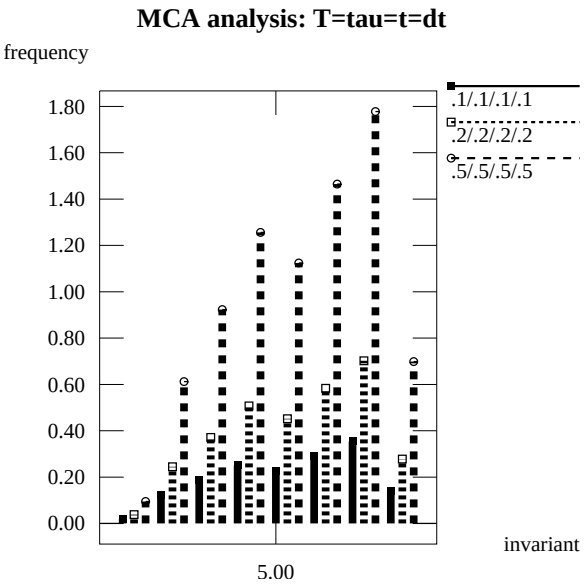
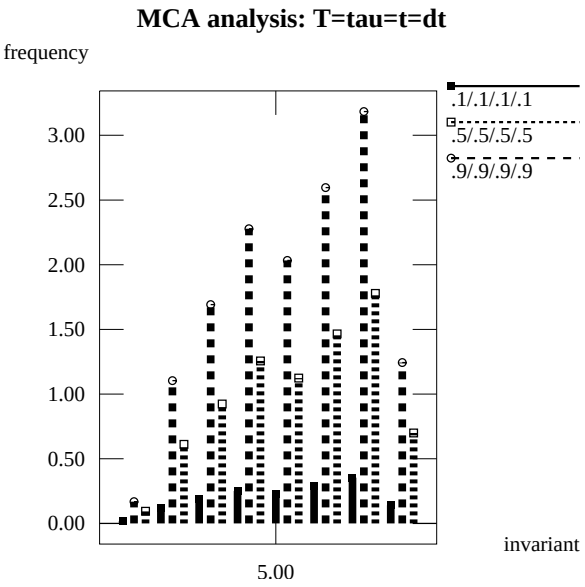
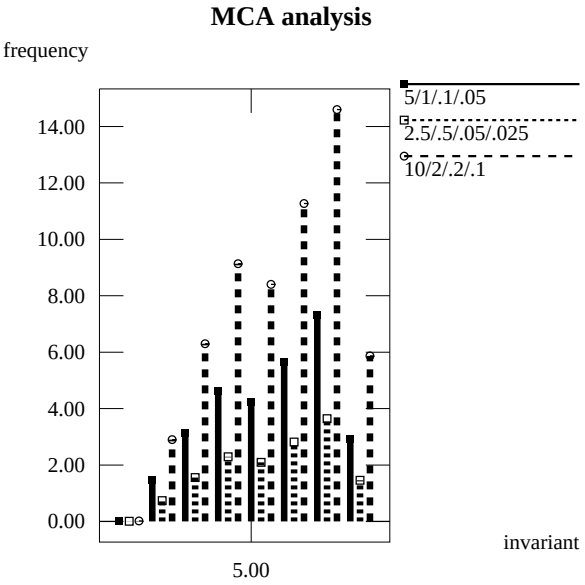
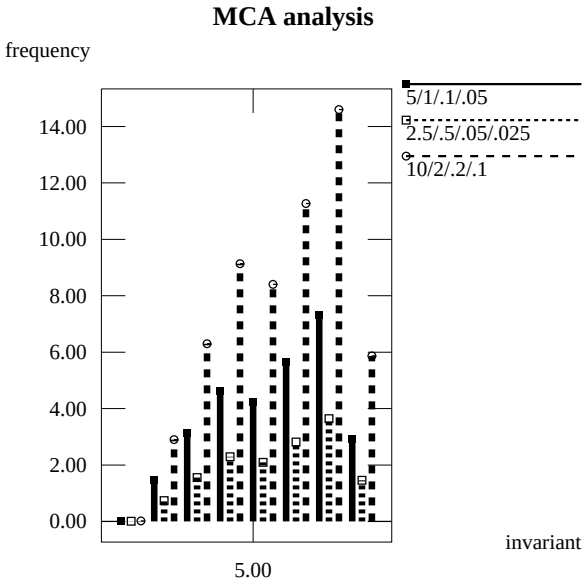
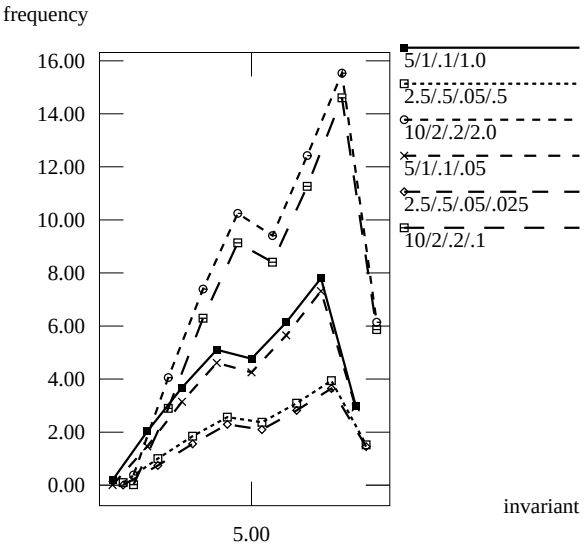
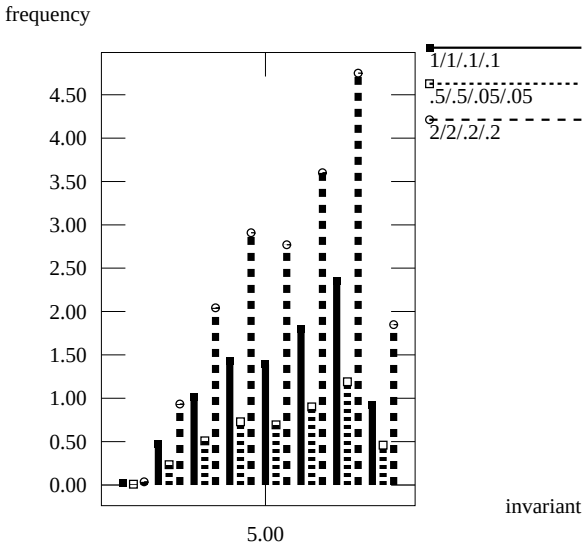

