Monitoring Tools for MBone Users

John Robinson and John Stewart

Network Systems Research

CRC -RP-98-008
Ottawa, December 1998

Communications Research Centre
Centre de recherches sur les communications

ABSTRACT

Multicast was introduced into Internet technology in the early 1990s as a mechanism to support multiparty interconnections but primarily to manage network resources efficiently in routing the data during such multiparty exchanges. When combined with emerging technologies for real-time communications and quality-of-service (QoS) management, multicasting is one of the key components for multiparty video-conferencing, distributed simulation and other emerging advanced distributed Internet applications.

To support experimentation with multicast concepts a multicast capable overlay on the Internet, called the MBone, was set up with a few initial sites in the mid-1990s and has grown steadily since then. While the MBone is capable of supporting numerous multicast services, the reality has been that video-conferencing has been by far the predominant use of the multicast backbone network. One of the missing components that is essential for this technology to be successful is a suite of good management tools. Despite the enthusiasm for the multicast services and the research effort that has been invested, in late 1998 the MBone is still an experiment and the technology is still in development.

The Communications Research Centre (CRC) has been participating, since 1995, in MBone R&D Projects with a consortium of European research organisations and industries under the European Union Framework 4 Telematics Programme. A CRC objective is to make the MBone videoconferencing technology usable for non-experts. The CRC research team has also been active in Canada in several CANARIE sponsored projects to support the deployment of multicast on the CA*netII.

Not enough attention has been paid to the design of the monitoring and diagnostic tools that are needed to manage this emerging real-time multicast network. With the interests and needs of the non-expert end-user in mind we have been looking at the existing MBone monitoring tools and have designed new tools to be added to the existing suite. New tools for IP multicast monitoring and for QoS performance diagnostics have been designed and prototypes have been implemented and tested on the MBone with the support of our research partners. This Technical Report presents the results of our research into MBone measurement and monitoring and describes the prototype tools that have been developed.

Table of Contents

1. Introduction

Multicast was introduced into Internet technology in the early 1990s as a mechanism to support connections for multiparty collaboration and to manage network resources efficiently in routing of data during such multiparty exchanges. When combined with emerging technologies for real-time communications and quality-of-service (QoS) management, multicasting is one of the key components for multiparty video-conferencing, distributed simulation and other emerging advanced Internet applications.

While waiting for the arrival of a new generation of switching and routing equipment that supports multicasting, the technique known as tunneling was introduced and a multicast capable overlay on the Internet, called the MBone [1, 2, 3], had come into existence by the mid-1990s. The MBone has grown steadily from the few initial sites set up to broadcast the IETF meetings in the early 1990s. Support for multicasting is now beginning to be found in the latest releases of some network equipment that is beginning to be deployed on some parts of the MBone.

While the MBone is capable of supporting numerous multicast services the reality has been that conferencing with audio, video and shared workspace collaboration tools [4,5] has been by far the predominant use of the MBone. A considerable research effort has gone into the development of these applications but relatively less attention has been paid to the monitoring and diagnostic tools [6, 7,8] that are essential to manage this experimental, real-time multicast network infrastructure. Despite the enthusiasm for the MBone service and the research effort that has been invested the MBone is still an experiment.

The MICE Project [9] played an important role in the initial deployment of an experimental MBone capability, especially in Europe, during 1991-1995. The Communications Research Centre (CRC) joined some of the MICE partners when they carried on with their MBone R&D collaboration, under a European Union sponsored research Project called MERCI (Multimedia European Research Conferencing Integration)[1] [10 , 11, 12] during 1996-1998 and the follow-on Project called MECCANO (Multimedia Education and Conferencing Collaboration over ATM networks and Others) that is currently underway. In Canada, CANARIE is sponsoring several Projects to support the deployment of multicast on the CA*netII [13, 14] and the CRC MECCANO team participates also in those activities.

For the real-time, multicast Internet to succeed one essential component that is missing is the ability to manage and control the videoconference sessions and the multicast network infrastructure. Our research has concentrated on the monitoring tools

[1] MERCI (1996-1998) & MECCANO (1998-1999) are European Union 4th Framework Telematics Projects. The consortium partners include CRC, the University College London (UCL), the French National Institute for Research in Computer Science (INRIA), the Swedish Royal Institute of Technology (KTH), Oslo University in Norway, the German National Centre of Computer Science (GMD), the Stuttgart University Supercomputer Centre (RUS) and the Polish Academic Computer Centre (Krakow) and others.

that are useful for an end-user, in line with our objective to make the video conferencing technology usable for the non-expert.

There is, for an MBone end-user, little visibility into the infrastructure of the Internet service providers and the management and control of those backbone networks must be left to those service providers. We have considered two main issues concerning MBone performance that are of direct concern to an end-user. One is to know how much of the local network capacity is being occupied by multicast traffic (video streams, for example, can be very bandwidth intensive). The second is to measure and monitor the quality of the service across the MBone network and characterise the end-to-end performance.

To address the first concern a tool was developed to measure and monitor the multicast traffic on a local area network. This protocol monitor (MultiMON) collects, organises and displays IP multicast traffic. MultiMON provides both a real-time display and a logging feature that collects data so that the traffic patterns over periods of time can be displayed.

To address the second concern an MBone network quality-of-service (QoS) monitoring system, that collects multicast traffic and extracts QoS statistics from the real-time protocol RTP/RTCP, was designed. Several tools (MERCInari, MReceipt) have been developed based on that design. MERCInari extracts the RTP/RTCP flow performance parameters for real-time performance monitoring as well as recording those statistics for later analysis and performance diagnostics. MReceipt displays reception statistics in a simple form for an individual source; it was designed to show in real time, the distribution and received quality of a sender's signal.

The purpose of this document is to present the results of our study of the end-user requirements for MBone management tools and to describe the new tools that we have developed and contributed to the MERCI Project. The paper is organised as follows: Chapter 2 briefly describes the IP multicast protocol and presents the IPmulticast monitoring tool MultiMON. Chapter 3 reviews briefly the existing monitoring tools, briefly discusses QoS architectures and reviews the performance parameters and metrics that are useful for the description of communication channel performance. Chapter 4 gives an introduction to the RTP/RTCP protocol and QoS parameters. Chapter 5 describes the design of our QoS monitor and the MERCInari and MReceipt tools. A concluding review is given in Chapter 6.

2.0 Monitoring IP Multicast Traffic

2.1 Multicast IP

Multicasting on the Internet is supported by extensions to the Internet Protocol (IP) that have been defined in RFC 1112 [15]. These extensions support the transmission of an IP datagram to a "host group", a set of zero or more hosts identified by a single IP destination address. This is called IP multicasting. A multicast datagram is delivered to all members of its destination host group with the same "best-efforts" reliability as regular unicast IP datagrams.

The membership of a host group is dynamic; that is, hosts may join and leave groups at any time. There is no restriction on the location or number of members in a host group. An implementation that provides full support for IP multicasting allows a host to join and leave host groups, as well as send IP datagrams to host groups. This requires implementation of the Internet Group Management Protocol (IGMP) specified in RFC 1112. To comply with IGMP, a host must join the "all-hosts" group (address 224.0.0.1) on each network interface at initialization time and must remain a member for as long as the host is active.

With the growth and changing nature of the Internet over the past decade it became apparent that the existing version of IP, IPv4 [16], was inadequate to meet the performance and functional requirements for the future Internet. IPv4 is becoming obsolete. In late 1990 the Internet Engineering Task Force (IETF) began an initiative to select the next generation Internet Protocol (IPng), a successor to IPv4 that will support Internet traffic for many years into the future by providing enhancements over the capabilities of the existing IPv4 service. At the July 1994 IETF the IPng Area Directors set up a working group to produce specifications for the core functionality of the new service. The formal name of the IPng protocol is IPv6; the specification of the core set of IPv6 protocols became an Internet Standard on December 1995 [17].

