Performance Evaluation of Multicast for Small Conferences

Stefan Egger and Torsten Braun

Institute of Computer Science and Applied Mathematics
University of Bern
Neubrueckstrasse 10, CH-3012 Bern, Switzerland

Abstract. 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 aims at providing more efficient support for example to audio conferences. In this work, we present a performance study of the concept, based on simulations of real-world scenarios with the ns-2 network simulation software. The results indicate that Multicast for Small Conferences has the potential of replacing IP Multicast for many delay sensitive small group applications, even with very limited support from the network infrastructure.

1 Explicit Multicast

IP Multicast does not scale well for (many) small groups such as in audio conferences or multi-player games. Multicast routing entries cannot be aggregated such as unicast routing entries since multicast address selection is arbitrary. Moreover, multicast routing entries do not only consist of destination addresss but may include source addresses. With many small group applications routing table sizes are increasing massively, which deteriorates the performance of (backbone) routers. Explicit Multicast [3] (Xcast, the successor of Small Group Multicast [4]) 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.

With the Xcast scheme, routers do not have to maintain per session state. This makes Xcast very scalable in terms of the number of sessions that can be supported. Also, no multicast addresses are used, which eliminates all problems related to multicast address allocation. Another advantage is the fact that no multicast routing protocols are required, neither intra nor inter domain. Xcast packets always take the correct path as determined by the unicast routing protocols.

2 Multicast for Small Conferences

Like XCase, the Multicast for Small Conferences (MSC) [5] 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.

In contrast to Xcast, MSC defines mechanisms to integrate native multicast and Xcast concepts [5]. These are beyond the scope of this paper. The basic MSC packet forwarding mechanism is identical to Xcast. However, instead of introducing a new protocol, MSC relies solely on the existing IPv6 protocol, in particular on the IPv6 routing header. A sender 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 (in hops). The group's multicast address should ideally be stored in the routing header as well. 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. This also means that no tunneling between MSC gateways is necessary, which simplifies a gradual deployment. 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*.

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 Simulation

In order to evaluate the performance of Multicast for Small Conferences, the protocol has been implemented in the ns-2 network simulator [6, 7]. 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) [8], the Italian Academic and Reseach Network (Garr) [9], Abilene [10], the Swiss Academic and Research Network (Switch) [11] and the German Research Network (DFN/G-WiN) [12]. In order to obtain information about MSC's sensitivity to group size and clustering, fifteen different setups (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). In this paper, we only present the results for a medium (or "weak") clustering. For example, in a group of eight end systems, there may be four pairs, each connecting to a node of the backbone network. In "strong" clustering scenarios, we would possibly have two clusters of four hosts each. Similarly, in a configuration with no clustering, each host might use a different node of a backbone network. Each setup 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 (which means that packets are forwarded between the receivers). The senders order the destination addresses by distance.
Full-scale MSC Standard MSC functionality is deployed in all end systems and backbone routers.
(E)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.

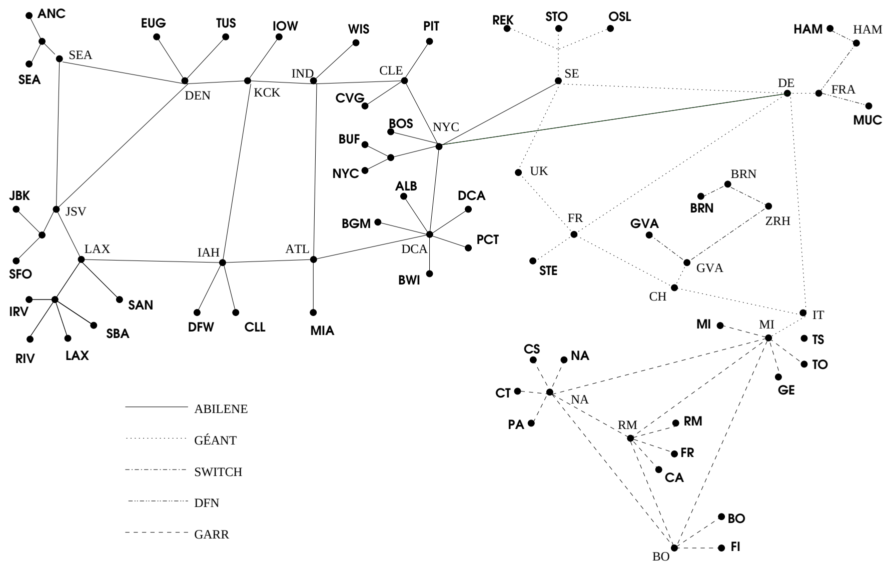

Fig. 1. The ns-2 simulation topology.

(E)MSC SIX In this scenario, only six routers (`LAX`, `KCK`, `NYC`, `DE`, `CH` and `MI`) in the topology are considered MSC capable, but these are more evenly distributed over the topology than in the previous configuration. This scenario was also simulated for both standard and enhanced MSC.

