Contract Number: **IST-2000-26417**
Project Title: **GN1 (GÉANT)**

Deliverable D9.2

Guidelines for delivering usable multicast to the end user

Deliverable Type:	PU-Public
Contractual Date:	30 June 2001
Actual Date:	22 August 2001
Work Package:	WP8
Nature of Deliverable:	RE - Report

Authors: Ladislav Lhotka/CESNET
Robert Stoy/DFN

ABSTRACT

The deployment of a reliable IP multicast service in GÉANT must be based on a sound design of multicast routing in the European backbone and within individual NRENs. However, problems with end-to-end multicast connectivity are often due to improper hardware and software configuration of end stations, switches or routers in Local Area Networks. Moreover, it is important to provide feedback channels for end users so that they can check the actual multicast connectivity on a pan-European scale and report any problems in advance before using multicast for important data transfers, videoconferences etc. This document gives guidelines to end users and LAN administrators on how to configure PC hardware and software and LAN switches and how to verify from real network traffic that multicast is working properly. Further, the previous experience with two multicast monitoring environments is described and recommendations are given for the setup of a hierarchical monitoring system covering both GÉANT and NRENs.

Keywords:

IP multicast, hardware and software configuration, multicast monitoring, Multicast Reachability Monitor, Multicast Beacon.

CONTENTS

EXECUTIVE SUMMARY

IP multicast is still a relatively new communication technology and as such it suffers from poor or limited feedback between users and providers of the technology. Backbone IP networks are in many cases ready to provide an inter-domain multicast service but since multicast functionality is not strongly demanded by large masses of users, both router vendors and network operation teams usually do not assign top-level priority to solving multicast issues. Users, on the other hand, generally consider IP multicast to be inherently unstable and unreliable and often seek alternative communication methods or workarounds like reflectors or tunnels. For the over-provisioned gigabit networks and multicast applications that are now prevalent – mainly the Mbone videoconferencing tools – users view the use of these workarounds more effective than attempting to solve multicast problems in the first place. Recent experience from TEN-155 and some NRENs shows that general multicast connectivity may be missing in some locations for extended periods of time without any serious complaints from the users.

The aim is to develop IP multicast into a reliable service in GÉANT that will be demanded by users and well supported by the network operation centers. To this end, the aim of this work is to assist end users by:

- giving them information necessary for a proper operation of IP multicast in their local environment: personal computers and local area networks;
- Developing a real-time monitoring infrastructure and present the output to end users so that the information on intra-domain and inter-domain multicast connectivity and transmission quality is continuously available.

The aim of this document is to provide basic guidelines in both areas. First, we give users hints on setting up the hardware and software in the most frequent case of PC architecture and Ethernet network. On the hardware part, the most important task of selecting an Ethernet card is explained in some detail. As far as software is concerned, most modern operating systems have multicast support built-in and multicast configuration is performed automatically as a part of general network setup. A web site is also under development where specific information about IP multicast in the popular operating systems will be available. Further, we also explain how to use traffic analysis tools for verifying correct multicast operation in local area networks.

In the second part of this document we describe our experience with two multicast monitoring tools – *Multicast Reachability Monitor (MRM)* and *Multicast Beacon*. Both are based on the principle of agents – active probes in various places of the network, which generate multicast traffic that is received by some or all of other agents. From this data exchange the following characteristics may be determined: packet loss, duplicates and reordering and, if the agents are precisely synchronised, also one-way transmission delay and jitter. In the first stage of GÉANT deployment we recommend to use the Multicast Beacon, mainly because MRM is rather complicated and its development in IETF has recently be stopped. The Multicast Beacon also fulfils better the important requirement of providing an easy-to-use web interface accessible to all users.

Finally, we suggest that Multicast Beacon agents be installed in all GÉANT access points from Day 1 and the results used for routine multicast monitoring. Moreover, all participating NRENs should set up a similar but independent Multicast Beacon infrastructure covering their national backbone. Users could then easily determine whether an observed multicast problem resides in their NREN or the GÉANT backbone.

1. INTRODUCTION

In most research and education networks, multicast is considered a standard part of IP service. So far, the most popular application has been the Mbone videoconferencing suite [Mbone] but there are certainly more uses for IP multicast including multimedia streaming, bulk data distribution, distributed simulation and games. Yet the usage of multicast (at least in the native form) is far from widespread. We see here the problem of limited feedback between users and providers of multicast services:

- There are few attractive and production grade applications.
- Development of multicast-related protocols and standards in IETF is far from complete.
- Manufacturers of routers and network interface cards mostly don't assign high priority to multicast features.

Commercial ISPs are rather reluctant to support IP multicast in their networks.

Our experience from the TEN-155 network and most NRENs shows that, unlike the unicast case, the end users and/or administrators of peripheral networks often do not provide timely feedback to the backbone NOCs in the case of problems with multicast connectivity or transmission quality. In fact, IP multicast is generally perceived as being inherently unreliable and some user communities thus turn to other data distribution schemes or deploy workarounds – tunnels, reflectors etc.

The process of developing IP multicast into a reliable service must certainly be based on the sound design, configuration and management of multicast routing and related protocols in the backbone networks – both GÉANT and NRENs. This task is up to network specialists that have the necessary knowledge and tools to accomplish it. Unfortunately, having IP multicast operational at the backbone level does not guarantee a usable multicast service for end users. Too often the multicast connectivity is broken in peripheral parts of the network or even in the terminal workstation.

