# Multicast for Small Conferences - A Scalable Multicast Mechanism Based on IPv6

Stefan Egger and Torsten Braun
Institute of Computer Science and Applied Mathematics,
University of Bern,
Neubrückstrasse 10, CH-3012 Bern, Switzerland

Many new Internet applications require data transmission from a sender to multiple receivers. Unfortunately the IP Multicast technology used today suffers from scalability problems, especially when used for small and sparse groups. Multicast for Small Conferences (MSC) is a novel approach aimed at providing more efficient support for example for audio conferences. It makes use of an IPv6 routing header. The unicast addresses of a small receiver group are put into that extension header in order to be forwared to all group members. The results indicate that MSC has the potential of replacing IP Multicast for many delay sensitive small group applications, even with very limited support from the network infrastructure.

## 1 Introduction

As the Internet more and more replaces traditional telecommunication networks, it seems very promising to use IP Multicast to simplify audio conferencing, which is complex to set up and very inefficient regarding bandwidth usage when conventional technology is used. To support a conference, the IP terminals and gateways serving the conferencing participants can join a common multicast group and exchange traffic via IP Multicast. This avoids the multiple transport of the same traffic over the backbone networks that is seen in traditional telephone conferences based on Multipoint Control Units (MCUs).

Unfortunately, IP Multicast does not scale well for (many) small groups such as in audio conference scenarios. The problem is that the multicast routing entries - which are required to be maintained by all routers that are part of a multicast forwarding tree - cannot be aggregated such as unicast routing entries. Multicast address selection

is arbitrary, so that multicast addresses with similiar prefixes do not necessarily have any relation to each other such as a common delivery tree. The scalability problem is even worse since multicast routing entries do not only consist of the destination address but may even include a source address. As new small group applications are becoming more and more popular, the routing table sizes are increasing massively. This not only makes routers more expensive (given the high prices of router memory), but also deteriorates the performance of these devices.
Several proposals addressing the problems of IP Multicast have arised recently [1,2,4,7]; One of these - Xcast - is described briefly in Section 2. Section 3 describes the Multicast for Small Conferences concept based on IPv6 extension headers. We present the simulation topology, parameters and scenarios used for a basic performance evaluation of the concept in Section 4. The simulation results are discussed in Section 5, while the major conclusions of the study are presented in section 6. Section 7 summarizes the paper.

# 2 Explicit Multicast (Xcast)

Explicit Multicast [1] (the successor of Small Group Multicast [2]) is a multicast scheme designed for supporting a very large number of multicast sessions as present in audio/video conferencing, network games or collaborative working. It differs from native multicast in that the sending node keeps track of all session members and explicitly encodes the list of destinations in a special packet header. This newly defined header introduces a new protocol between the network (IP) and the transport (UDP/TCP) layer. Xcast capable routers that receive such a packet parse the Xcast header and use the ordinary unicast routing table to determine how to route the packet to each destination, generating a packet copy for every affected outgoing interface. Each address list contains only the addresses that can be reached via that interface. If there is only one destination address for a particular next hop, the packet may be sent as a standard unicast packet.

# 3 Multicast for Small Conferences

## 3.1 MSC Protocol Overview

The Multicast for Small Conferences (MSC) concept aims at solving the scalability problem of native multicast by explicitly carrying all destination addresses in the data packets while at the same time avoiding the problems of Xcast.
A first difference between the two approaches is that MSC has been proposed as a concept for multicast packet delivery in the Internet backbone only, while existing intra-domain multicast routing mechanisms can remain in use for regional or access networks. This is achieved by using MSC gateways, which have the task of supporting end systems not capable of MSC and/or IPv6 by "translating" the transmissions, using native multicast in the local network.
Also, instead of introducing a new protocol, MSC relies solely on the existing IPv6

protocol, in particular on the IPv6 routing header. With MSC, the unicast address of each receiver is stored in each packet. A sender, which is either an MSC terminal or a gateway, will create a unicast address list of all group members and put the nearest one in the IPv6 destination address. All other member addresses are stored in the MSC routing header, preferably ordered by the distance from the sender. The group's multicast address should ideally be stored in the routing header as well. However, according to the IPv6 specification [3], multicast addresses must not appear in a routing header of type 0, or in the IPv6 destination address field of a packet carrying such a routing header. There are two ways to overcome this limitation. The first (short-term) solution is compatible with all current IPv6 routers, while the second solution is intended for long-term usage.