MCA analysis: T=tau=t=dt
frequency
1.80
1.60
1.40
1.20
1.00
0.80
0.60
0.40
0.20
0.00
.1/.1/.1/.1
.2/.2/.2/.2
.5/.5/.5/.5
invariant
5.00


MCA analysis: T=tau=t=dt
frequency
3.00
2.50
2.00
1.50
1.00
0.50
0.00
.1/.1/.1/.1
.5/.5/.5/.5
.9/.9/.9/.9
invariant
5.00


MCA analysis
frequency
14.00
12.00
10.00
8.00
6.00
4.00
2.00
0.00
5/1/.1/.05
2.5/.5/.05/.025
10/2/.2/.1
invariant
5.00


MCA analysis
frequency
14.00
12.00
10.00
8.00
6.00
4.00
2.00
0.00
5/1/.1/.05
2.5/.5/.05/.025
10/2/.2/.1
invariant
5.00


MCA analysis: Z var
frequency
16.00
14.00
12.00
10.00
8.00
6.00
4.00
2.00
0.00
5/1/.1/1.0
2.5/.5/.05/.5
10/2/.2/2.0
5/1/.1/.05
2.5/.5/.05/.025
10/2/.2/.1
invariant
5.00


MCA analysis: T=tau
frequency
4.50
4.00
3.50
3.00
2.50
2.00
1.50
1.00
0.50
0.00
1/1/.1/.1
.5/.5/.05/.05
2/2/.2/.2
invariant
5.00