This document is thus intended mainly for end users of IP multicast and administrators of local area networks. It concentrates on the multicast-related issues that may be effectively addressed by an average user, identifies the common functional and configuration problems in the hardware and software of personal computers and local area networks and provides a selection of software tools that may be used for diagnosing the situation. Finally, a hierarchical monitoring infrastructure is described that will give the end users an overview of large-scale multicast connectivity and transmission quality and, in the case of problems, help them in locating their cause.

We intentionally do not cover the area of DVMRP tunnels [RFC1075], reflectors and similar workarounds for native multicast. Of course, in some situations, their use may be legitimate or even necessary, but we would like to encourage users to look for a direct remedy against outstanding multicast problems first. Also, we do not delve into the complicated issues of multicast routing because in our opinion they are not of direct interest to end users.

2. IP MULTICAST IN THE END-USER ENVIRONMENT

End users typically do not have access to router configurations and network backbones and their view of IP multicast is thus limited to their personal computer or workstation and the local area network it is connected to. In this section we will give some guidelines concerning the hardware and software configuration and also interpretation of multicast related information that can be obtained from LAN traffic analysis.

2.1 PC Configuration

Unlike traditional Unix workstations, the ubiquitous Intel/PC platform is characterised by a considerable variety of hardware options and operating systems, which are both moreover moving targets. Therefore, we can give only general guidelines and recommendations.

2.1.1 Network Interface Cards

This discussion will focus on Ethernet technology since the alternatives (Token Ring, FDDI, ATM, etc.) are quite rare now.

As far as multicast is concerned, the main role of an Ethernet NIC is to set hardware filters on destination MAC addresses so that all undesired multicast packets are not sent up the network stack Ethernet adapters differ considerably in their support of hardware filtering. The importance of this function depends primarily on the LAN configuration: in completely switched Ethernet networks where the switches use some kind of multicast packet filtering – IGMP snooping or CGMP (see below), the Ethernet NIC need not do any filtering at all. On the other hand, for shared network segments with heavy multicast traffic (or for PCs operating as multicast routers) the NIC filtering capabilities may certainly save a lot of CPU capacity. Another point is that a NIC with a clean and efficient hardware design allows simpler software drivers that may be better debugged and optimised for throughput.

The multicast packet filtering in a NIC works as follows: If an application intends to receive traffic from a multicast group, it asks the IP layer (e.g. through a socket API function call) which, apart from other actions, maps the requested IP multicast address to a multicast MAC address. This is done by concatenating the 25 bit prefix `01:00:5e:0` with the 23 rightmost bits of the IP multicast address [RFC1700]. The Ethernet driver is then asked, in one way or another, to receive frames with such an address. From this procedure it is immediately clear that the mapping of IP to MAC multicast address is not one-to-one (32 IP addresses map to the same MAC address). Therefore the IP layer must be prepared to filter out multicast packets, which are not desired but pass through the link layer filter. In reality, anyway, the filtering done by most Ethernet adapters is much coarser than this, as will be explained below.

Most of the common Ethernet chipsets just sort the incoming Ethernet frames into a limited number of “buckets”, typically 64, according to the hash value that is obtained by computing the checksum of the first six bytes in the frame, i.e., the destination MAC address. Two of these buckets correspond to the NIC’s own MAC address and broadcast MAC address `ff:ff:ff:ff:ff:ff`, respectively – frames falling into them are always received. The rest of the buckets may be used for multicast frames.

More sophisticated chipsets are able, besides using the hashing procedure, to do an exact match for a number of MAC addresses, typically 16, again including its own and broadcast MAC addresses. And some of the recent chipsets with Content-Addressable Memory (CAM) are even able to match an unlimited number of MAC addresses exactly.

Unfortunately, there are also Ethernet adapters (most notably the 3Com EtherLink III series), which don’t support hardware filtering or have bugs in it. In that case it is always possible to put the Ethernet NIC in a special mode in which it receives all multicast addresses (ALLMULTI mode). All the filtering must then be done in software.

Table 1 below lists seven common Fast and Gigabit Ethernet chipsets together with the numbers of address slots for exact matching and buckets for hashing.

Table 1. Multicast properties of common Ethernet chipsets

Vendor	Chipset	Exact	Buckets	Remark
3Com	9xx	N/A	N/A	
DEC/Intel	2104x, 2114x	16	512	
Intel	Etherexpress PRO	64	N/A	(1)
Packet Engines	Yellowfin	N/A	64	
SMC	EPIC/100	N/A	8	(2)
SysKonnect	SK-98xx	16	64	
Texas Instruments	ThunderLAN 10/100	4	64	

Notes on Table 1:

1. Updates of the MAC address list are slow and can lead to congestion and packet losses.
2. The MAC address filtering is not usable due to firmware bugs and so ALLMULTI mode must be used.

The hardware filtering capability is not the only criterion for selecting an Ethernet adapter, especially if we are building a multicast router or multimedia streaming server. The other important functions are:

- calculation of the IP header checksum in hardware
- *scatter/gather*, which means that the NIC can read packet data simultaneously from several locations in memory (for example, the packet payload and IP header may be fetched from different places).