In IPv4 the Internet Control Message Protocol (ICMP) was defined for a gateway or destination host to communicate with a source host, for example to report an error in datagram processing. ICMP has been revised during the definition of IPv6. The protocol has been streamlined and some of the functionality that was present in the IPv4 ICMP, but was not used anymore, has been removed. The protocol has been made more complete by incorporating the multicast capabilities of IPv4[2], including the control functions of the IPv4 IGMP. Group management, for example the procedure by which stations join a group, is identical to that of IPv4 but in general the new ICMP is not compatible with the old one.

---

[2] The monitoring and diagnostic tools described in this Technical Report have been developed for IPv4 only. Upgrading to IPv6 should not be difficult and may be undertaken at some future time.

2.2 MultiMON - An IPmulticast Monitor

IP multicast activity could be the source of a large amount of Internet traffic with, for example, real-time video flows, yet there has been no simple way to monitor that traffic. It is of concern to users and also to system managers to know how much of the local area network capacity and the Internet access lines is being occupied by multicast traffic. MultiMON was developed to detect, collect, organise and display the IP multicast traffic passing by on the network at the location of the MultiMON server. While MultiMON is a very general purpose multicast monitoring tool, it was designed to be of particular use for monitoring local facilities, such as might be under the management of a corporate network administrator.

MultiMON provides both a real-time display and a logging feature that collects data so that traffic patterns can be shown over periods of time. MultiMON has a client/server architecture where data collection is undertaken at the server and displayed in a monitor window of local or remote clients. Figure 1 shows the Client main window. The total bandwidth currently occupied by the multicast traffic is given and a graphical (pie chart) breakdown of the traffic by application type is displayed. This information is presented as a percentage of a bandwidth value (512K in Figure 1) that can be selected by the user to match the local network or the data rate of the incoming line that connects the site to the Internet, etc. The numbers on the periphery of the pie chart are percentages for each traffic type relative to the selected bandwidth.

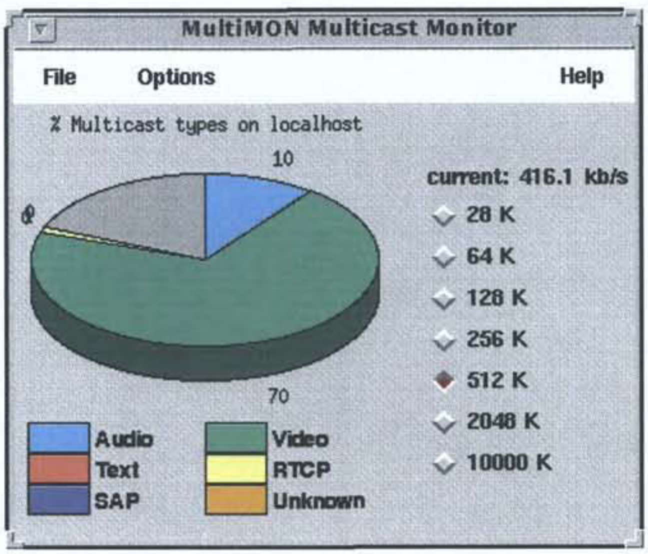

Figure 1: MultiMON Client Main Monitor Window

A second window (Figure 2) displaying graphs of the time variation of the bitrate per application type (time vs. the information given in the main window pie-chart) can be obtained from the "Options" drop-down menu. Other options, such as sub-window creation and SAP (SDR) packets seen are also available from the "Options" drop-down menu.

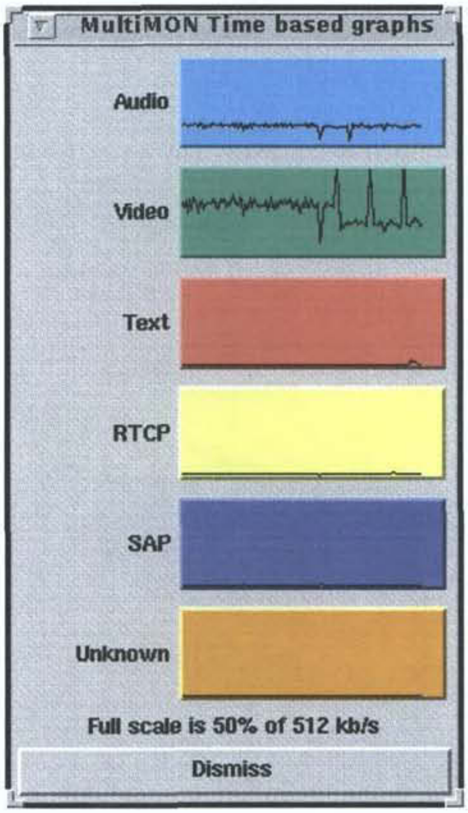

Figure 2: Data Rate vs. Time by Application Category

When MultiMON detects a new multicast session an individual session monitor is launched in a new window (Figure 3). This window can be used to monitor in real-time the traffic arising from that particular session. During a session the occupied bandwidth varies with time. The meter shows both the current bandwidth and the maximum bandwidth detected during the session. It is possible to join the session from an element on the 'Options' drop-down menu.

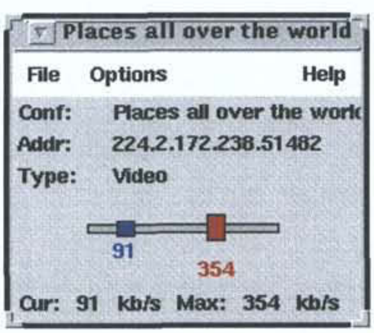

Figure 3: Individual Session Monitor

The MultiMON server sends the information on the traffic that is detected to a data logging file that can be queried after the fact by a historical, long-term monitor. The current release of MultiMON (v2.0) is the first version to include these historical logging features. The bandwidth of all the multicast traffic that is detected can be displayed for a selected period of time in graphical form such as is shown in Figure 4. Audio (red), video (green) and other (yellow) traffic types are distinguished by different colours. It is possible to select the time reference for displays, enabling presentation of the multicast traffic on the network for a selected period of time.