Appendix

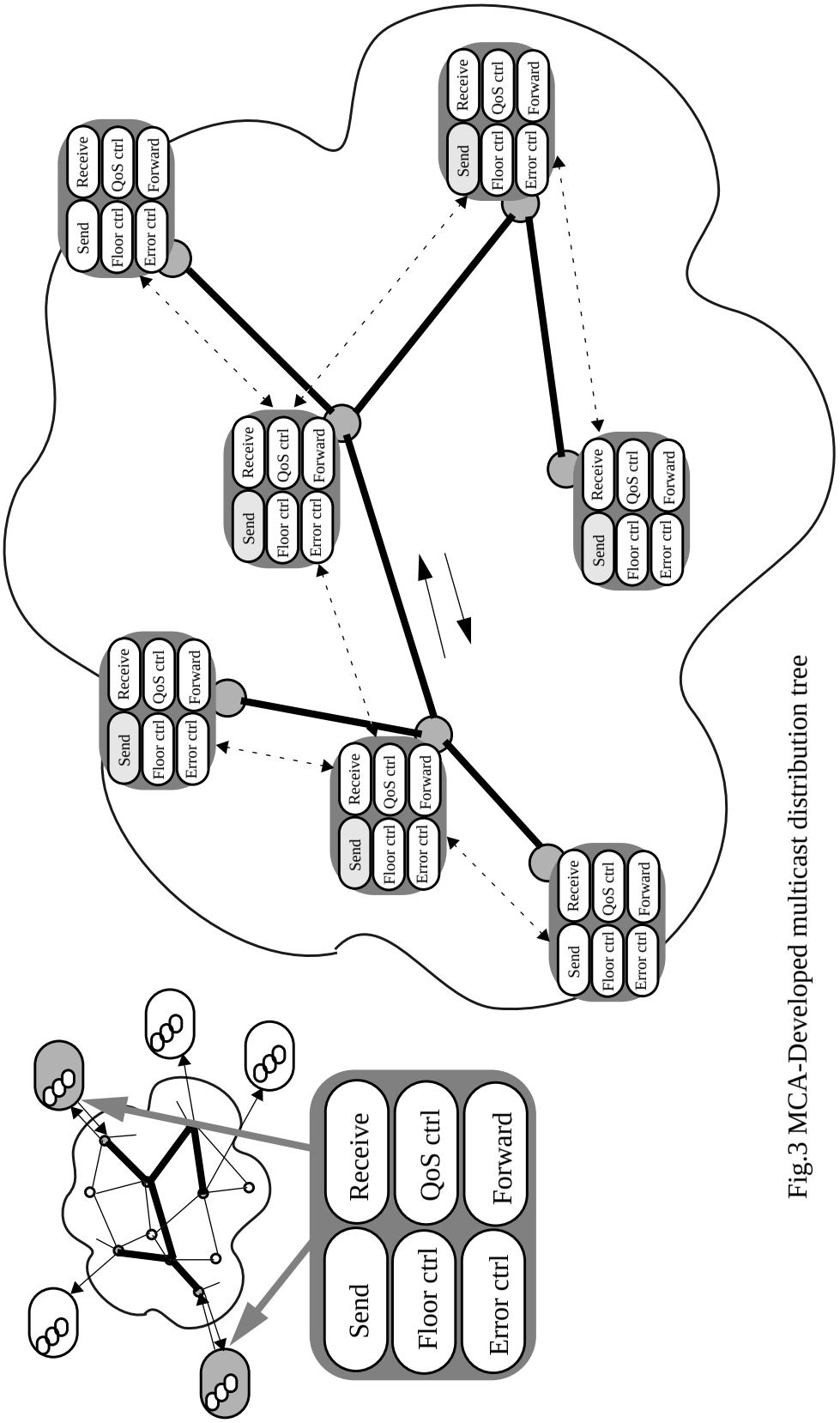


Fig.3 MCA-Developed multicast distribution tree

[11] Smirnov M.I. Architecture of a Scalable Multicast on Demand, submitted to the IDMSS'96 Workshop, Berlin, March, 1996.
[12] Deering, S., Estrin, D., Farinacci, D., Jacobson, V., Liu, Ch., Wei, L., Sharma, P., Helmy, A. "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification", *Internet draft, IETF*, Sept., 1995.
[13] Sifakis, J. "Use of Petri Nets for Performance Evaluation", *Acta Cybernetica*, Vol. 4, No 2, pp. 185 - 202, 1978.
[14] Kahn, B., Noel, R., O'Keefe, S. "XPetri - Graphical Petri net simulator for XWindows", Boston University, May, 1993.
[15] Buck, J., Goei, E., Huang, W.-J., Kamas, A., Lee, E. "PTOLEMY: Users Manual", *U.C.Berkeley, Department of EECS*, Apr., 1994.

port [14, 15] gives a rapid and flexible method for the MCA CE timing parameters estimation.