Many details on the operation of various Ethernet chipsets and other useful information can be found on the Gigabit and 100 Mbps Ethernet Technology page [SCYLD].

2.1.2 Operating systems

Most modern operating systems have built-in support for receiving and sending multicast packets. No additional user-space programs are normally required, unless we want the PC or workstation to act as a multicast router.

As part of the project, we started to collect system specific information about basic multicast configuration and operation. So far only Linux and some aspects of Solaris have been addressed but other operating systems will follow, including various versions of Microsoft Windows, NetBSD, FreeBSD, MacOS, IRIX and AIX. This information base is publicly accessible at the web site [Mcast-user].

2.2 Multicast operation in a LAN

It is relatively easy to verify proper operation of IP multicast in a local area network by observing packets of the *Internet Group Management Protocol (IGMP)* [RFC2236]. By watching its operation, we can check both the local workstation and the routers for proper multicast operation within a LAN.

Any network host that wants to receive data in a particular multicast group from sources external to its LAN must signal this intention to its Local Area Network router. This is accomplished by sending an IGMP *Host Response* to that multicast group. The routers also send periodically (every 60 seconds by default) a *Membership Query* to all multicast-enabled interfaces. At least one host on the LAN that is still interested in receiving a multicast group must respond by sending *Host Response*. In IGMP version 2, which is now prevalent, the router also sends the *Maximum Response Time* inside the Membership Query, which gives hosts the opportunity to delay their Host Responses (up to the Maximum Response Time or until another hosts sends the response for the same group). This is a measure against flooding the LAN with the *Host Responses*. Version 2 of IGMP also gave the hosts a possibility to signal that they are leaving the group immediately after the receiving application ceases. We will see an example of IGMP operation later on.

2.2.1 Required software tools

In order to see multicast traffic in a LAN, two programs are needed:

(a) A sender and receiver of multicast packets

(b) Traffic analyzer.

The Mbone videoconferencing tools [Mbone] are often used as the source and sink of multicast traffic. For the purposes of baseline traffic analysis, though, it is often preferable to use simpler traffic generators producing a regular flow of multicast packets in a single group.

The *Multicast Heartbeat* utility [MHB] was developed specifically for testing multicast connectivity in both local- and wide-area networks. The program starts two threads, one for sending and the other for receiving multicast packets. Command line parameters are used for defining the group IP address and destination port and, optionally, for setting the rate and TTL value of the packets. Inside the packets, in an XML-like syntax, the following data are recorded:

- sequence number
- TTL value with which the packet was sent – this helps to determine how many hops the packet travelled through
- timestamp, with the precision of the *gettimeofday* system call

The program can be started on any number of computers arbitrarily distributed in the network. From the terminal output, one can immediately see the sources whose packets are being received and thus determine the one-way multicast connectivity (provided the TTL value was set high enough).

We can also see the number of hops passed by each packet and, if the participating computers are time-synchronised, the value of one-way delay. We also plan to add new features like the option for changing the DSCP value in outgoing packets.

On Unix systems, the traditional tool for network traffic analysis is *tcpdump* [tcpdump]. Recently, another excellent program – *Ethereal* [Ethereal] - has been developed. It is available under the GNU GPL license for Microsoft Windows and various Unix platforms. It offers both text and graphical user interfaces and provides a detailed analysis of network packets for many (even non-IP) protocols in all layers of the protocol stack.

2.2.2 Multicast and IGMP traffic example

Table 2 shows Ethereal output of multicast traffic after an application was started on the host `zma.cesnet.cz`, which sends and receives packets in the multicast group 233.11.36.1. There are five other hosts sending to that group.

Table 2: Listing of the network traffic

```
1   0.000000 zma.cesnet.cz -> 233.11.36.1  UDP Source port: 32786
    Destination port: 56378
2   0.104361 zma.cesnet.cz -> 233.11.36.1  UDP Source port: 32786
    Destination port: 56378
3   0.214347 zma.cesnet.cz -> 233.11.36.1  UDP Source port: 32786
    Destination port: 56378
4   0.324337 zma.cesnet.cz -> 233.11.36.1  UDP Source port: 32786
    Destination port: 56378
5   0.434343 zma.cesnet.cz -> 233.11.36.1  UDP Source port: 32786
    Destinationport: 56378
6   0.544345 zma.cesnet.cz -> 233.11.36.1  UDP Source port: 32786
    Destination port: 56378
7   0.654328 zma.cesnet.cz -> 233.11.36.1  UDP Source port: 32786
    Destination port: 56378
8   0.694179 zma.cesnet.cz -> 233.11.36.1  IGMP Host response (v2)
9   0.764387 zma.cesnet.cz -> 233.11.36.1  UDP Source port: 32786
    Destination port: 56378
10  0.874439 zma.cesnet.cz -> 233.11.36.1  UDP Source port: 32786
    Destination port: 56378
11  0.956993 158.38.60.70 -> 233.11.36.1  UDP Source port: 32897
    Destination port: 56378
12  0.984343 zma.cesnet.cz -> 233.11.36.1  UDP Source port: 32786
    Destination port: 56378
13  0.987631 babar.switch.ch -> 233.11.36.1  UDP Source port: 33998
    Destination port: 56378
14  0.988134 sideral.rediris.es -> 233.11.36.1  UDP Source port: 32859
    Destination port: 56378
15  1.006730 sigma.dante.org.uk -> 233.11.36.1  UDP Source port: 32806
    Destination port: 56378
16  1.017413 vever.urc.uninett.no -> 233.11.36.1  UDP Source port: 65494
    Destination port: 56378
17  1.024829 jade.noc.dfn.de -> 233.11.36.1  UDP Source port: 32797
    Destination port: 56378

...

2656  42.483430 r40-ge1-0-vlan7.cesnet.cz -> ALL-SYSTEMS.MCAST.NET IGMP
      Router query

...

3018  48.105731 zma.cesnet.cz -> 233.11.36.1  IGMP Host response (v2)

...

3114  49.589801 zma.cesnet.cz -> ALL-ROUTERS.MCAST.NET IGMP Leave group (v2)
3115  49.590333 r40-ge1-0-vlan7.cesnet.cz -> 233.11.36.1  IGMP Router query
3116  49.599680 sideral.rediris.es -> 233.11.36.1  UDP Source port: 32859
      Destination port: 56378
3117  49.601668 babar.switch.ch -> 233.11.36.1  UDP Source port: 33998
      Destination port: 56378
...

3144  50.088215 r40-ge1-0-vlan7.cesnet.cz -> 233.11.36.1  IGMP Router query

...

3198  51.039722 sideral.rediris.es -> 233.11.36.1  UDP Source port: 32859
      Destination port: 56378
3199  51.043398 sigma.dante.org.uk -> 233.11.36.1  UDP Source port: 32806
      Destination port: 56378
3200  51.069564 vever.urc.uninett.no -> 233.11.36.1  UDP Source port: 65494
      Destination port: 56378
```

