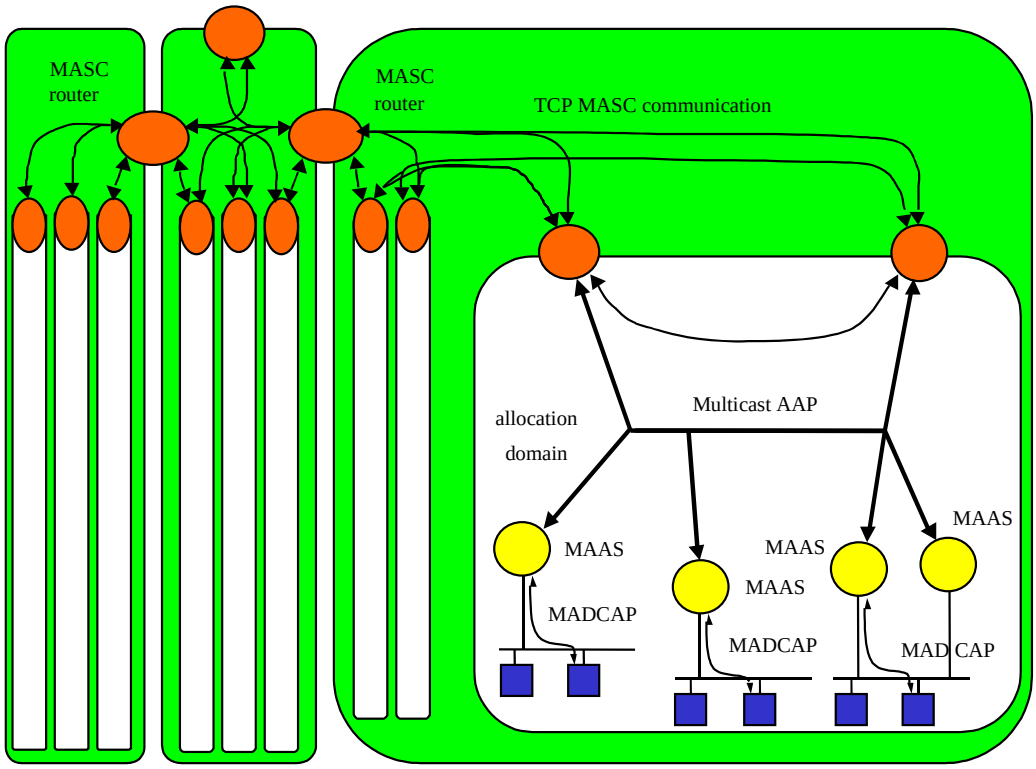

Figure 1. MAAA architecture

- Multicast Address Allocation Protocol (Multicast AAP) [3] is used by a MAAS server to claim multicast addresses that it has allocated, and if necessary to defend these addresses if another MAAS server attempts to allocate the same address. A MAAS server keeps track of all the other multicast addresses in use within the same allocation domain, and when it allocates an address it ensures that the address is not already in use. AAP is also used by routers performing MASC to inform the MAAS servers of the address set (consisting of a list of address/mask/lifetime) that is available.

- Multicast Address Set Claim (MASC) [4] forms the top level for the hierarchical address allocation architecture. MASC is used by routers to claim address ranges that satisfy the needs the MAAS servers within their allocation domain. The domains running MASC form a hierarchy based on the structure of the existing inter-domain topology. MASC then dynamically allocates address ranges to domains using a “listen and claim with collision detection” approach. In this approach, child domains listen to multicast address ranges selected by their parent, select subranges from their parent's range and propagate the claims to their siblings. The claimers wait for a suitably long period to detect any collision, before communicating the acquired range to the domain's MAAS servers. When a MASC router discovers that there are close to insufficient multicast addresses available for AAP to perform well, the MASC router claims a larger address range. Address ranges have also a lifetime assigned and that lifetime cannot be longer than the lifetime of the parent address range.