All simulations involve each end system 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. For the evaluation of the 120 simulation runs, four metrics were used. The average and maximum delays of all transmissions in a specific scenario can be used as indications of the performance of a given configuration. Usually, a delay of 150ms is the maximum that is acceptable for audio conferencing. Another important factor is *bandwidth consumption*. For this performance evaluation, the overall link usage in the scenarios has been measured. The *backbone usage* parameter uses the same data, but only data transferred on links in backbone networks is taken into account.

4 Results

4.1 Maximum Delays

There is a significant gap for maximum delays (Fig. 2) between native multicast, naive unicast, and full-scale MSC on one hand and end system MSC on the other. The other approaches deploying MSC partially are between both extremes. In particular (E)MSC at backbone interlinks suffer from higher maximum delays as

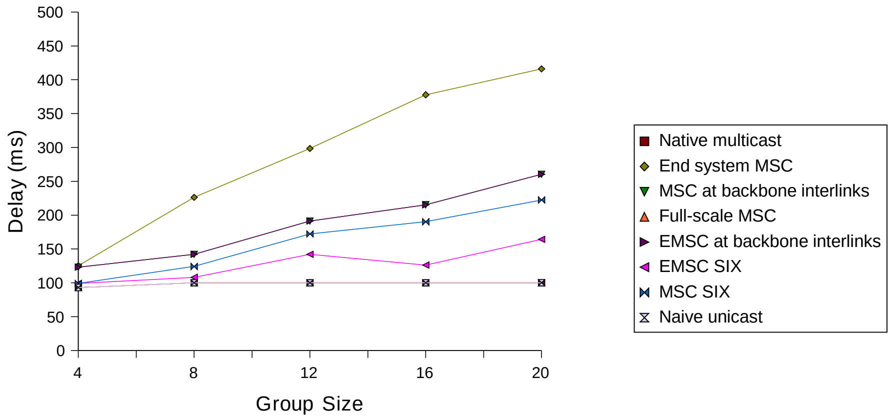

Fig. 2. Maximum Delays

group sizes increase (due to increased packet forwarding between end systems). There are a few cases where the measured maximum delay decreases as the group size increases, e.g. in the EMSC SIX configuration. The explanation of this effect is based on router distribution. When a new end system is introduced, the forwarding path of a packet from a given sender to the receivers may change, since the senders order the recipient's addresses by distance. In some configurations, the new sequence (and the resulting forwarding path) is much more efficient, for example because an MSC capable router is encountered earlier in the forwarding process. This in turn can lead to completely dfferent, potentially significantly lower end-to-end delays. This effect and a similiar occurence in average delays highlight the impact that the selection of end systems (i.e. the composition of a multicast group) can have on the performance of multicast mechanisms.

4.2 Bandwidth Consumption

Bandwidth consumption (Fig. 3) is best for native multicast, but full-scale MSC is very close. Naive unicast and end system MSC perform worst, while the other MSC approaches are again between the extremes. As with the average and maximum delay parameters, the bandwidth consumption among the eight configurations increases when group sizes increase. While the link usage of native multicast increases almost linearly, it does so exponentially for some of the other approaches, particularly naive unicast and end system MSC. This is a good indication for the scalability problems of end system-based mechanisms.

4.3 Backbone Usage

The trend of increasing differences for larger group sizes is also noticeable in the diagrams showing the backbone usage (Fig. 4). End system MSC suffers from excessive packet forwarding between end systems, which most affects the links close to the hosts. For native multicast, the explanation is different: the backbone usage is relatively high for small groups, since each destination is served using an optimal forwarding path. This drives up the number of backbone links that have to carry the packet, while the number of used access network links is low. When new receivers are added, the core of the forwarding tree does not change very much. Thus, the load on the backbone links will not increase significantly. This is further emphasized by the fact that packet size is independent from group size (unlike in MSC scenarios). However, some new access network links are added to the forwarding tree, adding bytes to the non-backbone counter and thus reducing the backbone usage.

At the same time naive unicast shows almost constant values for all group sizes. The reason is that with the naive unicast approach, each packet travels all the way from the sender to the receiver, using backbone and access networks. In the other configurations, the senders produce less packet copies and instead rely on packet forwarding between end systems, which drives up the usage of access networks.

Another interesting aspect is the performance of the (E)MSC at backbone interlinks and (E)MSC SIX configurations. While the MSC and EMSC approaches show almost identical results in terms of bandwidth consumption, they have completely different values for backbone usage. The enhanced MSC approaches consistently show higher values than the standard MSC concepts. Also, the differences increase with group sizes.