In the first solution the group's multicast address is carried in a newly defined IPv6 destination option (Figure 1). The destination options header is used to carry information that needs to be examined only by a packet's destination node(s). The destination options header is identified by a next header value of 60 in the preceding header. The options field is of variable length, but must be an integer multiple of 8 octets long (Figure 2). It contains one or more TLV-encoded (TLV: type, length, value) options.

In the second solution the multicast address is carried at the end of a newly defined type 1 IPv6 routing header. The syntax of this routing header is exactly the same as for the type 0 routing header but with type=1. The difference to the first solution is that now the IPv6 multicast address is located at the end of the address list in the routing header instead of an option field (Figure 3). If members have to be reached via different outgoing interfaces, a packet for each affected interface is generated with the list of members that can be reached via this interface. This means that the sender divides the address list into $N$ parts and sends $N$ copies of the packet to the $N$ generated lists.

A receiving end system which finds its address in the header creates a packet for the higher protocols encapsulated in the IPv6 packet by copying the multicast address into the IPv6 destination address and removing the routing header. An MSC gateway forwards the packet to local multicast receivers using the appropriate scope. If the routing header contains further unicast addresses, a new packet is generated with the address of the nearest node in the IPv6 destination address. As before, a routing header carries the remaining unicast addresses.

A router that does not understand the MSC header forwards the packet towards the address specified in the IPv6 destination field. MSC capable routers read the addresses from the destination field and the routing header and determine the outgoing interface for each destination. They then duplicate the packet for each involved link. Again, each packet contains only the unicast addresses that can be reached via that interface plus the multicast address identifying the group. In this document, this router behavior is denoted *standard MSC*.

For the illustration of this mechanism let us assume the topology depicted in Figure 4, a sender at Hamburg (HAM), as well as two receivers at New York (NYC) and Torino (TO). The sender inserts two destination addresses (one in the basic header and the other in the routing header) and transmits the packet. MSC router DE detects the extension header, duplicates the packet (one for New York and one for Torino) and

forwards the resulting ones towards their particular destinations.
A possible improvement of the basic MSC concept involves the use of topology information, which can for example be obtained from a link state routing protocol such as OSPF. The first MSC router that handles an MSC packet after it enters a certain network domain (e.g. a backbone network) determines the egress router (i.e. the router where the packet leaves the domain) for the destination address and all addresses listed in the routing header. A packet is then created for each involved egress router. Thus, packet forwarding between destinations connected to the same network can be eliminated, which potentially reduces the delays. On the downside, multiple packets may be sent over the same link, if two or more egress routers are reached via the same outgoing interface. In this document, this advanced concept is denoted *enhanced MSC* (EMSC).

## 3.2 Comparison with Xcast

Although MSC is based on ideas similar to Xcast, there are significant differences:

- MSC avoids introducing a new protocol and instead relies solely on IPv6.
- MSC requires no tunneling between gateways, as routers without MSC functionality can simply forward the packet according to the IPv6 destination address. This also simplifies a gradual deployment of MSC in the network.
- MSC uses unicast forwarding in the backbone; multicast routing can be retained in local networks using MSC gateways.
- MSC allows applications to use native IP Multicast. Gateways only need to insert an MSC routing header instead of doing complete address mapping as in Xcast. Therefore the same multicast address can be used at different sites without the need for synchronizing the gateways.

These items are an indication that Multicast for Small Conferences is less complex to introduce and use than Explicit Multicast. Similar IPv6 based mechanisms as MSC have been proposed in [1] and [4], but they do not propose to carry multicast addresses.

## 3.3 Problems

Multicast for Small Conferences suffers from the following problems:

- The IPv6 routing header creates overhead that is increasing with the group size. This might be a problem for audio applications, where the packets are usually relatively short. Severe complications might emerge in wireless networks. The overhead problem can be solved by gateways serving as MSC receivers and forwarding the received packets via native IPv6 Multicast to the other receivers after discarding the routing header.