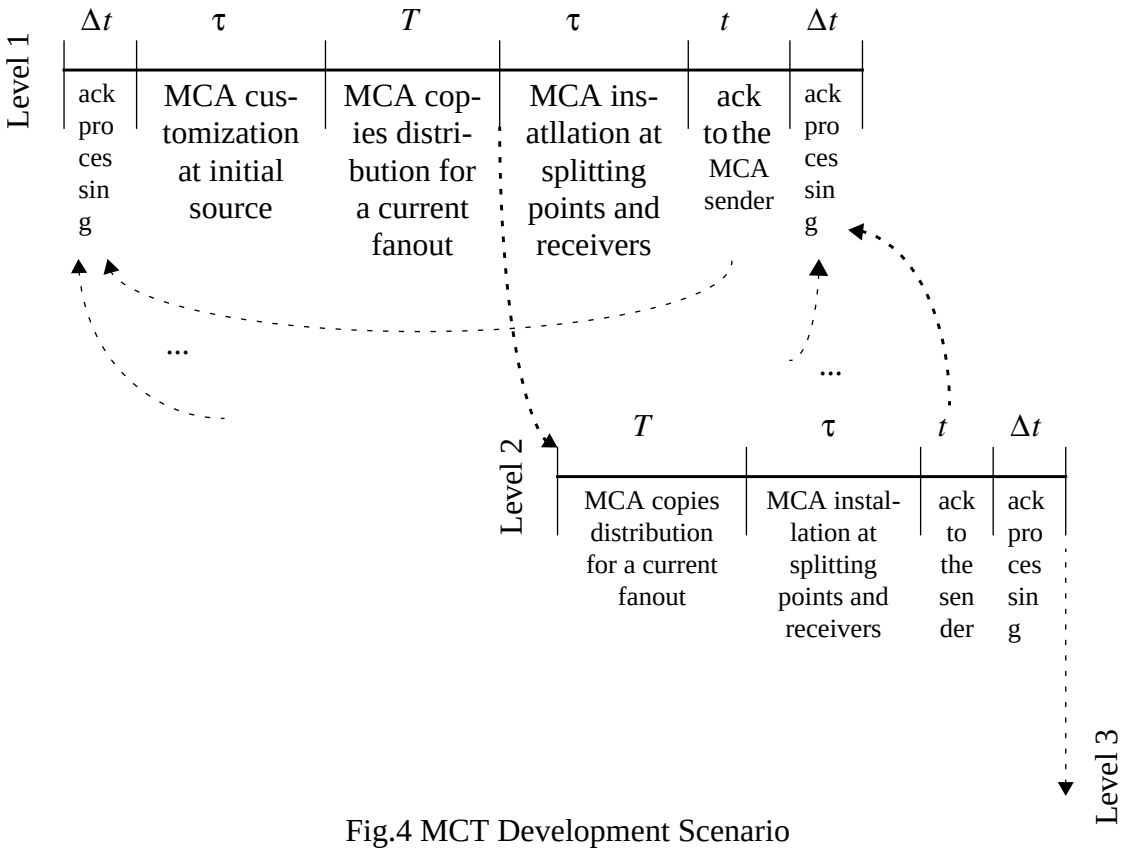


Fig.4 MCT Development Scenario

References

[1] Bangemann, M., e.a., “Europe and the global information society. Recommendations to the European Council” (available from the EuroCommission Web site)

[2] Annyal WWW user’s survey, No 4, 1995, available from http://www.cc.gatech.edu/gvn/user-surveys/survey-04-1995/#highsum.html

[3] Kumar, V., MBone Service Providers, at http://www.best.com/~prince/techinfo/contacts.html

[4] Eric-Jan,Bos. European Mbone backbone router managers, at ftp://ftp.nic.surfnet.nl/surfnet/net-management/mbone/eu-contacts.txt

[5] Thyagarajan, A., Deering, S., “Hierarchical Distance-Vector Multicast Routing for the MBone”, *SIGCOMM Symposium on Communications Architectures and Protocols,* Sept. 1995.

[6] Ballardie, T. “Minutes of the Inter-Domain Multicast Routing Working Group”, IETF *Meeting Report*,Mar., Apr., 1995.

[7] de Meer, J., Sarikaya, B., Vuong, S., Quality of service Engineering. From Conformance Testing towards Quality of Service Engineering. - to be submitted for publication in “*Computer Networks and ISDN Systems”*.

[8] Smirnov, M., Efficient Multicast Routing in High Speed Networks, - to appear in “*Computer Communications”*, Feb., 1996.

[9] Smirnov, M., Object-Oriented Framework for the Multicast Applications Integration, - Proceedings of the DOOC’95 Workshop during Object World/Frankfurt, Oct., 1995, pp. 4-13.

[10] Smirnov, M. Object-Oriented Framework for a Scalable Multicast Call Modelling, - to appear in Proceedings of the COST 237 Workshop, Denmark, Nov., 1995, 19 p.

MCA complex object but its instance customized according to a pre-negotiated set of session parameters (dedicated receivers list; media list; default QoS parameters; maximum number of active senders, etc.). The Call Establishment phase continues with invoking of MRA objects and their local negotiations with local receivers, session advertising, etc.

The Call Modification phase can start even during the Call Establishment. For example some of the dedicated receivers could become unreachable; crossing the domain boundary may cause changes in the default session parameters, one of possible active senders may ask for a floor even before the start of the initial flow, etc. We assume that this could not cause any problem because any MCA instance after the self-installation is ready to accept any of the messages which could take place within the entire multicast session. The shortcoming of this approach is that we have no opportunity to change all global parameters of the multicast session (DR list, Media List and default QoS for the reserved Media). The only possibility to change these parameters is to renegotiate with the service provider via the SP object. This object could be invoked only by any of the dedicated receivers (each of which have a potential to become a sender). The only possibility to change the DR list is to apply to the initial source.