We see that the application immediately starts sending data to the group but, on the other hand, first packets from external sources arrive only after the IGMP Host Response is issued (packet 8). Packet 2656 shows the Membership Query sent by the router `r40-ge1-0-vlan7.cesnet.cz` which connects this LAN to the Internet. The router sends this query to the multicast address 224.0.0.1 (`ALL-SYSTEMS.MCAST.NET`). Looking at the details of this packet, we see the following information:

```
Internet Group Management Protocol
      Version: 1
      Type: 1 (Router query)
      Maximum Response Time : 0x64
      Checksum: 0xee9b
      Group address: 0.0.0.0 (0.0.0.0)
```

The group address is not specified, meaning that the query asks about membership in *any* multicast group. The Maximum Response Time – hexadecimal 64 = decimal 100 – is given here in tenths of millisecond so that the Host Response is expected no later than 10 secs after the query – it actually comes in packet 3018.

Packet 3114 is a IGMP *Leave Group* message, which indicates that the application was shut down. The router continues forwarding the group data but sends twice (packets 3115 and 3144) the Membership Query to make sure there are no other receivers of that group left on the same LAN. This message is now sent directly to the group 233.11.36.1 and the IGMP data are also different:

```
Internet Group Management Protocol
      Version: 1
      Type: 1 (Router query)
      Maximum Response Time: 0x0a
      Checksum: 0xe1e8
      Group address: 233.11.36.1 (233.11.36.1)
```

The Group Address is filled with the multicast address in question and the Maximum Response Time is shorter, only 1 second. The router waits until this period expires and then stops the incoming multicast traffic.

Without going into the details, such a traffic analysis can provide the following basic information:

- When the Host Responses are not present, the host’s multicast functionality is broken.
- When the Membership Queries are not present, the router(s) in the LAN probably are not multicast-capable or have IP multicast forwarding disabled.

In the subsection on Ethernet NIC capabilities we have seen that some adapters may behave differently if they have to handle a number of multicast groups. It is thus important to perform the test with few (or many) groups in order to see that the adapter and its driver operate correctly for any number of groups.

2.2.3 IPGM snooping and CGMP

These two mechanisms may be employed in LANs with Layer 2 switches. Both attempt to constrain the multicast traffic to those switch ports that really have receivers for a particular multicast group. IGMP snooping does it by observing the traffic and decoding the IGMP Host Responses, whereas CGMP is a lightweight protocol in which the same information is communicated to the switch from the router. Intelligent switches should be able to reveal the state of these mechanisms, i.e., the mapping of individual multicast groups to a set of output ports.

3. MONITORING MULTICAST TRAFFIC

If IP multicast does not work properly even after its correct operation was verified in a LAN, it is probably time to contact a network administrator of some of the higher level backbone networks. In order to know who is the right target for such complaints, the user should be able to locate the spot in the network which is the likely source of the problem. This is the role of multicast monitoring tools, or at least those whose results can be given to end users in an easily digestible form.

It has been argued [SA01] that effective monitoring of IP multicast requires more information than general (unicast) traffic monitoring. This is partly due to the existence of multiple receivers and UDP-only traffic but also due to the different scope and complexity of the multicast routing and forwarding mechanism, especially in the inter-domain and sparse-mode case.

In general, the following types of information are needed for multicast management and problem solving:

- Discovery of multicast routes and entire distribution trees; the route or tree may be constructed either from the source to the receivers (using forwarding state in routers) or in the opposite direction (using multicast routing information).
- QoS parameters: packet loss, reordering and duplicates, delay and jitter.
- State of the multicast routing and forwarding protocols: PIM, MBGP and MSDP, in some cases also IGMP
- Statistics on the numbers of sources and receivers that are active in a particular multicast group.
- Traffic statistics: packets and octets forwarded in all participating routers; these can be obtained either from SNMP or, for better granularity, from flow-based statistics.