- MSC is an IPv6 only solution and requires the MSC routers and gateways to support IPv6; solutions such as IP options have to be found to support IPv4 end systems.
- All senders need to know the IPv6 unicast address of the group members. This problem can be solved by a group control protocol by which the MSC receivers announce conference group membership to each other. This information might be distributed within session descriptions of the Session Description Protocol (SDP) by which session descriptions are distributed over a well-known multicast address.
- While reducing the routing table size by avoiding the use of Multicast addresses, MSC increases the complexity of the routing process: When handling an MSC packet, an MSC capable router has to perform multiple lookups, one for each unicast address carried in the packet. Furthermore, the router cannot simply create packet copies, but has to make sure that each packet only contains the addresses that can be reached via a certain interface.

# 4 Simulation

## 4.1 Simulation Topology

In order to evaluate the performance of Multicast for Small Conferences, the protocol has been implemented in the ns-2 network simulator [5]. This software was subsequently used for a basic performance study of Multicast for Small Conferences. Due to the similarity of MSC and Xcast the results can be applied to Xcast as well.
Since choosing an appropriate topology is critical for useful results, the simulations were based on real-world information. Since MSC has been proposed for use in backbones, the simulation scenarios were based on information from actual Internet backbone networks. Particularly, the structure of the simulation topology was formed on the basis of five research networks: The Pan-European Gigabit Research Network (Géant), the Italian Academic and Reseach Network (Garr), Abilene, the Swiss Academic and Research Network (Switch) and the German Research Network (DFN/G-WiN).
For realistic data about end systems, web servers of universities connecting to the Internet via the selected backbone networks were used. Using these systems makes sense since well-known names (`www.universityname.edu`) can be used and no special information (IP addresses or hostnames) is required. Furthermore, the delay between the web server and any other host in the university networks should be negligible.
In order to collect information about the actual delays in the networks and the routing behavior, extensive `ping` and `traceroute` measurements were performed from the end systems and looking glasses in the backbone networks.
The information collected from the backbone networks and the end systems resulted in an ns-2 topology consisting of 78 nodes and 89 links (Figure 4).

## 4.2 Simulation Parameters

In order to obtain information about MSC's sensitivity to group size and clustering, fifteen different sets of end systems were defined. These include combinations of five different group sizes (4, 8, 12, 16 and 20 hosts) and three degrees of spatial locality (clustering). Each of these end system sets was run in eight different configurations:

**Native multicast** IP Multicast (PIM-SM)

**Naive unicast** Unicast transmission from the sender to all recipients.

**End system MSC** All end systems are MSC capable, but there are no MSC routers. The senders order the destination addresses by distance.

**Full-scale MSC** Standard MSC functionality is deployed in all end systems and backbone routers.

**MSC at backbone interlinks** All end systems, and all nodes (routers) with a link to another network domain have MSC functionality. This scenario was simulated for both standard and enhanced MSC.

**MSC SIX** In this scenario, only six routers (`LAX`, `KCK`, `NYC`, `DE`, `CH` and `MI`) in the topology are considered MSC capable (independent of the group structure). However, in contrast to the previous configuration, these are evenly distributed over the topology. This scenario was also simulated for both standard and enhanced MSC.

All simulations involve each end systems sending a single packet to the group. Packet size is calculated on the basis of an audio transmission with 80 bytes payload over RTP, UDP and IPv6.

## 4.3 Evaluation Metrics

For the evaluation, several metrics were used. The *maximum delay* found in a scenario can be used to decide whether an audio conference is feasible with the selected parameter; a usual threshold is 150ms. The *average delay* measured from all sender-receiver delays is an indication of the overall performance of a given configuration. These values are used to calculate the Relative Delay Penalty (RDP) values shown in Table 1. The RDP is the ratio of the MSC delay between two hosts to the IP Multicast delay between them. Another important factor is *bandwidth consumption*. For this performance evaluation, the overall link usage in the scenarios is measured. These values are used to calculate the Normalized Resource Usage (NRU), shown in Table 2. It should be kept in mind that the difference between MSC configurations and native multicast is partially caused by the longer IP header.