Note, that if the number of dedicated receivers is too large for a service provider to permit them to send in parallel, then the number of active sources is to be introduced. However this number is to be somewhat less than the actual possible number of concurrent sources because we assume no information exchange between the instances of the MCA. Consequently the only way for them to know the number of actual active sources is to listen to the multicast. However, due to a delay introduced by the network several sources could start sending simultaneously based on their local permission. To prevent this situation we assume simply that the MCA instance nearest to the last source has to stop sending when recognizes that actual number of active sources is greater than default maximum.

6. MCA Performance Aspects

As long as the proposed agent technology for the MCT development is totally distributed we have to pay special attention to some potentially dangerous events during the development. Main features which will be addressed here are: scalability; manageability and global state. The global state problem exists because no one MCA has a complete knowledge of the global state, however MCA at all splitting points notify all receivers joined within their fanout in a bottom-up manner. Finally, the decision making on total MCT development quality is made by initial MCA. All reasonings will be made regardless the network structure; we require for the multicast agents only the knowledge of initial MCT. First, we assume that exchange of information between MCA entities exists only within each fanout. The MCT development scenario is presented in Fig.4 together with the explanation of major timing parameters of this scenario.

To check the performance of the MCA Call establishment protocol the Timed Petri Net model was studied with the use of the invariant method, proposed by J.Sifakis [13]. The analysis method assumes that the system under modelling is to be decomposed into the set of invariant subsystems (comprising the monolithic model however). We have found 8 invariant subsystems in the MCA CE model and their resulting behaviour (measured in a form of frequencies of corresponding Petri net transitions) is presented in diagrams 1 -6 (Appendix).

Note, that these estimated frequencies could not be greater than particular value found analytically out of the Petri net model (here 3.0), which fact together with the available tools sup-

functionalities are distributed over the multicast tree and managed by the contract and the multicast flow (received by routers at the MCT splitting points). Scalability depends on the number of splitting points in the multicast distribution tree; the source deals only within one level of the tree.

5. The MCA Components.

The Multicast Call establishment scenario development is not the purpose of this paper, however we would like to show what components are essential with the use of the MCT Development concept and the Multicast Agents object presentation. Actually we consider the MCT Development scenario as a framework for the Call Establishment Model design.

The *Src* method of the complex MCA object is an object with the following methods:

- attach/add/detach/remove media;
- define medium-to-medium relations (for multi-media multicasting);
- include/exclude applications;
- floor control;
- give current multicast state (optional).

The *SP* method of the complex MCA object is an object with the following methods (parameters):

- enable/disable multicast session (contract);
- quality of service parameters (DR list; Media list).

The *MRA* method of the complex MCA object is an object with the following methods:

- filtering (Media list; QoS requirements list);
- include/exclude applications;
- attach/add/detach/remove media;
- patching (Medium, badly received packets);
- advertising (optional);
- floor control;
- participation control;
- give current multicast state (optional).

The *MTA* method of the complex MCA object is an object with the following methods:

- clone (MCA);
- send copy (MCA);
- invoke (MCA_method);
- prune (MCA_instance);
- split, i.e send copy (multicast packet).

We assume that when the SP method is activated it starts negotiation with the service provider for a multicast session with required dedicated receivers set and required QoS parameters for a certain media involved. This contract is assumed to be assured by the service provider all over the MCT. If different receiver hosts will require changes to the contract defaults then they have to negotiate them with their MRA.

Actually the MTA entity realizes a protocol for distributed multicast management; this particular method has to realize essential multicast functionality to achieve scalability of the sparse multicast groups.

This set is called a framework because it is not elaborated yet in the sense of object messages, also some methods are optional and others could arise during the multicast implementation. These building blocks can support the whole lifecycle of the multicast session, including Call Establishment and Call Modification.

The Call Establishment phase needs to invoke the Src and SP objects to establish primary contract with the service provider and MTA object to clone, transmit and self-install the MCA complex object within the nodes of the MCT. Note that we suppose to clone not an abstract

assume manually configured MCA, called initiator, which is capable to start the process.

The multicast agent was proposed to enable scalable and reliable multicast of, mainly, multimedia flows across the internetwork. Let us turn now to the main task of the MCA. The initiator sends the multicast flow to all its children collaborating with the regular routing protocol implemented within a particular internetwork domain. This collaboration has a server-client style. The MCA gives the multicast flow to the routing process together with the set of addresses (where the MCA children already reside), and the routing process sends the flow out to these addresses. Any MCA child, having received the flow makes this flow its own property and acts further as if it is initiator. Obviously, there are some technical details. First, the multicast flow could be seen as the multicast object. The size of this object (measured in disk space or transmission delay/bandwidth) is, however, so large, that the internetwork has to partition this object into packets. Also, the multicast child may itself become a sender of the multicast flow, etc.