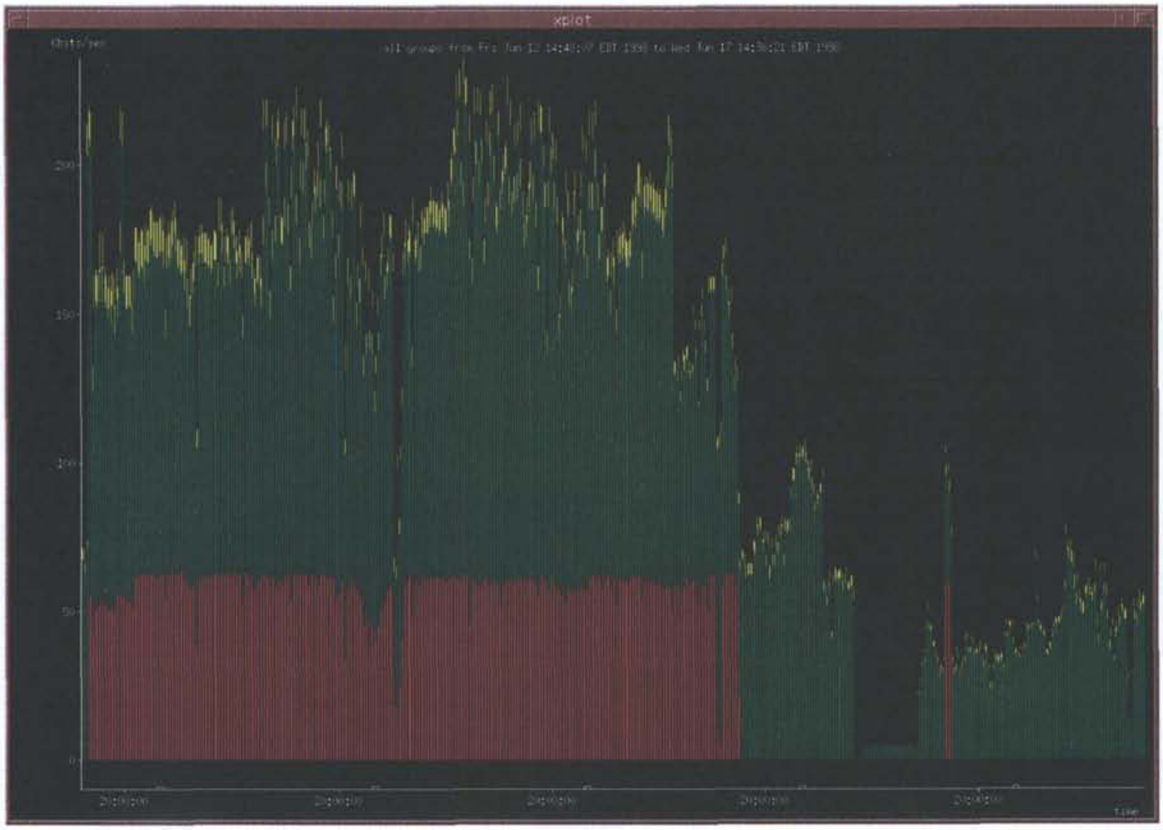

Figure 4: Typical Log of Multicast Traffic over a 24 Hour Period

A breakdown of that data, to show the individual sessions that have been detected and the time duration of those sessions, can be extracted and presented graphically as shown in Figure 5.

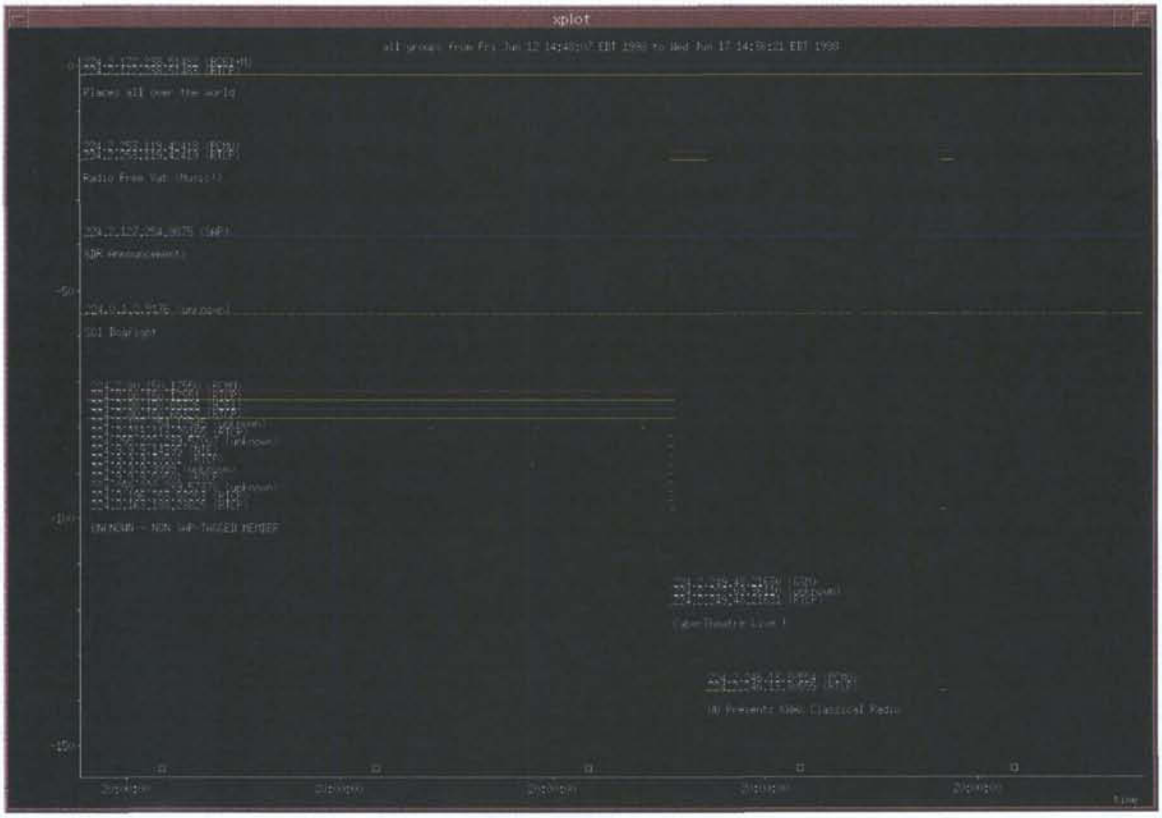

Figure 5: Breakdown by Traffic Type

A prototype of the MultiMON was delivered by CRC to the MERCI partners and released to the international Internet/MBone community for testing and evaluation in early 1997. Based on the response to that release a revised prototype was included in the final MERCI deliverables in late 1997. The final release of the MultiMON prototype with the data logging features included was made in the summer of 1998.

MultiMON is written in tk/tcl and uses the distributed processing (tcl-dp) and object-oriented (stoop) additions to tk/tcl. Also the publicly available utilities TCPDump and Xplot are needed. The current version runs on Sun workstations, but there is reason to believe that it will run on other platforms, (at least the client) including Windows 95 and NT. The MultiMON software can be found on the World Wide Web at the CRC and in several MBone software repositories [4, 5, 28]. The software carries a CRC copyright notice. This technology is available for commercial exploitation through the CRC technology transfer program.

3.0 Monitoring MBone Performance

3.1 Multicast Monitoring Tools

Measurement tools that can monitor performance and diagnose and manage failures are essential to operating and maintaining a reliable multicast service. Instrumentation for management and control of the multicast Internet is a work in progress. Management tools have been under development for the last few years and shareware versions of many of these are available from various Internet repositories [4, 5]. An overview of those tools can be found in [6] where they are categorised (Table 1) based on their reliance on one or more of the following facilities: RTCP source and receiver reports, SNMP MIBs, IGMP trace facility, IGMP ASK_NEIGHBORS message, routing arbiter database, internal structures.

Facility	Diagnostic Tool
RTCP source and receiver reports	RTPMon MSessMon RTPquality RTPdump RTPcast/RTPlisten Duppkts
SNMP MIBs	multicast heartbeat mconfig MStat MView MRTree
IGMP trace facility	MTrace
IGMP ASK_NEIGHBORS message	mrinfo MRTree Map-MBone
Routing arbiter Database	asn asname
Internal structures	TCPDump NetStat mrouted.dump mrouted.cache

Table 1: Multicast Diagnostic Tools Categorised by Facilities Utilised

We suspect that without an expert understanding of multicasting, it is difficult for the average user to interpret the information and displays generated by most of these tools. Furthermore, for even very basic and limited management and control of MBone sessions constant human intervention is needed. This situation contrasts with our concern for the interests of non-expert users and motivated our study of multicast monitoring from the end-users requirements point of view.

3.2 Quality-of-Service Framework

Meeting QoS guarantees is essential for successful interactive multimedia communications. In distributed multimedia systems all the end-to-end elements must work together to achieve the desired application level behaviour. A QoS architecture provides the broad framework that is needed for an overall characterisation of the QoS management that is required. That framework must provide a specification of:

- flow synchronization;
- flow performance;
- level of service;
- QoS management policy;
- cost of service

with the related categories of QoS parameters shown in Table 2. More details can be found in the excellent state-of-the-art reviews presented in [18] and [19].

Category	Parameter Examples
synchronization oriented	skew between sequences
performance oriented	bit rate and delay
format oriented	video resolution, frame rate
user oriented	subjective image and sound quality
cost oriented	connection and transmission charges

Table 2: Categories of QoS Parameters

The real QoS choices available to the user depend on all the system components: the operating system, the transport system (transport processes and communications flow) and the application (media sources and I/O devices) (Figure 6).