4.4 Configuration-specific analysis

The results of the native multicast configuration form the basis of the performance evaluation. In terms of delay, native multicast is insensitive to group size, because packets are always forwarded along an optimal tree. The bandwidth consumption is low, since there is no unnecessary packet duplication. Compared to the MSC approaches, the backbone usage is relatively high.

Naive unicast suffers from high link stress, because 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.

The end system MSC approach 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. The consistently low backbone usage percentage is an indication that this configuration heavily burdens the networks close to the end systems (access networks).

In full-scale MSC, packet forwarding between end systems is reduced to a minimum due to the optimal forwarding paths. This shows 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.

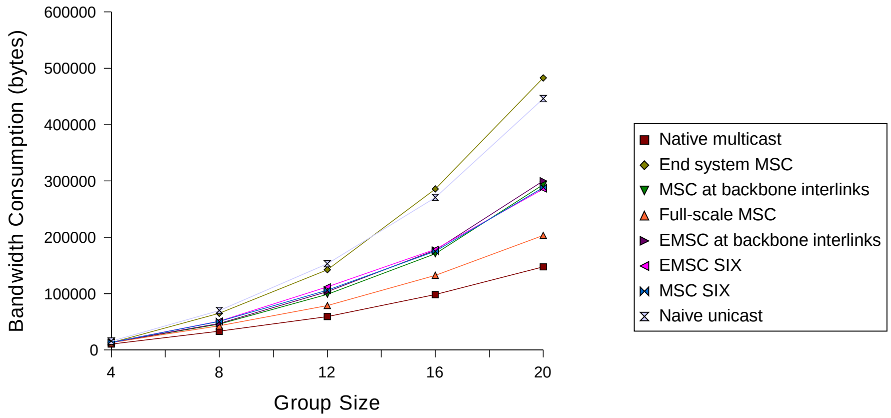

Fig. 3. Bandwidth Consumption

In terms of delays, the (E)MSC at backbone interlinks scenarios 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 severely deteriorates the performance. The major performance difference between standard and enhanced MSC in this scenario (and also in (E)MSC SIX) is the backbone usage. The packet duplication of enhanced MSC puts more strain on the backbone links, but exonerates the access networks because packet forwarding between receiving end system is minimized. Compared to the previous configuration, the (E)MSC SIX concept 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.
Overall, MSC can only achieve a performance similar to native multicast (in terms of delays) when full-scale MSC is used. With an optimized "intermediate" approach such as in the (E)MSC SIX scenarios, an average delay penalty of 30-40% (with a maximum of 90% for standard MSC and 60% of enhanced MSC) has to be accepted. However, depending on the group size, 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.

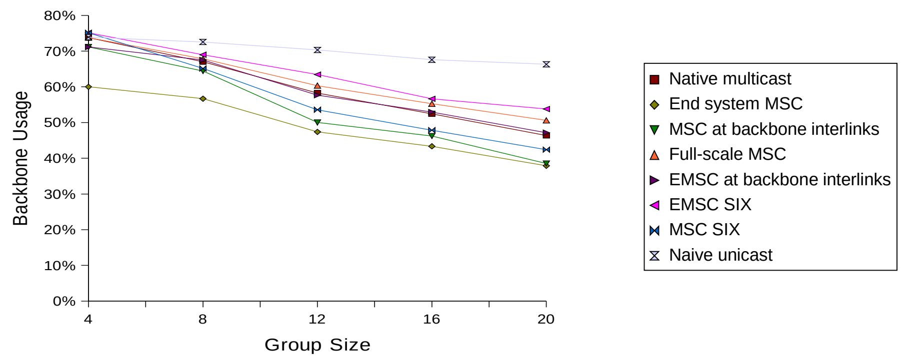

Fig. 4. Backbone Usage

5 Summary

In this paper we have presented a performance study of Multicast for Small Conferences (MSC). The results of the extensive simulations 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. The simulation results of several configurations also highlight the importance of appropriate router distribution.

References

[1] Explicit Multicast website. `http://www.watersprings.org/links/xcast/`.

[2] Myung-Ki Shin, Ki-Il Kim, Dong-Kyun Kim, and Sang-Ha Kim. Multicast Delivery Using Explicit Multicast Over IPv6 Networks. *IEEE Communications Letters*, 7(2), February 2003.

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

[4] 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.

[5] Torsten Braun. Multicast for Small Conferences. Technical Report IAM-00-008, Institute of Computer Science and Applied Mathematics, University of Berne, Switzerland, July 2000.

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

[7] Stefan Egger. Performance Simulation of Multicast for Small Conferences. Master's thesis, University of Berne, December 2002.

[8] Géant website. `http://www.dante.net/geant/`.

[9] Garr website. `http://www.garr.it`.

[10] Abilene website. `http://www.internet2.edu/abilene/`.

[11] Switch website. `http://www.switch.ch`.

[12] DFN/G-WiN website. `http://www.dfn.de`.