4. MCA Technology Motivations

Topologically, any multicast tree could be seen as a composition of several serial interconnections (unicasts) and fanouts of nodes. Therefore, any communication supporting and providing multicast could be composed out of several unicasts and simple multicasts. The multicast could be called simple when the corresponding multicast tree is a fanout, for that reason a new term could be introduced - *a fancast,* i.e. inevitable multicast which could not be reduced. From these reasonings comes the idea of a scalable multicast. This decomposition is complete, i.e. no more fanouts or unicasts could be found. However, this is not a single possible decomposition. Different structures can arise during the multicast call establishment. Out of this structural motivation it is evident that due to inevitable overlapping of unicasts and fanouts the functionalities needed to support unicasts and simple multicasts are to be placed within their intersection points, while these points are actually the *splitting points* of the multicast flow. A splitting point could be defined as a network router where packets of the multicast flow have to be reproduced in at least 1 copy. All over the paper we assume that receiving hosts connected to routers are not shown in the figures.

The desired scalability feature could be achieved via limited visibility out of each splitting point: we say that from each splitting point only a set of nearest reachable splitting points and receivers could be observed. This improves scalability and manageability of these partial multicasts. However, the need to provide the self-distribution of the multicast functionality over the multicast tree is the cost for these improvements.

In this paper we are especially aware of the case of the so called *sparse group multicast* [12], when the number of receivers is very large comparatively to the number of possible senders; receivers could also be distributed across several domains. Current multicast systems are composed with applications and network support, sender and receiver applications communicating via the network. The network support consist of special address scheme, host extensions and routing support. Many authors argue in favour of more and more support for future multicast multimedia applications form the network. However, networks become more and more high speed, i.e. tend to provide less functionality during the data transfer. Hence, one of the requirements for a successful multicast implementation - to keep it simple for the network, e.g. following the Internet best-effort philosophy. The second reasoning comes from the fact that one can not change the placement of a multicast flow (source) for obvious reasons (human participants, cameras, data bases, etc.), however the *starting point* for the multicast tree could be placed *anywhere* within the network, provided that this anywhere place has a good connection to the source. Out of these two considerations comes the proposed scenario for a scalable multicast call establishment, which was named the *multicast tree development*.

Figure 3 shows the difference between conventional multicast tree with 2 senders (smaller picture) and the same tree with the MCA installed at splitting points. In the MCA case all control

trary, the QoS Engineering approach, proposed in [7] for the emerging collaborative technologies, emphasizes the need of both pre-implementation and the run-time QoS testing.
The rest of the paper describes one possible solution to achieve the desired features for the multicasting with the proposed multicast agents (MCA), being the internetwork-mobile software objects capable to bring needed functionality to the most valid locations. These locations aims to guarantee the QoS level negotiated with the service provider, to provide cost-effective reliability of the multicast flow delivery as well as flexible floor-control.

3. The MCA protocol

The MCA protocol does not exist. Moreover, the MCA does not exist too, with the exception of paper versions [8 - 11]. MCA means “Multicast Agent”; it is a networked software object aimed to enable scalable and reliable multicast over the internetwork. This agent, according to the author’s proposal, should be able to clone itself (i.e. produce its own copy) when receiving such a command. Then the mother copy is to be able to distribute its child[ren] (most probably, in a form of a data stream) to several destinations, when receiving such a command. At the “receiver” (better to say, destination) side this agent should extract (and install) itself from the data stream, when receiving such a command or immediately if it was configured to do so. And,finally, this agent is a complex object, i.e. it is supposed to be filled with other useful objects fulfilling the common functionality. The MCA should be able to activate any object which it encapsulates and/or be able to invoke any object’s method at its common interface.

The set of objects inside the MCA can communicate with one another via the MCA message system. Messages are interpreted by the MCA objects as commands or as commands with data (parameters). Actually, this message system is a broadcast. When an object receives a message it filters it: if the message received has this object as its destination identifier then this object accepts it, otherwise discards the message. The MCA entities can communicate with one another as well, however, with some limitations. First, the original MCA and the copy MCA may communicate only after the copy entity installation has been completed in the remote site. That is, when the parent MCA clones, its child can not speak while being at the same network location. We say that this MCA is in the *embryonic* state. Second, the MCA is a framework for encapsulated objects, and it is assumed that frameworks can not communicate other than by populating objects’ talk. That means, remote objects belonging to different MCA entities can exchange messages, providing for additional MCA identifier within the message format. Third, only those objects can exchange messages which reside within the MCA entities being in the “mother - child” relation. That is, outside MCA the message system is a limited multicast. Fourth, inter-MCA communication of objects is facilitated by a special transportation object within the MCA. Each MCA may have at maximum two transportation objects: one for upstream communication and one for downstream communication. That is, any object willing to communicate with a remote object (upstream or downstream) has to broadcast its message MCA-wide (providing for an up/down bit within the message format), the transportation object will pick it up and deliver to the needed (upstream or downstream) MCA entity. At the remote MCA entity, the transportation object modifies the message header to conform the MCA message system and broadcasts the message MCA-wide. We assume total reliability of the MCA internal message system.