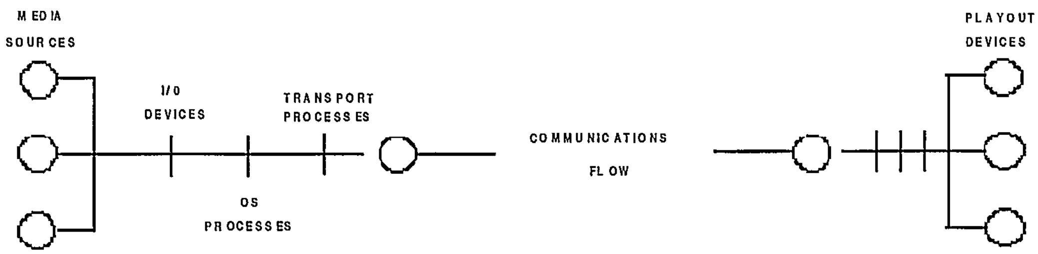

Figure 6: Communications Model for QoS Negotiation

The basic QoS constraints on media sources and operating systems relate to their real-time behaviour . Operating system QoS parameters can be identified at different levels of abstraction where the lowest level parameters include performance, scheduling and memory size. For the transport processes and communications flow the essential QoS parameters can be defined in relationship to a three level view of the communications protocol hierarchy (Figure 7).

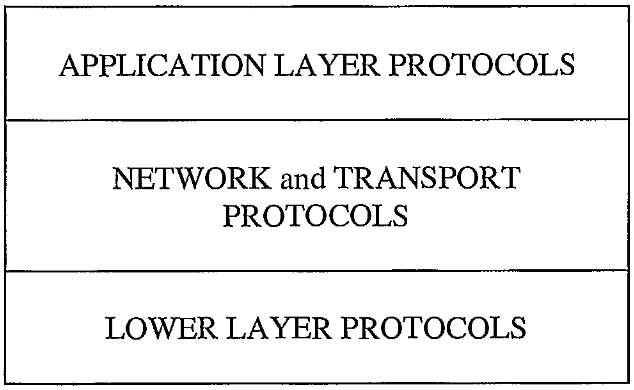

Figure 7: Three Level View of Communications Protocol Hierarchy

The lowest level manages QoS parameters to provide bandwidth and acceptable delay. The top level protocols support an overall QoS negotiation between the components involved. The middle level protocols manage QoS handling over multiple networks. This is the perspective on the QoS framework that is most easily related to the interests of this study, where QoS is being monitored with network and transport protocol QoS information.

3.3 Performance Parameters & Metrics

The Internet was designed to provide a best-effort (connectionless) delivery service and so issues such as performance characteristics and service guarantees were not initially of great concern. Recently however, the move towards the commercialization of the Internet and the interest in deploying real-time applications has forced a growing interest in defining performance metrics (and measurement methodologies) that will help in managing the provision of Internet services [20, 21, 22].

For successful interactive multimedia communications, mechanisms that can provide a good quality service to real-time flows must be designed, developed and deployed. Parameters that can be used to measure the performance of the information flow are an essential baseline in the design of those mechanisms.

From the QoS framework discussed above it is the performance-oriented category of QoS parameters that relate to the information flow. Throughput and delay and additionally loss/error characteristics for the communication channel are the parameters needed for a comprehensive and rigorous characterisation of the QoS [23]. Several metrics associated with channel stability that can be derived from these, examples are loss rate variation and delay (interarrival) jitter, are of particular interest for real-time communications. We return to this topic and provide some rigorous definitions with the description of RTCP and the RTP protocol in the next Chapter.

4.0 RTP - the Real-Time Transport Protocol

4.1 RTCP

The real-time transport protocol (RTP) defined in RFC 1889 [24] provides end-to-end delivery services for data with real-time characteristics, such as interactive audio and video. RTP supports data transfer to multiple destinations using multicast distribution and provides basic services for media synchronization and for QoS feedback. Applications typically run RTP on top of UDP to make use of its multiplexing and checksum services; both protocols contribute parts of the transport protocol functionality. RTP represents a new style of protocol following the principles of application level framing and integrated layer processing proposed in [25]. By itself it does not provide any mechanism to ensure timely delivery or provide other quality-of-service guarantees.

RTP consists of two closely-linked parts: the real-time transport protocol (RTP), to carry data that has real-time properties; and the RTP control protocol (RTCP), to monitor the quality of service and to convey information about the participants in an on-going session. RTCP is based on the periodic transmission of control packets to all participants in the session, using the same distribution mechanism as the data packets.

RTCP performs three mandatory functions:

1. The primary function is to provide feedback on the quality of the data distribution.

2. RTCP carries an identifier for an RTP source called the canonical name (CNAME) to keep track of each participant and to associate multiple data streams from a given participant in a set of related RTP sessions, for example to synchronize audio and video.

3. The first two functions require that all participants send RTCP packets, therefore the rate must be controlled in order for RTP to scale up to a large number of participants. By having each participant send its control packets to all the others, each can independently observe the number of participants. This number is used to calculate the rate at which the control packets are sent.

Each RTCP packet begins with a fixed part (8 octet header) that is followed by structured elements of variable length according to the packet type. RTP receivers provide reception quality feedback using RTCP report packets which may take one of two forms depending upon whether or not the receiver is also a sender. The only difference between the sender report (SR) and receiver report (RR), besides the packet type code, is that the sender report includes a 20-byte sender information section for use by active senders. The SR is issued if a site has sent any data packets during the interval since issuing the last report or the previous one, otherwise the RR is issued.

Both the SR and RR contain zero or more reception report blocks depending on the number of other sources heard by this sender since the last report. Each reception report block conveys statistics on the reception of RTP data packets from a single synchronization source. This uses the timestamp and sequence numbers carried in the

RTP data packet headers. A list of the QoS parameters that can be associated with the RTP/RTCP packet flow is given in Table 3.

packet counts:	
N_{Rx}	**number of RTP packets received**
N_{Tx}	**number of RTP packets sent**
sequence number (S^{i}_{Tx}) counts:	
S_{MAX}	**last sequence number received**
S_{BASE}	**first sequence number received**
S_{WRAP}	**sequence number wraparounds**[3]
time stamps:	
t_{Tx}^{SR}	**NTP**[4] **timestamp on transmission of an RTCP SR**
$t_{Rx}^{SR/RR}$	**NTP timestamp on reception of an RTCP SR or RR**
t_{Rx}^{j}	**timestamp at reception of RTP packet j**
t_{Tx}^{j}	**timestamp on transmission of RTP packet j**

Table 3: RTP/RTCP QoS Parameters

[3] S_{WRAP} is the product of the count of sequence number wraparounds and the number of packets in one cycle. Since the sequence number is 16 bits the latter number will be 2^{16}-1.

[4] NTP is the Network Time Protocol defined in RFC 1119.

4.2 QoS Metrics and Performance Measures

Using the time stamps, the sequence numbers, and the packet counts of incoming packets, QoS metrics can be computed and a number of performance measures can be associated with an MBone session. For example, packet counts and sequence numbers can be used for loss rates; timestamps can be used for delay and jitter.

The QoS parameters and metrics that are carried in RTP data and RTCP SR & RR packets are shown in Table 4.

RTP	RTCP SR	RTCP RR
t_{Tx}^{j}	t_{Tx}^{SR}	J_i
S^{i}_{Tx}	N_{Tx}	Δt_{LSR}
		Fr^{ij}_{LOST}
		t_{LSR}

Table 4: QoS Parameters and Metrics Carried in RTP Packets

Measures of Loss

The number of RTP packets lost (N_{LOST}):

$$N_{LOST} = N_E - N_{Rx} = S_{MAX} + S_{WRAP} - S_{BASE} + 1 - N_{Rx}$$

where the number of packets expected (N_E) is calculated using the sequence numbers of the incoming RTP packets.