4. PROBLEMS WITH THE MAAA ARCHITECTURE

The MAAA architecture cannot strictly guarantee that every address request will be honored in a short time [5]. If a client wants to allocate a new multicast address and there is no available address in the allocation domain, then the MASC router has to send a claim to one of its parents for address range extension. This operation may be considerably long. Consequently, the multicast applications that cannot rely on the latency offered by the MAAA architecture would necessarily have to allocate addresses ahead of time. This means that we need “pre-allocated addresses” that are removed from the set of globally available addresses but are not given immediately to an application. The advantage of pre-allocating addresses is that the MASC routers can honor

requests without communicating with the parent MASC router. On the other hand, the disadvantage of pre-allocating addresses is the lower address utilization; i.e. it is possible, that there are no more available addresses while in fact not all of them are used. In this paper an architecture is suggested, which is built on a so-called MADS servers' hierarchy. In this proposal the addresses are statically bound to the MADS servers, which are hierarchically structured, just like in the Domain Name Server (DNS) architecture.

5. MULTICAST ADDRESS DISTRIBUTION SERVERS' HIERARCHY (MADSH)

This chapter presents a different solution for IP multicast address allocation, which significantly differs from the MAAA architecture. The proposed Multicast Address Distribution Servers' Hierarchy [9] can be considered as a competitor of the IETF's MAAA architecture. The aim of the architecture is the same like the aim of the MAAA architecture, to provide the management of the Class D IP address range. The MADSH architecture has been made according to the requirements defined in chapter 2.1 and it also uses the dynamic allocation of the multicast addresses but the different design approach results in significantly different behavior.

5.1. The hierarchy of the MADS servers

The MADSH architecture can be considered as the multicast extension of the DNS hierarchy. In the hierarchically organized architecture, similarly to MASC architecture, we can differentiate between top, middle level and bottom level Multicast Address Distribution Servers (MADS's). The top and the middle level servers are responsible for allocation the multicast address ranges, the bottom level servers provide the multicast addresses to the clients. The difference between the functioning of the MAAA and the MADSH architectures is that in the MADSH architecture a parent MADS server provides a static size address range to their children, while in the MASC architecture a child MASC router has to claim an address range with a dynamically allocated size from the parent router. This solution has the advantage over the MAAA architecture that it is very simple, and the time between the allocation and the distribution of the multicast address is as short as possible.

The Top Level Servers (TLS) have the Class D IP multicast address range. They also store which subranges are the their child servers responsible for. This principle is further effective going down in the hierarchy. A client is configured with the unicast address of a bottom level MADS server and in case of a multicast address allocation the client has to send its claim to this server.

The next figure contains an example for the MADSH hierarchy:

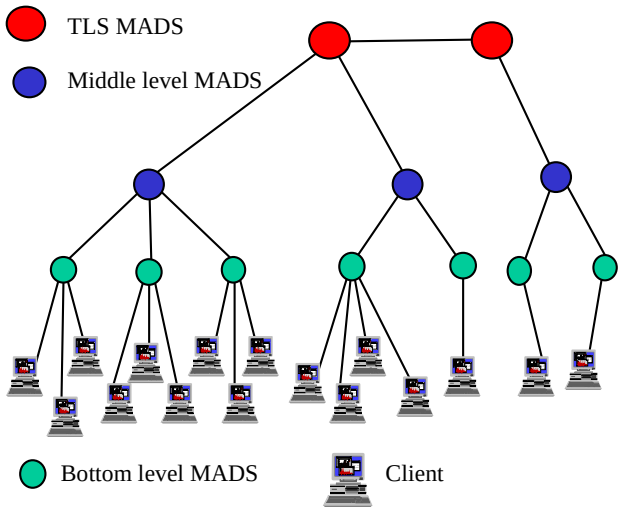

Figure 2. Example for the MADSH architecture

5.2. Forwarding the Multicast Address Allocation Request

The main problem is in this architecture to solve the following problem: what happens if a child server's static size address range, which was offered by its parent server, becomes exhausted? In the MAAA/MASC architecture the MASC router has the possibility to ask address range extension from its parent router but this is not possible in case of static size. In the MADSH architecture if a client wants to allocate a multicast address from a bottom level MADS server and the server has already distributed all the available multicast address for the clients, the server must forward the multicast address allocation request to one of its siblings. Every server in the hierarchy (except the TLS servers) has to distribute periodically to its parent server the number of the unallocated multicast addresses. If a bottom level MADS server cannot serve a multicast address allocation request due to the lack of free addresses, then forwards it to its parent. The parent MADS server observes from the periodically sent status messages if its child servers have any free multicast addresses.