Consequently, the parent MCA broadcasts its messages to all its children MCA, and every child MCA sends its messages (if any) to the parent MCA as a regular unicast. Hence, the parent MCA will receive messages from all its direct children. Almost every MCA could be seen as a child of some parent MCA. However, two exceptions exist. First, we assume that sleeping embryonic MCA could exist. This sleeping agents have no knowledge of their parents. However, they could be awoken by a special “get up” message. This message is a command with data, which provides the parent-child relation knowledge to the sleeping MCA. Second, we

that of the whole Internet).

Table 1:

Class of the MBone event (Jan-Nov.15,1995)	MBone occupation [hours]	Number of sessions	Average hours per session
On-line video, radio and art programs, press-conferences and presentations	95	19	5
Conferences, workshops, etc.	969	56	17.3
Meetings	148	15	9.9
Lectures, distance learning, teleteaching	232	64	3.64
Other (private conferences, demos, etc.)	39	11	3.55

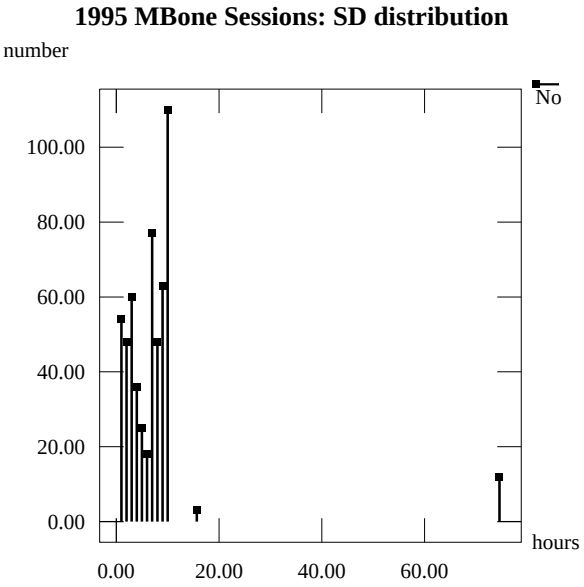


Fig. 1

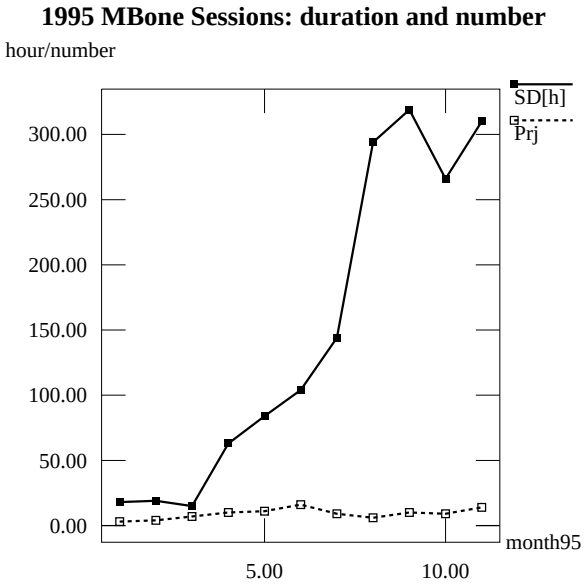


Fig. 2

As it could be seen from this Table, the most often events were educational ones, however the most time-consuming were multicast transmissions from various conferences and workshops. It is extremely important to mention rather high number and duration of meetings proving somehow even the current MBone applicability for collaboration, i.e. for active participation.
Unfortunately, the current MBone, growing so rapidly, is near to its essential limit: this artificial virtual structure over the Internet connectivity cannot scale well enough to the demand from both the users and the service providers.
The Internet Engineering Task Force, and, in particular, its Inter Domain Multicast Routing working group are currently proposing a number of hierarchical solutions to achieve the needed scalability for the multicasting [5, 6]. However, the best-effort philosophy remains untouched due to the inevitable evolutionary deployment of protocols and structures. On con-

The paper is organized as follows. Chapter 2 aims to prove that multicasting is mature enough to influence the telecommunications and applications market. The MCA protocol is introduced informally in chapter 3, technology motivations could be found in chapter 4, while chapter 5 discusses the agent components in more detail. The paper is concluded with the performance aspects of the proposed approach.