For the above purposes, an array of free software tools and utilities are available and also several commercial packages. A comprehensive overview of them can be found in [SA01]. Unfortunately, each of the tools concentrates on a particular area and so the integration and evaluation of the data and consequent problem diagnosis still requires a skilled person.

3.1 Selection of tools for the GÉANT network

Multicast monitoring tools can, in principle, use the following sources of information:

- Multicast-related SNMP MIBs
- Flow-based traffic statistics
- Data collected by active probes in the network

SNMP and flow-based data are usually available only through sophisticated network management platforms, which can be accessed by a limited number of network specialists. Also, access to this type of information is not generally granted across administrative domains of NRENs and the international backbone. Therefore, at least in the short term, we see the third approach – active probes – as the most promising one for providing relevant connectivity and QoS information to end users. We tested two such tools that are described in the following paragraphs.

3.1.1 Multicast Reachability Monitor (MRM)

This tool and protocol was introduced in 1999 with the aim to provide an effective monitoring environment for multicast reachability and QoS among both routers and end systems. Soon after it was published as an IETF draft, several experimental implementations have been done – for Cisco and Nortel routers by the respective vendors and for end systems by students of Kevin Almeroth (UCSB) [SA01]. Nortel has started work on a resubmission of the MRM draft, which expired in summer 2000. However, further developement of MRM is now an open issue, as after a discussion at the December 2000 IETF meeting it was recommended to withdraw the MRM draft from IETF. The reason was mainly that MRM uses IPSec, which is not expected to be widely available in end systems in the near future – and an equivalent of the MRM functionality with security features can be provided through SNMPv3 very soon.

An MRM monitoring system is based on a cooperation of a single *MRM manager* with a number of *MRM agents* that are installed in selected nodes of the network. Two variants of the agent software are envisioned: one for routers, with a limited set of features and another full-featured for end hosts. The agents can either be Test Senders (TS) or Test Receivers (TR). The TRs can measure either traffic sourced by the TSs or general multicast traffic. An important part of

MRM is also a communication protocol, which defines the messages used between the MRM manager and MRM agents.

The operation of an MRM system consists of four steps:

1. The tests to be performed are specified to the MRM manager, which then sets up the test scenario and instructs the MRM agents by means of the MRM protocol.
2. For a given time period, the TSs send and TRs receive multicast traffic.
3. The TRs generate reports as instructed by the MRM manager and send them to the latter. For example, the TRs may be asked to send reports regularly or only after a certain threshold is exceeded, etc.
4. The MRM manager processes the received data and presents results.

MRM allows for necessary security mechanisms: Access to the MRM agents may be limited to predefined IP addresses and prefixes and, moreover, the communication between the manager and the agents uses the IPSec Authentication Header [RFC2402].

The available implementations of the basic MRM protocol measure packet losses and packet duplicates. Results are displayed on the central manager but cannot be easily communicated to a large number of end users. The host implementation [MRM] written in C is currently in a beta state. We tested it on SUN/Solaris workstations and found it quite stable. We are not aware, though, of any ports of this software to other platforms.