In addition to the cumulative counts which allow long-term packet loss measurements using differences between reports, the difference between the last two reports received can be used to estimate the recent quality of the distribution. Loss rate per second can be calculated from differences over the interval between two reports using the NTP timestamp. Since that timestamp is independent of the clock rate for the data encoding, it is possible to implement encoding- and profile-independent quality monitors. The fraction lost measure provides such a short-term characterisation. This becomes important as the size of a session scales up enough that reception state information might not be kept for all receivers or the interval between reports becomes long enough that only one report might have been received from a particular receiver.

The RTP packet loss fraction (Fr^{ij}_{LOST}) over the time interval (i,j) is calculated by receivers and transmitted in the RTCP RR:

$$Fr^{ij}_{LOST} = (N^{i}_{LOST} - N^{j}_{LOST}) / (N^{i}_{E} - N^{j}_{E}) \qquad ; j < i$$

Throughput

A sender report contains cumulative counts for packets and bytes sent from which the average payload data rate and the average packet rate can be calculated. Taking the ratio of the two gives the average payload size. If it can be assumed that packet loss is independent of packet size, then the number of packets received by a particular receiver times the average payload size (or the corresponding packet size) gives a measure of the throughput available to that receiver.

Delay

The RTP transmission delay (Δt_{RTP}):

$$\Delta t_{RTP} = t_{Rx} - t_{Tx}$$

The RTCP transmission delay (Δt_{RTCP}):

$$\Delta t_{RTCP} = t_{Rx}^{SR} - t_{Tx}^{SR}$$

using the RTCP SR NTP timestamp and with t_{Rx}^{SR} measured at the receiver.

The reception report block returned by a receiver contains timestamps, the time of the last sender report $t_{LSR} = t_{Tx}^{SR}$, and the DLSR timestamp (Δt_{LSR}) (providing a measure of the processing delay in the receiver) that are needed to calculate the round-trip delay.

The round trip delay (ΔT):

$$\Delta T = t_{Rx}^{RR} - t_{LSR} - \Delta t_{LSR}.$$

Channel Stability and Congestion

The Interarrival Jitter[5] (J_i) is calculated by receivers and transmitted in the RTCP RR.

---

5 This algorithm is the optimal first- order estimator and the gain parameter value α=1/16 is recommended to give a good noise reduction ratio while maintaining a reasonable rate of convergence [24].

$$J_i = J_{i-1} + \alpha (|D_{i\,i-1}| - J_{i-1})$$

where

$$D_{ij} = \Delta t^{i}_{RTP} - \Delta t^{j}_{RTP} \qquad ; j < i$$

The interarrival jitter provides a short-term measure of network congestion. Packet loss tracks persistent congestion while the jitter measure tracks transient congestion. The jitter may indicate congestion before it leads to packet loss. Since the interarrival jitter field is only a snapshot of the jitter at the time of a report, it may be necessary to analyze a number of reports from one receiver over time or from multiple receivers, e.g., within a single network. (This is why J_i is a running average). It is worth noting that the association of jitter with congestion should be undertaken with some caution; poor link quality can also cause packet loss and a surge in traffic load could cause significant jitter variation without inducing congestion.

5.0 QoS Monitoring with RTCP

5.1 Introduction

A general MBone quality-of-service (QoS) monitoring system has been designed to capture RTCP traffic during multicast events and extract the QoS parameters for performance analysis, and protocol and network diagnostics. The monitor was intended to be general enough to assist users, network managers and protocol developers. The design supports both the presentation of performance in real-time and recording of statistics for later replay and analysis. From that design we have developed the two monitoring tools MReceipt and MERCInari, that are described in later sections of this Chapter.

QoS performance information is exchanged between senders and receivers of real-time RTP streams by the RTCP sender and receiver reports (Figure 8). This information can be used potentially in various ways to manage the real-time flow. A sender might modify its transmission based on this feedback. (For example, the direct use of feedback for control of adaptive encoding is described in [26, 27].) Another possible use is for the sender to use the feedback from the receivers to determine if, and with what quality the transmission is being received. That is the purpose of MReceipt. An end-user can obtain a detailed QoS characterisation of the streams that are being used to send and receive data by extracting and analyzing the QoS parameters carried in RTCP sender and receiver reports. This is the purpose of MERCInari.

Network managers and network service providers, who are not explicitly participating in a session, can use profile-independent third-party monitors (M_x - Figure 8) to receive only the RTCP packets, and not the corresponding RTP data packets. This makes it possible to monitor passively the end-to-end performance of multicast traffic passing through their networks as an aid to diagnosing network problems. While we have chosen to emphasize the needs of the end-user in this study, this option for MERCInari is noted here as an unexplored use for the tool.

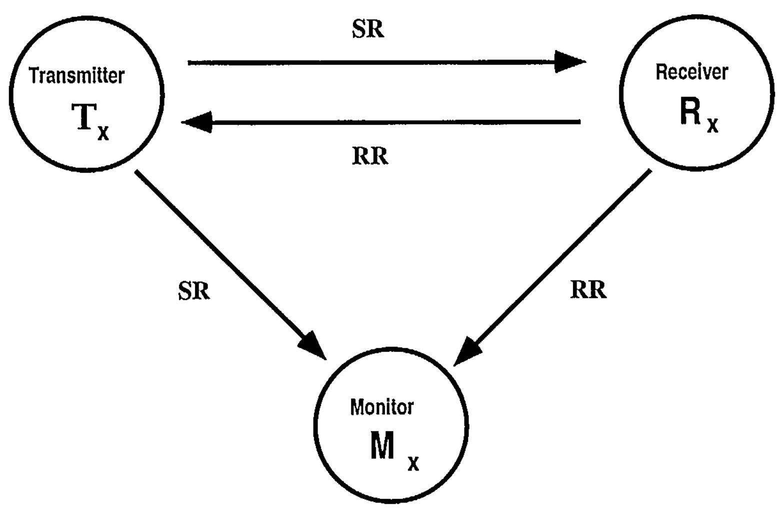

Figure 8: RTCP Packet Flow

5.2 Monitor Design

Figure 9 shows the components in the monitoring system.