- If there is more than one server, which has free multicast addresses, the parent selects the server has the most addresses, and forwards the multicast address allocation request to it. Henceforth the client has to send its further messages related to the management of the

allocated multicast address to this server.

- If the child servers do not have any free multicast address, the parent server forwards the request to its parent.

The following figure represents an example for the multicast address allocation request forwarding:

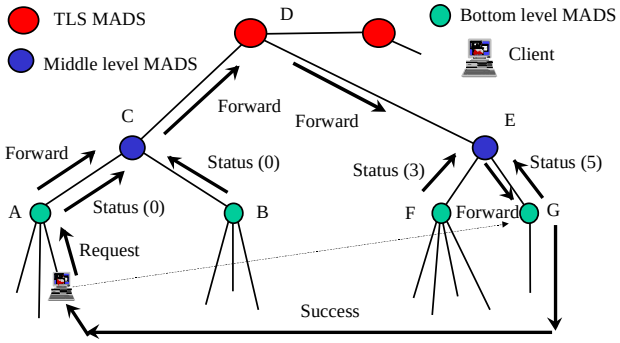

Figure 3. Multicast address request forwarding in MADSH architecture

Since the address ranges of Server A and B are exhausted, Server C forwards the multicast address allocation request to Server D. Server F and G have free available addresses, and since Server G has the most, the "Forward" request arrives there.

5.3. Reinitialization of the address ranges

In course of initialization of the MADSH architecture the TLS servers assign to the middle level servers (and also the middle level servers to the bottom level servers) multicast address ranges from its own multicast address range. The initial size of a bottom level server's address range can be based on the number of the clients, which are configured with the unicast address of the server. If the number of the clients in a physical region significantly increases, the MADSH architecture provides the possibility to reinitialize the static size of the address ranges in the architecture in order to give a bigger address range to a MADS server. The reinitialization means that the MADS server claims the unallocated address ranges from its siblings, so it is at the expense of the other low level MADS servers. It is important to note that in case of temporary increase of the clients the MADS server can solve the problem with the mentioned address allocation request forwarding, there is no need to reinitialize the address ranges. Another thing is that the address ranges have lifetimes, which also increase the reusability of the address ranges.

5.4. The protocols in the MADSH architecture

The functioning of the architecture is provided by two protocols:

1. The Multicast Address Client-Server Protocol (MACSP) provides the clients to allocate multicast addresses from the MADS server. The protocol works according to the request-response model, the client unicasts its information message to the configured address of the MADS server, the server answers it with a unicast message.

   There are four kinds of requests:
   - Multicast Scope Information (MSI) to discover the available multicast scopes
   - Allocate Multicast Address (AMA)
   - Deallocate Multicast Address (DMA)
   - Renew Multicast Address (RMA)

   There are three kinds of responses:
   - Success: indicating that the request was executed successfully
   - Transient error: indicating that the request cannot be executed because of some reason. The client should retry its request after a definite time.

   Since the protocol is planned to be working over UDP, the client must acknowledge the server's response with an ACK message.

2. The Multicast Address Server-Server Protocol (MASSP) provides the communication between MADS servers. There are four kinds of messages between MADS servers:

   - Status message: this message is sent periodically by a MADS server to its parent server and contains the number of the unallocated multicast addresses by the MADS server.

   - Forward message: this message is sent by a bottom level MADS server to its parent server when an AMA message arrives from a client and the server has already distributed all the available multicast addresses for the clients. The behavior is the following:
     - If the children of the middle level MADS server do not have any available multicast addresses (the middle level and TLS MADS servers keep a database about the number of the free addresses available at the children according to the received Status messages) then forwards this message to its TLS server. If the

children and the siblings of the TLS MADS server do not have also any available multicast addresses (the whole Class D address space is currently in use by the clients) then the TLS MADS server should send an "All Addresses In Use" to the server who initiated the "Forward" message.
    - If one or more children of the middle level MADS server have available multicast addresses, the server selects the one who has the most, and forwards the "Forward" message to it.