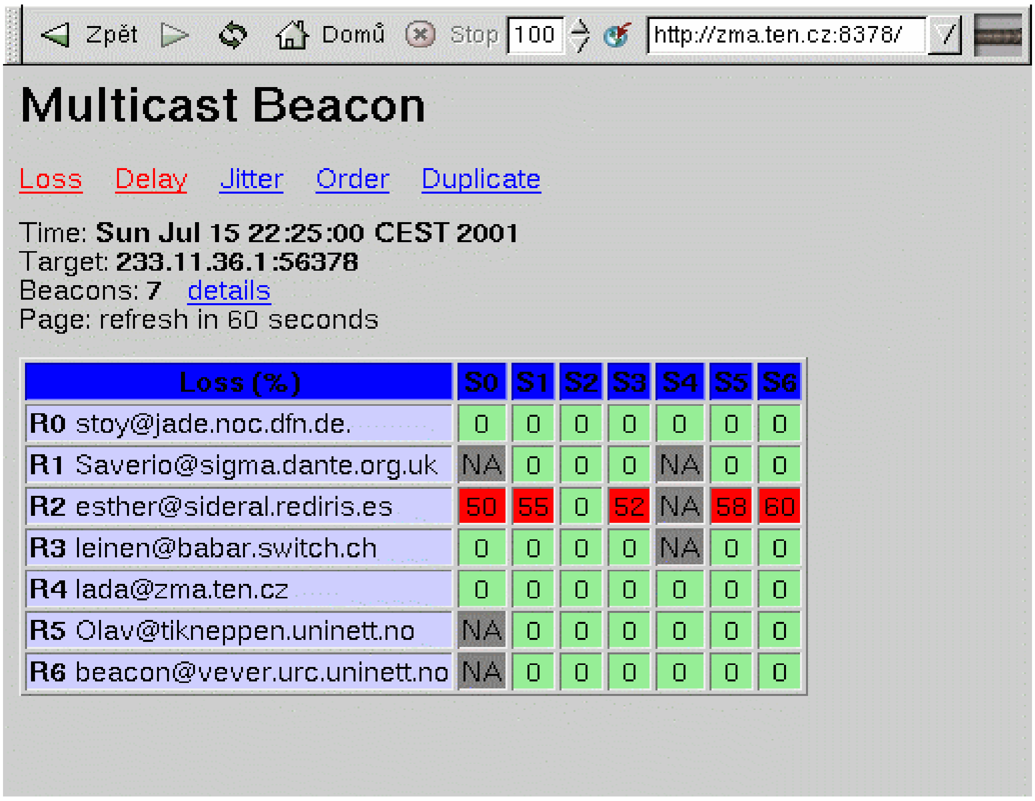
3.1.2 Multicast Beacon

Following the experience of the Internet2 colleagues, we have also been testing another tool – *Multicast Beacon* [Mbeacon] – which is simple and flexible and does not require any special support from routers. Like MRM, it relies on a number of agents spread over the network that communicate with a central server-manager. The agents-beacons simultaneously send and receive multicast packets carrying a packet sequence number and a timestamp. The beacons are statically configured with the multicast group address and port that these packets are sent to and with a few other parameters. In the absence of multicast connectivity problems, the beacons thus form a full mesh where each of them receives the packets from all others. From the packet sequence numbers they can determine packet loss, duplication and reordering. Naturally, in order to get sensible results for one-way delay measurements, the beacons have to be synchronised, e.g., through NTP.

All active beacons send their raw statistics periodically to the Beacon Server via unicast UDP. The server collects the data from all participating beacons and presents them in two forms:

- Web interface (see Figure 1) displays matrices of the five monitored parameters: packet loss, one-way delay, jitter, packets arriving out of order and duplicate packets.

- Special graphical interface accessible though the Beacon Viewer application. Besides the views provided by the web interface, it is also able to display graphically time series of the parameters.



Multicast Beacon

Loss Delay Jitter Order Duplicate

Time: **Sun Jul 15 22:25:00 CEST 2001**
Target: **233.11.36.1:56378**
Beacons: **7** details
Page: refresh in 60 seconds

Loss (%)	S0	S1	S2	S3	S4	S5	S6
R0 stoy@jade.noc.dfn.de.	0	0	0	0	0	0	0
R1 Saverio@sigma.dante.org.uk	NA	0	0	0	NA	0	0
R2 esther@sideral.rediris.es	50	55	0	52	NA	58	60
R3 leinen@babar.switch.ch	0	0	0	0	NA	0	0
R4 lada@zma.ten.cz	0	0	0	0	0	0	0
R5 Olav@tikneppen.uninett.no	NA	0	0	0	0	0	0
R6 beacon@vever.urc.uninett.no	NA	0	0	0	0	0	0

Figure 1. Web interface of the Multicast Beacon server

The main drawback of both interfaces is that they lack the possibility of logging the data to a disk file.

The three programs (beacon, server and viewer) are written in Java, which means that they are, at least in theory, easily transportable to all major hardware and software platforms. However, we also found that some versions of the Java virtual machine are not backward compatible – for example, the server did not work properly under Java 1.3 (both from Sun and IBM). With the newest Java version (1.3.1), though, everything works as expected. Nevertheless, we plan to rewrite the beacon agent program in C language and offer compiled binary versions for diverse platforms – this would eliminate the need for installing the Java run-time environment with each beacon agent.

From the security viewpoint, the communication among the beacons and between them and the server is not protected in any way. Consequently, mere knowledge of the multicast group address and destination port suffices to connect new beacons without authorization or, which is even worse, to attempt malicious attacks. Therefore, if the set of beacons participating in a monitoring session is to be controlled, the IP multicast address should not be published. In this case, the standard Web interface in Figure 1 must be modified so that the address and port is not displayed.

Thanks to the open source character of the Multicast Beacon software, we were able to address some of the missing features:

- Roman Lapacz (Poznan Supercomputing and Networking Centre, Poland) implemented the option for the beacon server that enables it to save the data in a disk file.

- Josef Mašek (University of West Bohemia in Plzeň, Czech Republic) works on the C version of the multicast beacon agent.

3.1.3 Comparison of MRM and Multicast Beacon

The features of Multicast Beacon that are not present in MRM are:

- Measurements of packet one-way delay and jitter.
- Web-based interface for real-time measurements

Table 3 below summarises the features of MRM and Multicast Beacon.

Table 3: Features of MRM and Multicast Beacon.

Feature	*Multicast Beacon*	*MRM*
Measured values	Packet loss, duplicates, one-way delay, jitter	Packet loss, duplicates
Availability of results	WWW Interface Java GUI	On the MRM manager
Agent configuration	Static	Dynamic session started by a central manager
Security	–	Access lists and IPSec
Router Implementation	N/A	Cisco IOS

3.1.4 Beacon Statistics Visualisation

At DFN-NOC, a Beacon Statistics Visualisation (BSV) tool has been developed. The real-time measurements are gathered from the beacon server, then archived and statistics are computed. The gathering process uses the WWW interface of the Beacon server. The query interval used is less than the report interval of the Beacon agents in order to get accurate results. The statistics can be viewed through a WWW interface, see Figure 2 below.