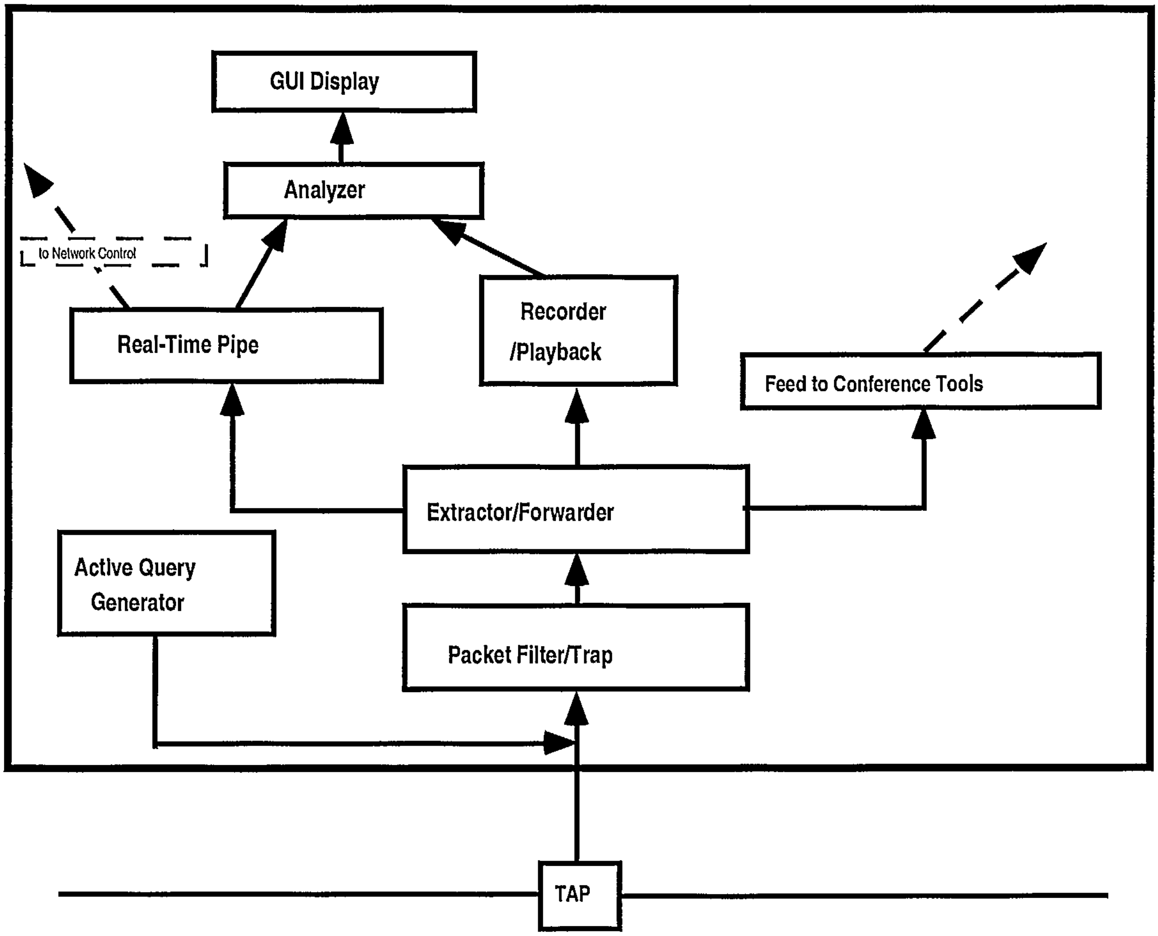

Figure 9: MERCInari Platform Design

These components include:

i) physical network access with a Network Interface Tap to enable reception of network traffic.

ii) a packet trap & filter to monitor the local network and retrieve packets destined to/from a specific network address or addresses.

iii) an extractor/forwarder to retrieve appropriate data (performance statistics) from the packets and deliver to various application modules as appropriate

iv) application modules consisting of:
iv.i) a record/playback process that records selected data from the network to disk for playback to analysis programs (to use for example to refine protocol design).

iv.ii) a real-time pipe: a path for real-time flow of statistics to the analyzer/display. [this is the approach with MReceipt]
(this could also be a route for information to be passed to automated network control processes).

iv.iii) an analyzer & GUI display to provide graphical presentations of performance in real-time, that are understandable to an operator, and after storage and analysis to support development. [this is MERCInari]

iv.iv) a forwarding process specifically to provide a direct feed to the conference tools (video/audio) (where the statistics can be used to adjust encoding, fps, etc. to dynamically optimize the use of the available network service.) [26, 27]

v) an active query process to generate ICMP and/or IGMP queries. Triggered initially by a user; the eventual goal is to let the active monitoring system generate queries that may allow it to interpret networking performance enhancements/degradation. [this has not been implemented]

The monitoring process ideally will introduce as little interference as possible into the tasks being monitored, hence the monitoring program is run on a WorkStation that is not actively involved in the MBone session (M_x - Figure 8). Locating the monitoring platform close to an actively participating MBone WorkStation simplifies the gathering of information directly to/from a contributing session user, however physical co-location is not a requirement.

5.3 MERCInari

MERCInari collects the RTCP traffic associated with a multicast session and extracts QoS statistics in real-time for performance monitoring as well as recording those statistics for later analysis and performance diagnostics. With this information an end-user could quantify the quality of the service from the MBone service provider. This could be useful, for example, in settling disputes with the service providers about service guarantees. Also network managers and network service providers could monitor the end-to-end performance of multicast traffic associated with their networks as an aid to diagnosing network problems. Protocol designers can also use MERCInari to observe the behaviour of RTP in their test implementations. While we have chosen, in these studies, to address the needs of the end-user these latter two applications are other possible uses for MERCInari.

MERCInari captures the RTCP packets on the flow between every pair of participants in a session. There is a viewer that will display the QoS information extracted from each RTCP packet that has been captured and performance plots can be generated.

When MERCInari is started the main control panel (Figure 10) is launched to display a list of the participants that are active in the session that is being monitored.

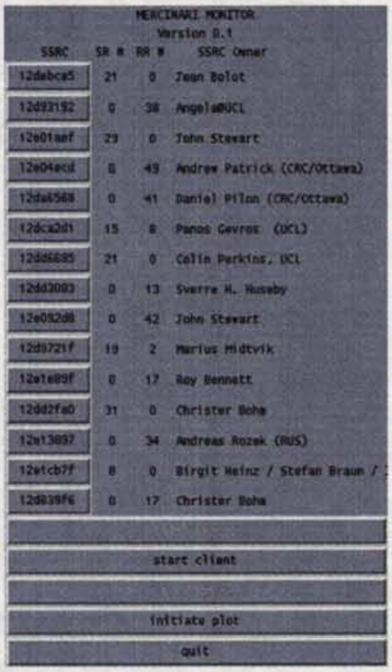

Figure 10: The Main Control Panel of MERCInari v0.1

The 'SSRC' button will spawn a control panel (Figure 11) that identifies the participant with the information from the source description (SDES) type RTCP packet.

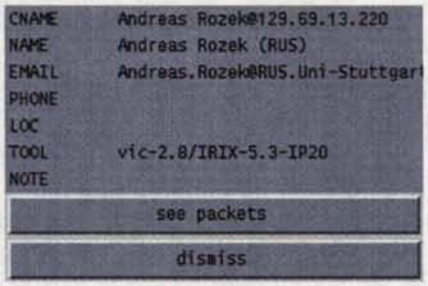

Figure 11: Participant Identification

The 'see packets' button can be used to access detailed information about all the SRs and RRs that have been sent in the user's traffic being studied (Figure 12).

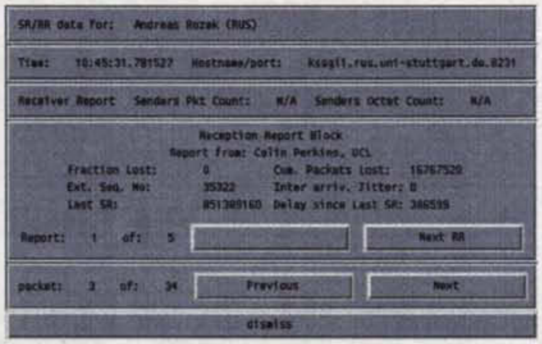

Figure 12: Packet Viewer

From the main control panel the "initiate plot" button generates performance graphs using the QoS information that has been collected (Figure 13). The data sample is from a video session between a sender in England and a receiver in Canada. The lower graph plots the time evolution of the cumulative packet-count from the SRs (top curve) and cumulative packet-loss from the RRs (bottom curve). There was some loss of packets and hence the two curves diverge.

During this work unanswered questions arose about the quality of the implementation of the QoS elements of the RTP specification. Unfortunately we did not have the resources in this project to investigate this further. For a rigorous QoS monitoring and analysis activity there would be value in undertaking an RTP implementation as part of the project.

MERCInari was initially implemented as a standalone tool and later was integrated with MultiMON. It can be invoked from the 'options' drop down menu in the MultiMON main client window (Figure 3).