- All addresses in Use: if a middle level MADS server receives this message as a response for the "Forward" message sent to the TLS server, it will forward the message to that bottom level server, who initiated the "Forward" message. Receiving this message the bottom level server should send a "Transient Error" message to the client indicating in it that the whole multicast address range is allocated, the client should wait until the lifetime of one multicast address expires.

- Reinitialization: this is an optional message if the number of the "Forward" messages exceed a certain limit in a defined time interval. The message is sent by a middle level or a TLS MADS server to all the child servers. Receiving the "Reinitialization" message the server should send a special "Status" message containing the available free addresses (not only the number of them). After receiving this information the server rearranges and reallocates the static size multicast address ranges for its child servers and sends them in a new "Reinitialization" message. The clients served by the child servers do not notice anything from the process, because the reinitialization concerns only the free addresses.

5.5. Signaling diagrams for different address allocation cases

Figure 4 contains an example for using the MACSP and MASSP protocols in the MADSH hierarchy if it is not necessary to forward the multicast address allocation request:

1. In course of initialization the middle level MADS server allocates a multicast address range for the bottom level MADS server. This could happen with the "Reinitialization" message.
2. The client optionally can discover the available multicast scopes with a MSI message.
3. The client selects one scope, which is suitable for the size of the required multicast group, and sends an AMA message to allocate one multicast address from the requested scope for a special lifetime.
4. For renewing or deallocating the address the client sends a RMA or a DMA message to the server.
5. The "Status" message is periodically sent by the bottom level MADS server to the middle level MADS server about the status of the number of the free addresses.

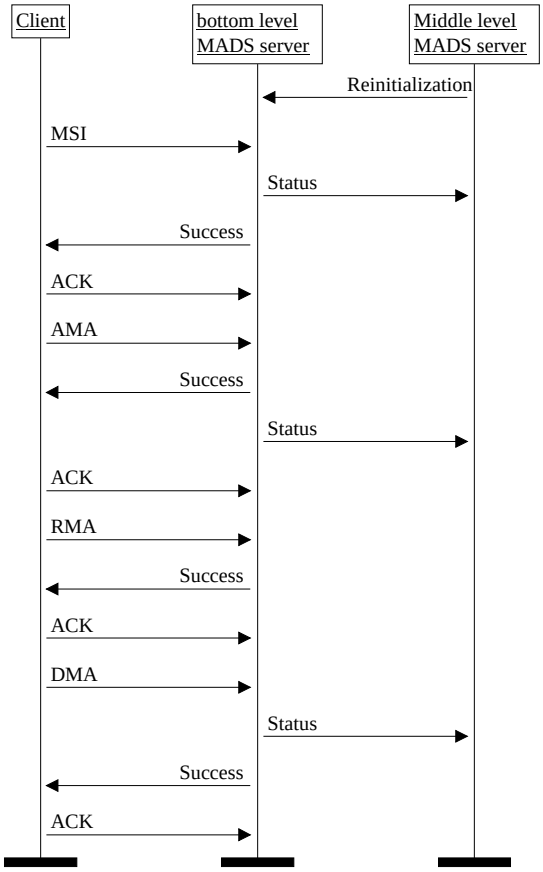

Figure 4. The message exchanges if it is not necessary to forward the multicast address allocation request

In Figure 5 an example is presented for message exchanges of the MACSP and the MASSP protocols if the serving MADS server forwards the multicast address allocation request to one of its siblings. The figure represents the behavior of the client: it sends its RMA and DMA messages to that server, from where it receives the "Success" message.