# 5 Performance Evaluation Results

## 5.1 Parameters

Tables 1 and 2 summarize the relative delay penalty, the normalized resource usage, and the backbone usage for the various multicast mechanisms. More detailed performance results and graphs can be found in [6].

| Configuration | min | max | avg | EXD |
|---|---|---|---|---|
| Native multicast | 1 | 1 | 1 | 0% |
| Naive unicast | 1 | 1 | 1 | 0% |
| End system MSC | 1.1 | 4.3 | 2.7 | 44% |
| Full-scale MSC | 1 | 1.1 | 1 | 0% |
| MSC at interlinks | 1 | 2.4 | 1.6 | 15% |
| EMSC at interlinks | 1 | 2.2 | 1.5 | 14% |
| MSC SIX | 1 | 1.9 | 1.4 | 6% |
| EMSC SIX | 1 | 1.6 | 1.3 | 1% |

Table 1: Relative Delay Penalties (RDP) of the average delays, and percentage of delays >150ms (EXD)

| Configuration | min | max | avg |
|---|---|---|---|
| Native multicast | 1 | 1 | 1 |
| Naive unicast | 1.4 | 3.7 | 2.5 |
| End system MSC | 1.3 | 3.3 | 2.4 |
| Full-scale MSC | 1.2 | 1.6 | 1.4 |
| MSC at interlinks | 1.3 | 2.3 | 1.8 |
| EMSC at interlinks | 1.3 | 2.3 | 1.8 |
| MSC SIX | 1.3 | 2.1 | 1.7 |
| EMSC SIX | 1.3 | 2.1 | 1.7 |

Table 2: Normalized Resource Usage (NRU)

The results of native multicast form the basis of the performance evaluation. In terms of delay, native multicast is insensitive to group size and clustering, because packets are always forwarded along an optimal tree. The bandwidth consumption is low, since there is no unnecessary packet duplication. Naive unicast suffers from high link stress, i.e. many identical packet copies are sent over the same links. In combination with the excessive bandwidth consumption, this disqualifies the concept from being an alternative to IP Multicast.

End system MSC suffers from very high delays due to the packet forwarding between receiving end systems. In terms of bandwidth consumption, end system MSC also shows poor performance. While bandwidth consumption increases linearly for native multicast, it grows exponentially in the case of end system MSC. Due to the optimal

forwarding paths, packet forwarding between end systems is reduced to a minimum for full-scale MSC. This shows in almost no delay penalty compared to native multicast and naive unicast. In terms of bandwidth consumption, full-scale MSC suffers from the routing header, which increases the packet size.
In terms of delays, (E)MSC at backbone interlinks perform significantly better than end system MSC. Unfortunately they also perform a lot worse that full-scale MSC or native multicast. Especially in scenarios with large groups, the uneven distribution of MSC functionality and the resulting packet forwarding between end systems severly deteriorates the performance. Compared to the previous configuration, (E)MSC SIX yields an improvement in delays, as the number of delays in excess of 150ms is significantly lower. Also, a slightly lower bandwidth consumption has been measured.
The results of these two configurations prove that MSC can deliver an acceptable performance with just a few MSC capable routers, at least for smaller groups, even with widely spread group members. Compared to full-scale MSC lower deployment costs (less routers) have been traded against higher delays and increased bandwidth consumption. The comparison of (E)MSC SIX against (E)MSC at backbone interlinks proves that router distribution is critical.
In terms of delay, MSC can only achieve performance similar to native multicast when full-scale MSC is used. With an optimized "intermediate" approach such as in the (E)MSC SIX scenarios, a delay penalty of up to 90% (60%) with an average of 30% (40%) has to be accepted. However, depending on group size and clustering, the difference may be significantly smaller. Due to the longer IP header, packet duplication and packet forwarding between end systems, MSC also has a higher link usage than native multicast.

# 6 Lessons Learned

The simulation results and their interpretatation provide the basis for a number of conclusions on the concept of Multicast for Small Conferences:

1. **End system MSC is not feasible.** End system MSC produces unacceptably high delays in almost all scenarios. Given the 150ms delay limit, supporting audio conferences is only possible for very small groups with strong spatial locality. This does *not* mean that end system multicast does not work. It merely proves that end system multicast on IP level without router support cannot deliver the performance required to support delay-sensitive applications. A possible solution is the use of an application-level multicast scheme such as Narada [7], at the cost of increased bandwidth consumption and link stress.