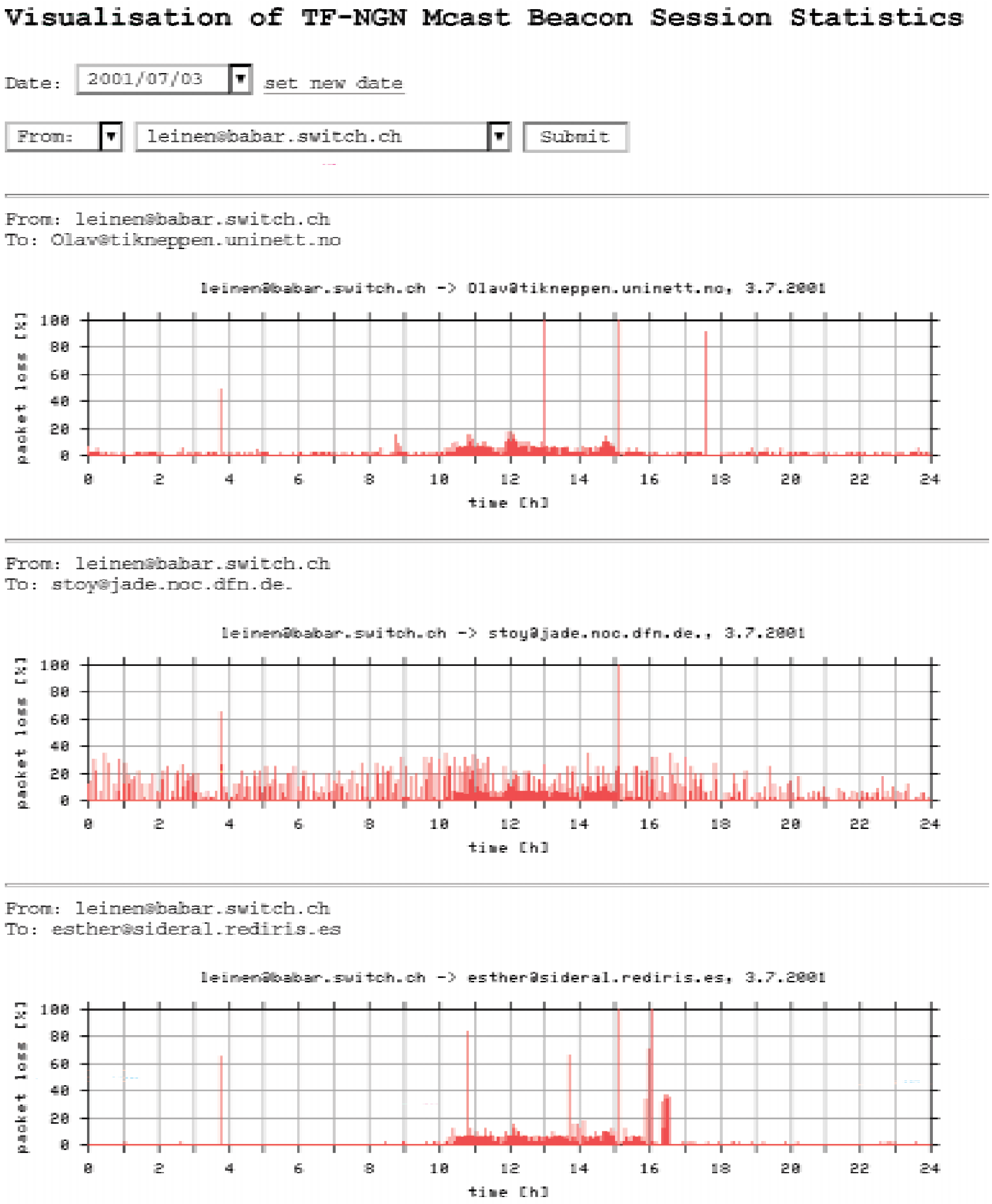
The current version of BSV supports visualisation of the history of packet losses in the last 24 hours. In one view the statistics are shown on a multicast distribution tree from a selected beacon client to all other participants, or from all other participants to a selected beacon client.

The output of the BSV facility in Figure 2 below shows the actual packet loss data of the experimental TF-NGN Beacon Session that is described below. The graphs show packet losses observed by all participating beacon agents for traffic coming from a particular sender (`babar.switch.ch` in this case) and a selected date.

3.1.5 Experimental monitoring environment in TEN-155

Since the beginning of 2001 we have been testing the Multicast Beacon software on the TEN-155 European backbone. The server `zma.ten.cz`, located in Prague, collects the information from seven agents in six countries (see Figure 1), four of them running on the Sun/Solaris platform, two on Linux and one on NetBSD. After solving the Java-related problems mentioned above, both the server and the beacons have been running continuously for about two months.

From this experience, we can conclude that the Multicast Beacon monitoring environment is suitable for deployment in the production GÉANT network.