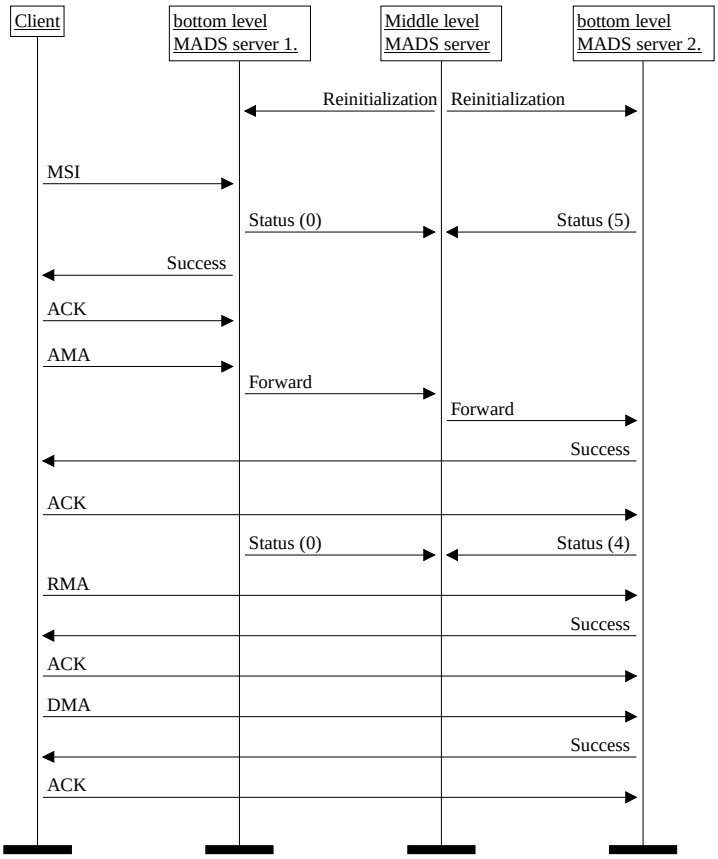

Figure 5. The message exchanges if it is necessary to forward the multicast address allocation request

6. CONCLUSION AND FURTHER WORK

Because of the statically assigned method, the functioning of the MADSH architecture is simple and fast, there is no need to have complicated address allocation algorithm, or three-layered protocol-hierarchy. There are no pre-allocated addresses, therefore it provides better address utilization than the MAAA architecture.

The protocol proposal is under development by this time together with a comparison with the IETF's MAAA architecture. The development of the new protocol is in the phase of textual description, some theoretical and practical problems are still unsolved. Under the following formal description and verification phase hopefully it will turn out that the principles are correct in the textual description.

REFERENCES:

[1] M. Handley, D. Thaler, D. Estrin, “The Internet Multicast Address Allocation Architecture”, Internet Draft, University of Michigan, MALLOC WG, January 2000, draft-ietf-malloc-arch-04.txt

[2] B. V. Patel (Intel), M. Shah (Microsoft), S. R. Hanna (Sun), “Multicast Address Dynamic Client Allocation Protocol (MADCAP)”, RFC 2730, MALLOC WG, December 1999

[3] M. Handley, S. R. Hanna, “Multicast Address Allocation Protocol (AAP)”, Internet Draft, March 2000, draft-ietf-malloc-aap-03.txt

[4] D. Estrin, R. Govindan, M. Handley, S. Kumar, P. Radoslavov, D. Thaler, “The Multicast Address-Set Claim (MASC) Protocol”, Internet Draft, MALLOC WG, January 2000, draft-ietf-malloc-masc-05.txt

[5] G. Phillips, M. Smirnov, “Address utilization in the MASC/BGMP architecture”, Internet Draft, July 1998, draft-phillips-malloc-util-00.txt

[6] T. Pusateri, “Distance Vector Multicast Routing Protocol”, Internet Draft, September 1999, draft-ietf-idmr-dvmrp-v3-09.txt

[7] J. Moy, “Multicast Extensions to OSPF”, RFC 1584, March 1994

[8] L. Wei, D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, “Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification”, Internet Draft, November 1999, draft-ietf-pim-v2-sm-01.txt

[9] K. Kiss, “IP multicast and the multicast address allocation mechanisms”, M. Sc. Thesis work, Technical University of Budapest, 1999