2. **Only a small number of MSC capable routers is required to significantly improve the performance.** The results of the (E)MSC at backbone interlinks and (E)MSC SIX configurations impressively prove this point. This leads to the following conclusions:

    - **There is no need for full-scale MSC.** A very good performance can be delivered with just a few MSC capable routers at considerably lower deployment costs.

- **Gradual MSC deployment is possible.** Assuming the availability of appropriate end systems, MSC can start at a low performance level with no or very few dedicated routers. The performance can then be gradually improved by the deployment of additional MSC routers.

3. **MSC router distribution is critical.** The fact that the MSC SIX configuration (with its more evenly distributed routers) yields better results than the MSC at backbone interlink setup is a good indication of this. It seems logical to place MSC routers on nodes interconnecting backbone networks, particularly because these nodes have a high traffic load. However, with several members of a group connecting to the same backbone network (or even the same router), this approach can result in high delays as packets are forwarded between end systems without encountering an MSC router. Two arguments affect the development of rules for MSC router placement:

   - The (potential) performance of a *standard* MSC router depends on the number of outgoing interfaces. The more packet copies a standard MSC router can create (at most one per outgoing link), the less addresses remain in the various routing headers, reducing the necessary forwarding between end systems accordingly. In contrast, the performance of *enhanced* MSC routers is not affected by the out-degree of the node, as the number of packet copies created only depends on the number of egress routers involved.
   - Once an MSC packet has been processed by an enhanced MSC capable router the optimal forwarding strategy through that domain is executed for all destination addresses. Thus, enhanced MSC is most effective when MSC packets are intercepted as early as possible after entering a certain network domain. This is not necessarily true for *standard* MSC, since the routers have no knowledge of possible branching further down the forwarding tree.

   Unfortunately these two rules alone do not guarantee a good performance. For example, a standard MSC router with a lot of outgoing links will not be helpful if the relevant traffic bypasses the node. On the other hand, trying to intercept all packets entering a domain as early as possible would almost certainly result in full-scale MSC. Thus, a compromise has to be found. The most obvious solution is to distribute the MSC functionality as evenly as possible over the entire network (similar to the (E)MSC SIX configuration). This maximizes the probability that a packet encounters an MSC router, thus reducing the potential for packet forwarding between end systems. The performance advantage of the (E)MSC SIX configurations indicates the potential of this approach. However, it should be kept in mind that the actual performance in a scenario also depends on other factors such as group size and clustering.

4. **Enhanced MSC performs better than standard MSC.** The main advantage of enhanced MSC are the reduced delays. In scenarios with a small number of hosts (four/eight), enhanced MSC shows the same performance as standard

MSC. For large groups however (16 or 20 hosts), enhanced MSC produces lower (10-20ms) average delays. This is the direct result of much more efficient packet forwarding, as can also be seen from the massively reduced percentage of delays in excess of 150ms (1% vs. 6%).
Another difference between standard and enhanced MSC is visible in the backbone usage. In each scenario, enhanced MSC consumes about the same amount of bandwidth as standard MSC. The difference lies in *where* it is consumed: enhanced MSC burdens backbone links with duplicate packets (addressed to different egress routers), but in turn exonerates the access networks by ensuring that they never have to "return" an incoming packet to the backbone network. Depending on the network infrastructure, this feature may be considered as an advantage or a disadvantage.

# 7 Conclusions

In this paper we have presented initial performance evaluation results of Multicast for Small Conferences (MSC). The simulation results with ns-2 indicate that supporting delay-sensitive applications without dedicated routers is not feasible. However, we have shown that already a small number of MSC routers can significantly improve the performance of the concept. We have also compared two different variants of Multicast for Small Conferences, and shown that an enhancement of the basic concept with topology information (obtained for example from a link state routing protocol) allows a significant gain on performance in terms of delay and bandwidth consumption.