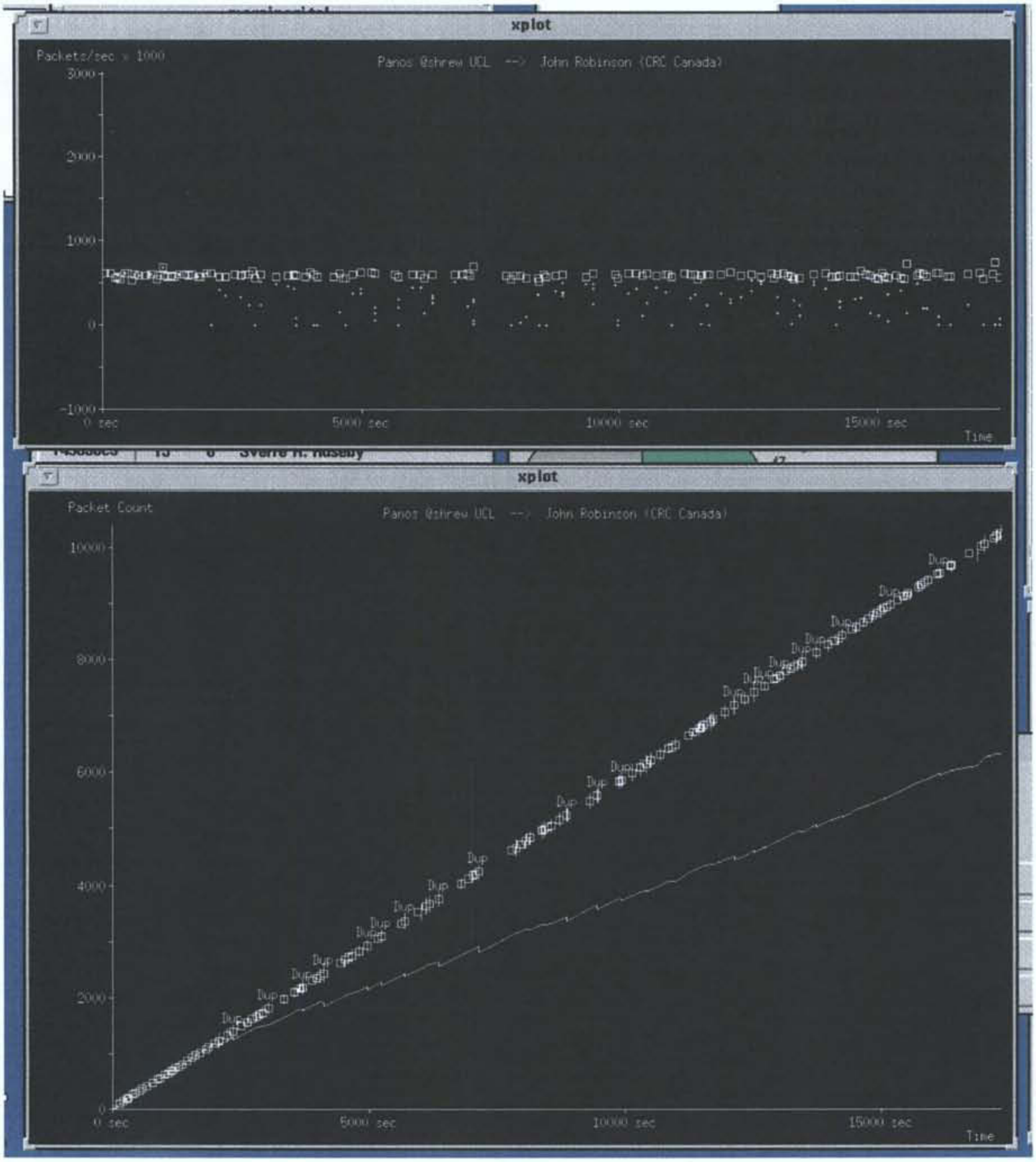

Figure 13: Packet Count & Packet Loss vs. NTP Time

MERCInari was developed on a UNIX platform. The software has been written in tcl/tk. The Packet Trap is based on the public domain tool TCPDump. The program for graphically displaying the statistics is Xplot. Version 1.0 is a preliminary implementation that can be used only in the record/playback mode to extract and display RTCP QoS information. This experience has been important to us as a proof of concept and it has been used with our partners at UCL to investigate the RTCP implementation in the Robust Audio Tool (RAT). MERCInari v1.0 has been released for public use and can be found in the MBone software repositories at UCL[5] and CRC [28].

5.4 MReceipt

While MERCInari can provide detailed and extensive QoS information about the flows in an MBone session a simpler monitor seemed useful to address the concern of end-users who at least want only to know if the other parties in the conference can see and hear them. A simple visual indicator that would display, to a sender, the channel loss statistics from reception reports seemed a useful addition to the suite of multicast monitoring tools. MReceipt was designed and implemented for this purpose

Using the RTCP reception reports as a feedback mechanism the sender can monitor the quality of the signal received at participating receiver sites. MReceipt displays reception statistics in a simple form for any individual source; user names and percentage loss are shown in colour visually indicating the quality of reception. The quality of the video reception from the VIC session shown in Figure 14 is shown in Figure 15. The MReceipt display panel for two points in time is presented and illustrates the progression of warnings from green to yellow to red as the signal quality decays.

MReceipt is written in tcl/tk and uses RTPdump. It can be retrieved from the MECCANO software repository at UCL [5] and from CRC [28]. Included in the distribution are simple instructions that allow a button that will launch MReceipt to be added to the VIC window (Figure 14).