2. MBone Becomes Mature

The current Internet could be thought of as the largest ever been pool of information resources with different access facilities from the user working place. These differences could be found in many aspects. To mention some of them, let us refer to the end-user working place hardware and software features, network connections, including, again, hardware and software solutions, administrative regulations, financial and political constraints for the user access, security issues and so on. However, despite all these differences, there exists the World Wide Web technology, fast growing, yet a very new one, which already became "a default interface to the Internet"[2], with the number of Web sites doubling every 53 days. The wide acceptance of the networked hypetext in academic and engineering communities, in educational, governmental, commercial, etc., spheres has been proven by a tremendous growth of both consuming and producing of the HTML documents. Note, that producing the Web-readable documents is additional labour for the user. Thus, the HTML really becomes a new sort of the *human collaboration* supporting media. The well known Nolan's "stages of growth" model is highly applicable to the WWW. The "contagion" stage for the Web information technology is far away, and the "mature" stage is very near. The latter is proven by the fact that Sun's Hot Java is based on the existing Web browsers, adding a new dimension to it; Video on Demand is becoming available via the Web, etc. The "Web industry" is profitable; for example, according to the Internet Business Centre' monthly reports the number of sales generated by the Web sites is constantly growing.

Another, yet experimental, technology - multimedia multicasting - seems to be at the very end of the contagion stage. The number of hypertext reference links to the Internet's Multicast Backbone (MBone) is growing daily. Thus, we may say, the "MBone industry" is coming into reality too. The number of the MBone service providers is about 90 in the USA [3] and more than 40 in Europe [4], the number of the commercial vendors for the MBone industry is significant for the "pure experimental initiative": 6 workstation vendors, 7 IP router vendors, unknown number of the multicast software vendors (the number of only video conferencing products is 55 and about 20 packages of desktop application software for the multicasting are available through the Web [3]). Figure 1 shows the histogram of the MBone session duration (SD) for the first 10.5 months of 1995 in hours, and Figure 2 shows the overall MBone occupation in hours together with the number of the projects. Despite the constant growth of the MBone usage shown in the diagrams, these data refer only to planned and scheduled events. All data were derived from the "MBone Global Agenda" Web site.

Let us now have a look at the structure of the MBone events in 1995. Table 1 shows some classification of those multicast sessions which took place in the MBone according to the Global Agenda (Note, however, that the MBone is still considered as experimental initiative, the bandwidth is limited in total to 512 Kb/s, protecting the Internet from flooding; not all of the Internet routers support the MBone tunnelling, and, generally the MBone structure differs from

The Multicast Agent Technology.

Motivation. Introduction.Analysis

Smirnov M.I.
GMD FOKUS
Hardenbergplatz 2, Berlin 10623 Germany
smirnow@fokus.gmd.de

Abstract

The Internet Multicasting Backbone (MBone) proves its maturity as an essential information technology enabling rapid growth of collaborative applications and reshaping the world of multimedia multiuser communications. However, the existing MBone lacks needed scalability to meet the growing demand. The paper proposes a new approach based on the multicast agent technology bringing needed multicast enabling functionalities (error correction, floor control, Quality of service negotiations, etc.) to the most valid locations within the multicast distribution tree. The appropriate call establishment scenario is proposed and analysed with the use of modelling.

1. Introduction

The Information Society most probably will never come into reality in its straightforward meaning: is anybody calling the current Society a Telephone Society or Television Society? However, telephone and television, being nowadays *essential social* communication and information technologies, influence the Society greatly and reshape its behaviour in almost all spheres. Complementary to this, when smart computer communications will also become an *essential social* technology, the Information Society will promote some, yet unknown, new catchword.

We should speak about smart computer communications first as the basic enabling technology facilitating desired changes in other spheres. Look for example into the ten applications to launch the information society in Europe [1]: Teleworking, Distance learning, A network for universities and research centres, Telematic services for SMEs, Road traffic management, Air traffic control, Healthcare networks, Electronic tendering, Trans-european public administration network, City information highways. All these applications are networked ones and their progressive growth depends not only on the communication capabilities enhancement, but on the middleware support and interoperability as well.

Current progress in the multimedia multipoint-to-multipoint collaboration support from the internetwork could be seen both as the challenge for the application designers and the testbed for the design alternatives. New applications supporting collaboration are on their way to customers in various domains such as distributed production management, scientific collaboration, medicine, distributed computer aided design, and so forth. However, the complete understanding of the multicasting technology especially in the case of multimedia collaboration still is not achieved. The Internet best-effort philosophy does not permit the Quality of Service assurance - the basic requirement for the technology to become *essential*, i.e. to be accepted as such in the Society's consciousness.

A new dimension of the Quality of Service may arise from the recent IT innovations, such as multimedia collaboration, hypemedia and agent technology. A user satisfaction could be measured in the nearest future in terms rather different from those we are currently accustomed to. Due to this reasons it seems reasonable to propose and analyse rather new approaches supporting scalable, reliable and QoS guaranteed multiuser multimedia collaboration.