# References


[1] R. Boivie et al. Explicit Multicast (Xcast) Basic Specification, Internet Draft, work in progress, January 2003.

[2] R. Boivie, N. Feldman, and Ch. Metz. Small Group Multicast: A New Solution for Multicasting on the Internet. *Internet Computing*, 4(3), May/June 2000.

[3] S.Deering and R.Hinden. Internet Protocol, Version 6 (IPv6) Specification, RFC 2460, December 1998.

[4] Imai Yuji. Multiple Destination Option on IPv6 (MDO6), Internet Draft, work in progress, draft-imai-mdo6-01.txt, March 2000.

[5] Ns-2 website. `http://www.isi.edu/nsnam/`.

[6] Stefan Egger and Torsten Braun. Performance Evaluation of Multicast for Small Conferences. In B. Stiller, G. Carle, M. Karsten, and P. Reichl, editors, *Group Communications and Charges; Technology and Business Models*, volume 2816 of *Lecture Notes in Computer Science*, pages 226–233, September 2003. ISBN 3-540-20051-7.

[7] Yang hua Chu, Sanjay G. Rao, and Hui Zhang. A Case for End System Multicast. In *Proceedings of ACM SIGMETRICS*. Carnegie Mellon University, 2000.

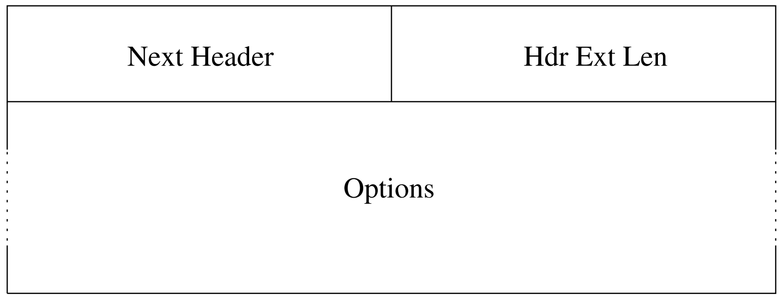


Figure 1: IPv6 Destination Options format

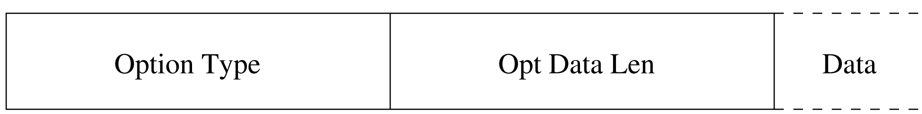


Figure 2: IPv6 Options

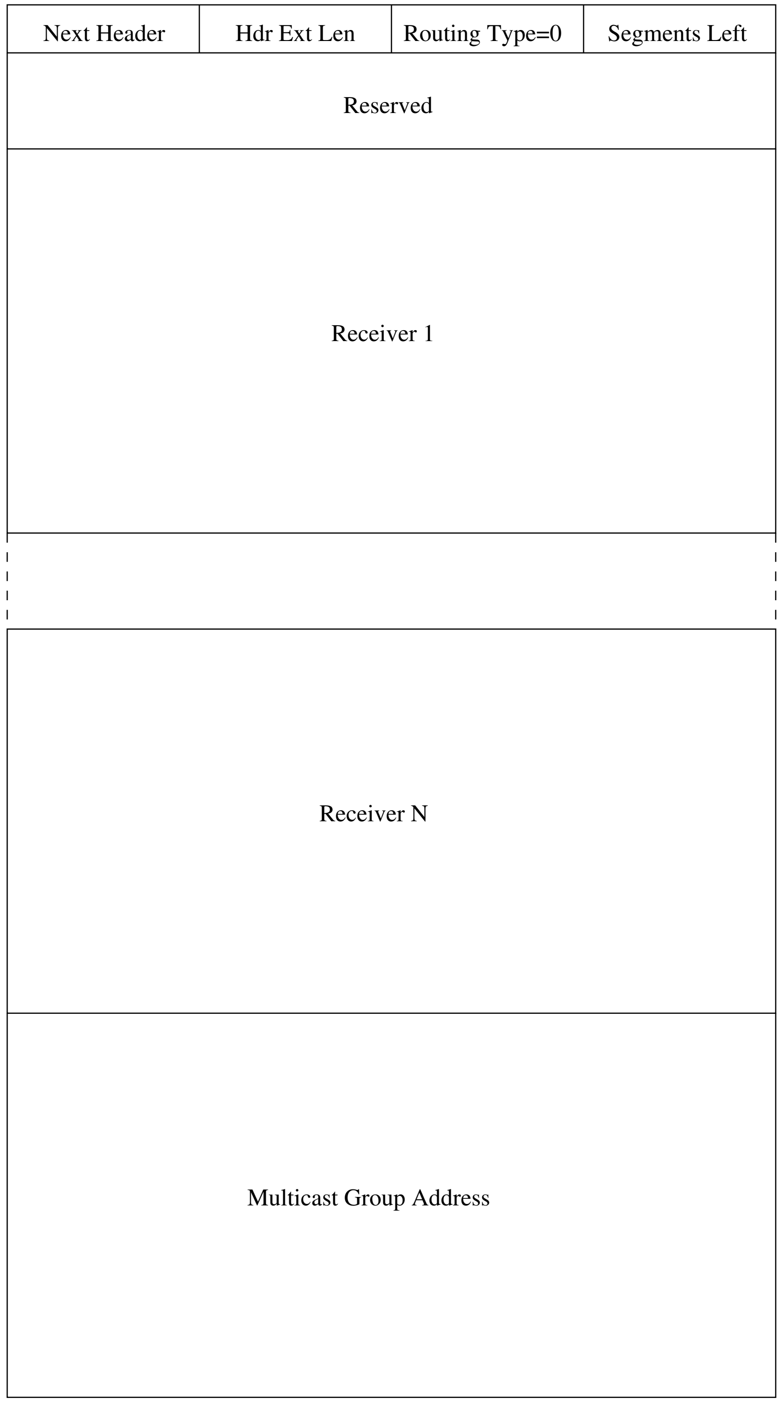


Figure 3: IPv6 routing header

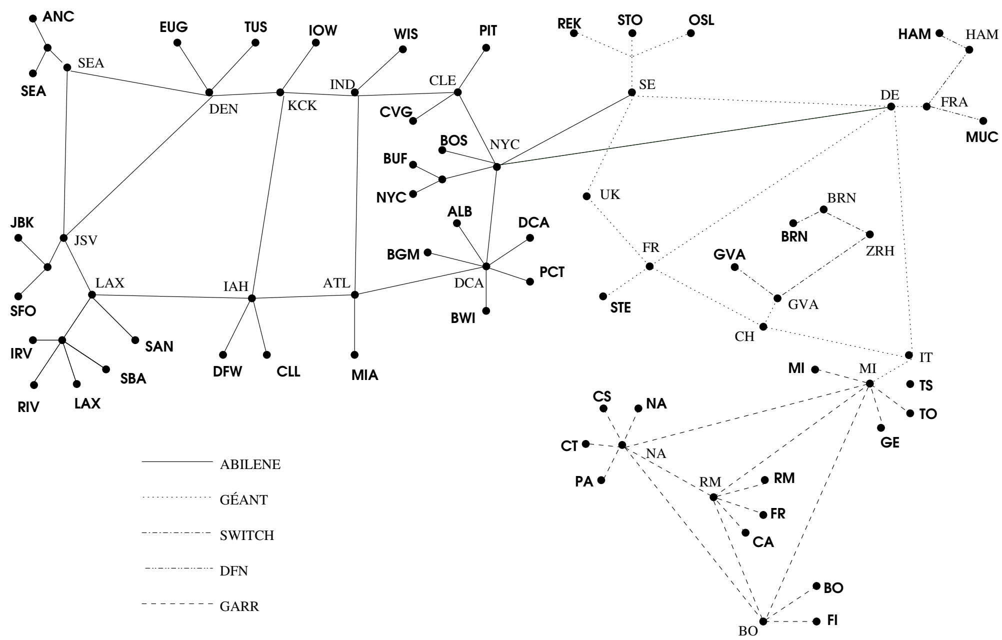


Figure 4: The ns-2 simulation topology.