Visualisation of TF-NGN Mcast Beacon Session Statistics

Date: 2001/07/03 ▾ set new date

From: ▾ leinen@babar.switch.ch ▾ Submit

From: leinen@babar.switch.ch
To: Olav@tikneppen.uninett.no

leinen@babar.switch.ch -> Olav@tikneppen.uninett.no, 3.7.2001
packet loss [%]
100
80
60
40
20
0
0
2
4
6
8
10
12
14
16
18
20
22
24
time [h]

From: leinen@babar.switch.ch
To: stoy@jade.noc.dfn.de.

leinen@babar.switch.ch -> stoy@jade.noc.dfn.de., 3.7.2001
packet loss [%]
100
80
60
40
20
0
0
2
4
6
8
10
12
14
16
18
20
22
24
time [h]

From: leinen@babar.switch.ch
To: esther@sideral.rediris.es

leinen@babar.switch.ch -> esther@sideral.rediris.es, 3.7.2001
packet loss [%]
100
80
60
40
20
0
0
2
4
6
8
10
12
14
16
18
20
22
24
time [h]

Figure 2. Example of Beacon Packet Loss History Visualisation

3.1.6 Recommendations for GÉANT and NRENs

For the GÉANT network we recommend that:

- the beacon agent be installed in all access points of GÉANT from Day 1
- a multicast group be selected from the GLOP range [RFC2770] based on the GÉANT autonomous system number
- the GÉANT NOC set up the beacon server and visualization tools and use it for routine IP multicast monitoring.

Moreover, we strongly suggest that all NRENs participating in GÉANT set up their own Multicast Beacon infrastructure, each using a *different* IP multicast address. This way, multicast traffic can be monitored separately for each administrative domain. In the case of problems, selected end users may also be asked to temporarily join the monitoring session.

The URLs for the GÉANT and NREN Multicast Beacon servers should be widely publicized so that each user may inspect, at his or her discretion, the actual multicast connectivity situation in GÉANT and NREN networks.

3.1.7 Future work

Future development of the Multicast Beacon software in TF-NGN will concentrate on the following items (in the order of priority):

1. Consolidate the history and statistics features so that long term reports on multicast performance will be possible.
2. Provide a way to define the parameters of Multicast Beacon sessions dynamically from a central administration point as it is with MRM.
3. Include some kind of authentication and authorization mechanisms so that the set of cooperating beacons can be controlled.

4. CONCLUSIONS

We have demonstrated that a dependable IP multicast services currently needs an active cooperation of end users and LAN administrators. To this end, the GÉANT project aims at

- educating the users so that they are able to select network hardware for their PC, properly configure the operating system and also eliminate the most common problems in a LAN environment;
- providing a comprehensive monitoring infrastructure whose real-time and statistical results will be publicly available.

This document gives just basic guidelines for both areas. New and detailed information will be available on-line [Multicast-user].

References

[Chariot] http://www.netiq.com/Products-Network_Performance/Chariot

[Ethereal] http://www.ethereal.com

[Mbeacon] http://dast.nlanr.net/Projects/beacon

[Mbone] http://www-mice.cs.ucl.ac.uk/multimedia/software

[Mcast-user] http://www.cesnet.cz/tf-ngn/multicast

[MHB] http://staff.cesnet.cz/~lhotka

[Mmon] http://www.hpl.hp.com/mmon

[MRM] http://imj.ucsb.edu/mrm

[RFC1075] Waitzman, D., Partridge, C. and Deering, S. (1988). Distance Vector Multicast Routing Protocol. RFC 1075, IETF.

[RFC2236] Fenner, W. (1997). *Internet Group Management Protocol, Version 2*. RFC 2236, IETF.

[RFC2402] Kent, S. and Atkinson, A. (1998). *IP Authentication Header*. RFC 2401. IETF

[RFC2770] Meyer, D. and Lothberg, P. (2000). *GLOP Addressing in 233/8*. RFC 2770, IETF.

[RFC1700] Reynolds, J. and Postel, J. (1994). *Assigned Numbers*. RFC 1700, IETF.

[SA01] Sarac, K. and Almeroth, K.C. (2001). Supporting multicast deployment efforts: A survey of tools for multicast monitoring. *Journal of High Speed Networks*, Special Issue on Management of Multimedia Networking, March 2001. http://imj.ucsb.edu/papers/JHSN-00.ps.gz

[SCYLD] http://www.scyld.com/expert/100mbps.html

[tcpdump] ftp://ftp.ee.lbl.gov

LIST OF ACRONYMS

API	Application Programming Interface
ATM	Asynchronous Transfer Mode
BSV	Beacon Statistics Visualisation
CAM	Content-Addressable Memory
CESNET	Czech Educational and Scientific Network
CGMP	Cisco Group Management Protocol
CPU	Central Processing Unit
DFN	Deutsches Forschungsnetz (German Research Network)
DVMRP	Distance Vector Multicast Routing Protocol
FDDI	Fiber Distributed Data Interface
GNU	GNU’s Not Unix
GPL	General Public License
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IPSec	IP Security Protocol
ISP	Internet Service Provider
LAN	Local Area Network
MAC	Medium Access Control
MBGP	Multiprotocol Border Gateway Protocol
MIB	Management Information Base
MRM	Multicast Reachability Monitor
MSDP	Multicast Source Discovery Protocol
NIC	Network Interface Card
NOC	Networks Operations Centre
NREN	National Research and Education Network
NTP	Network Time Protocol
QoS	Quality of Service
PIM	Protocol Independent Multicast
RFC	Request For Comments
SNMP	Simple Network Management Protocol
SNMPv3	Simple Network Management Protocol version 3
TEN-155	Trans-European Network at 155 Mbit/s
TF-NGN	Task Force – Next Generation Network
TTL	Time To Live
UDP	User Datagram Protocol
WWW	World Wide Web
XML	Extended Markup Language