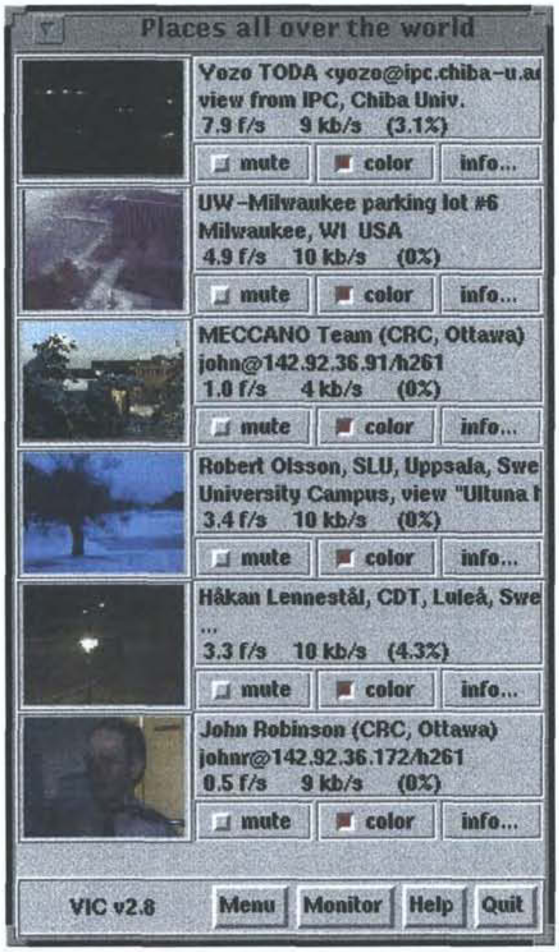

Figure 14: Typical VIC Video Session

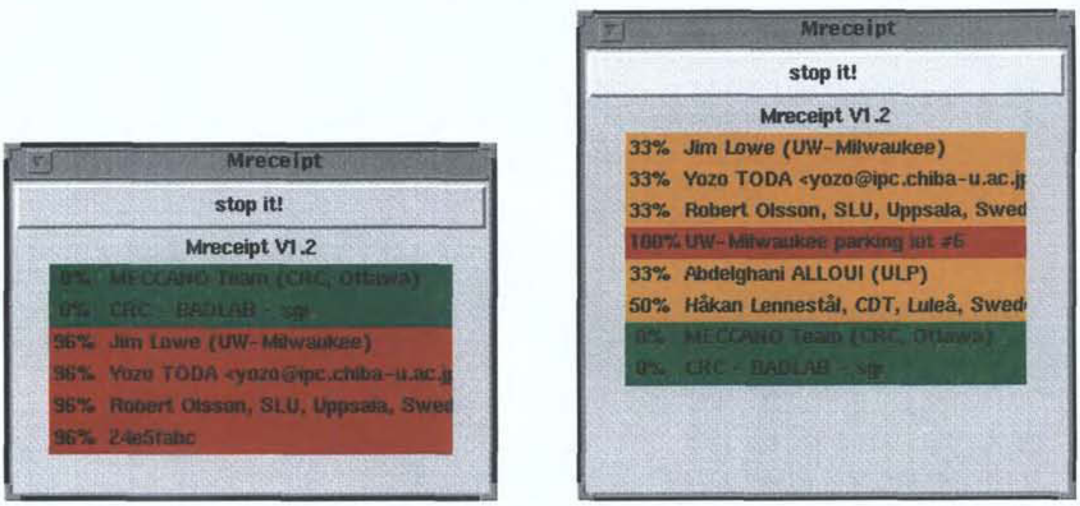

Figure 15:
MReceipt Display Panels for Two Moments in Time
During the VIC Session of Figure 14

6.0 Concluding Review

One of the missing components that is essential for the success of the real-time, multicast Internet is the ability to manage and control videoconference sessions and the multicast network itself. There is in reality very little visibility for an end-user into the infrastructure of the Internet service providers and the management and control of those backbone networks is best left to the service providers. Nevertheless, for an MBone end-user, there are concerns that can be addressed with the appropriate end-user monitoring tools. Our research has concentrated on the needs of those end-users. One concern is to know how much of the local network is being occupied by multicast traffic (video streams in particular are always expected to be very bandwidth consuming). Another is to characterise the service that is being obtained from the MBone network; for example, for a source to know how well they can be seen and heard by the recipients.

A review of existing MBone monitoring tools looked at their suitability for use by non-expert MBone users. It seemed that, in general, an expert understanding was needed to interpret the displays generated by most of the current tools and that those tools are concerned as much with the multicast network as with those aspects of the service that an end-user can hope to manage. One of our objectives has been to support non-expert end-users. New tools for IP multicast monitoring and QoS performance diagnostics, that have been developed with this objective in mind, have been presented in this Report.

To address one end-user concern a tool was developed that can measure and monitor the multicast traffic on a local area network. The protocol monitor (MultiMON) collects, organises and displays IP multicast traffic. MultiMON provides both a real-time display and a historical logging and display capability. It is potentially a general purpose multicast monitoring tool but was developed with particular consideration for monitoring in the local area of the end-users.

To address another end-user concern a quality-of-service monitoring platform was designed to collect multicast traffic and to extract QoS statistics for real-time performance monitoring as well as recording those statistics for later analysis and diagnostics. Several tools (MERCInari, MReceipt) have been implemented based on that design, using RTP/RTCP and the QoS flow performance parameters that can be obtained from that protocol. MERCInari enables collection and display of QoS statistics on an MBone channel and supports the presentation of throughput and packet loss distributions. MReceipt displays reception statistics in a simple form for an individual source; it was designed to show to a sender, in real time, the reception quality and distribution of the transmitted signal.

This research took place at CRC during the international MERCI and MECCANO Projects. Early prototypes were tested and evaluated by partners in those projects after which they were released to the world-wide Internet community for further testing and for general use. The prototype releases carry a CRC copyright notice. The technology is available for commercial development through the CRC Technology Transfer Program. The prototypes can be found in MBone software repositories on the World Wide Web at CRC, in the USA and in Europe.

7.0 References

[1] *MBone Provides Audio and Video Across the Internet,*
M.R.Macedonia & D.P.Brutzmann, Computer, April 1994

[2] *MBone: the Multicast MBone,*
H.Ericsson, Communications of the ACM, August 1994.

[3] *MBone: Interactive Multimedia on the Internet,*
V.Kumar, New Riders Publications, 1996

[4] *MBone Software Archives,*
http://nic.merit.edu/~mbone/index/titles.html

[5] *MECCANO Home Page,*
http://www-mice.cs.ucl.ac.uk/multimedia/projects/meccano

[6] *Multicast Debugging Handbook,*
D.Thaler, B.Aboba, draft-ietf-MBoned-mdh-01.txt, 14October 1998.

[7] *QoS Monitoring for MBone Diagnostics,*
J.L. Robinson & J. Stewart, Net'97 Conference Proceedings,
Dalhousie University, Halifax, Canada, June 1997,

[8] *Managing IP Multicast Traffic: A First Look at the Issues, Tools, and Challenges,*
Stardust Forums Inc., White Paper at the 3rd Annual IP Multicast Summit,
February 1999

[9] *Multimedia integrated conferencing for European researchers (MICE): piloting activities and the conference management and multiplexing centre,*
M.J.Handley, P.T.Kirstein, M.A.Sasse,
Computer Networks and ISDN Systems 26 (1993)

[10] *The MERCI Project: Final Report,*
J. Robinson, CRC-RP-98-004, September 1998

[11] *Technical Innovations deployed by the MERCI Project,*
R.Bennett & P.T.Kirstein, Proceedings of Networkshop 25, Belfast, March 1997

[12] *Final Report RE 1007 - MERCI - Telematics Applications Programme Research,*
P.T.Kirstein & R.Bennett, v4 23 March 1998

[13] *CANARIE MultiCast Test Project Web site,*
http://www.wnet.ca/multicast/.

[14] *MBone Wurcnet Knowledge Networking Project,*

http://www.acs.ucalgary.ca/~mbone/

[15] *Host Extensions for IP Multicasting*,
S.Deering, RFC 1112, August 1989.

[16] *Internet Protocol - DARPA Internet Program Protocol Specification*,
Postel, J. (ed.), RFC 791, September 1981.

[17] *Internet Protocol Version 6 (IPv6) Specification*,
S. Deering, R. Hinden, RFC 1883, December 1995.

[18] *A Survey of Quality of Service Architectures*,
C. Aurrecoechea, A. Campbell, L. Hauw,
Multimedia Systems Journal Vol.6 (3), May 1998

[19] *Distributed Multimedia and QoS: A Survey*,
A.Vogel, B.Kerherve, G.von Bochmann, J.Gescei,
IEEE MultiMedia, Summer 1995

[20] *Towards a Framework for Defining Internet Performance Metrics*,
V.Paxson, INET'96, Montreal 1996

[21] *Terminology for IP Multicast Benchmarking*,
K.Dubray, draft-ietf-bmwg-mcast-00.txt, Nov.'96

[22] *Framework for IP Provider Metrics*,
G.Almes, W.Cerveny, P.Krishnaswamy, J.Mahdavi, M.Mathis, V.Paxson,
draft-ietf-bmwg-ippm-framework-00.txt, Nov.'96

[23] *TIP's Performance Quality of Service*,
S.Bocking, IEEE Communications Magazine, August 1995

[24] *RTP: A Transport Protocol for Real-Time Applications*,
IETF AVT WG: H.Schulzrinne, S.Casner, R.Frederick, V.Jacobson,
RFC 1889, January 1996,

[25] *Architectural considerations for a new generation of protocols*,
D. D. Clark and D. L. Tennenhouse,
Computer Communications Review, Vol. 20(4), Sept. 1990.

[26] *Scalable feedback control for multicast video distribution in the Internet*,
J.-C. Bolot, T. Turletti, and I. Wakeman,
SIGCOMM Symposium on Communications Architectures and Protocols,
Aug. 1994.

[27] *Dynamic QoS control of multimedia applications based on RTP*,
Busse, B. Deffner, and H. Schulzrinne,
Computer Communications , Jan. 1996.

[28] *CRC MERCI Web site*,
http://www.merci.crc.ca