Monitoring of Multicast Networks for Time-Synchronous Communication

---

Überwachung von Multicast-Netzwerken für zeitsynchrone Kommunikation

Der Technischen Fakultät der
Universität Erlangen-Nürnberg
zur Erlangung des Grades

DOKTOR-INGENIEUR

vorgelegt von

Falko Dreßler

Erlangen - 2003

Als Dissertation genehmigt von

der Technischen Fakultät der

Universität Erlangen-Nürnberg

Tag der Einreichung:	22. November 2002
Tag der Promotion:	9. Mai 2003
Dekan:	Prof. Dr. rer. nat. A. Winnacker
Berichterstatter:	Prof. Dr. rer. nat. F. Hofmann
	Prof. Dr.-Ing. U. Herzog

Acknowledgements

Primarily I would like to thank Bettina Krüger. Without her continuous support, her helping hand and the love she showed me the work would have probably never been completed.

I want to thank Prof. Fridolin Hofmann. He was always ready to read my work and to discuss emerging problems.

Dr. Peter Holleczek, I wish to thank for his support and for the ideas which promoted my work.

Additionally, I would like to thank Prof. Ulrich Herzog as a second source of expertise.

I would like to thank all coworkers at the Regional Computing Center in Erlangen and at the Department for Computer Science 4 for their continuous support and assistance.

Not least, my thanks go to my parents. Their support made my study possible.

Finally, I want to thank the revisers of this work. In particular I would like to mention Dr. Phineus Markwick and Dr. Gerd-W. Krüger.

Abstract

The internet is increasingly used for the transmission of multimedia content. Real-time applications such as video conferences and TV broadcasts are in the center of attention thereby. The development and the implementation of IP multicast is the basis for a meaningful use of these new services. Additionally, a minimum quality of service is necessary for the transmission of high quality multimedia data.

Unfortunately, IP multicast is still a complex and fault-prone technology. Only a few ISPs are capable to deploy these techniques in a correct and stable manner. Furthermore, far few or no mechanisms are implemented in the large backbone networks, which offer or even guarantee a higher quality of service than the typical best effort behavior of IP networks.

It is one of the most important tasks to develop tools for monitoring multicast networks. The objective of such tools is to estimate the reliability of a particular network connection as well as the measurement of the currently available quality of service.

In the context of this work, a new concept is being described which allows the measurement of the reliability and of the quality of service in distributed multicast networks. Attention is paid to the scalability of this specification. In order to ensure this scalability new mechanisms are developed which allow the measurement even in really large and distributed environments.

It has been turned out that the measurement methods as well as the optimal distribution of the measurement stations are closely coupled to the objective, the guarantee of the function and of the quality of particular multicast services.

In the present work, these mentioned aspects are examined. A new model is being specified which integrates information about the available network infrastructure as well as about the most important multicast services. Based on the modeled data an implemented routing algorithm can be used to identify the parts of the network which are used by the active and by the scheduled multicast services respectively. Additionally, it provides information about the required measurements of an optimal estimation of the achievable quality of service.

In conclusion it was possible to define and to test a framework which complies with the mentioned goals. Based on prototypical implementations the functionality is tested in a lab environment as well as in real networks.

Table of Contents

1 Introduction

IP multicast is one of the leading technologies which has an important influence on the contemporary development of new multimedia applications to be used on the internet. Considering the history of the internet and its usage, it is possible to identify three fundamental development phases. In the early stages of development, only experts were able to operate the internet. After a short period of time several applications, such as email and discussion forums like NetNews were established. A new era in the history of the internet began with the introduction of the world wide web (WWW). Simultaneously, a dramatic increase in the public usage of the network can be identified, stimulated by the introduction of easy-to-use graphical interfaces and the developments in the comprehensive usage scenarios of the hypertext oriented WWW.

Today, the next development phase is already underway. New technological advances, such as the multiplication of the available bandwidth both in the backbone networks of the large ISPs (internet service provider) as well as in the private sector, are rapidly expanding. These new advances have promoted substancial interest in internet based multimedia applications such as real-time audio or TV broadcasting. IP multicast development plays a specific role in this field: In general, most multimedia applications involve the simultaneous transmission and distribution of data to numerous destinations. IP multicast achieves this task using only the minimum possible amount of network and server resources.

The deployment of IP multicast started in about 1992 with the implementation of the MBone, the multicast backbone [73]. Steve Deering explicitly defined the host model of IP multicast [52], [55], as described in detail in his Ph.D. thesis [53]. This provided the basis for future extensions and developments based on his original work. In comparison to the early days of IP multicast, significantly improved protocols and much more stable implementations are readily available nowerdays.

Before the advent of IP multicast, developers of internet-based applications had to choose between either unicast transmissions or broadcasts. A unicast connection is defined as the transport of packets over a network from a single sender to exactly one receiver. The specific protocol is described in [164]. Even in local area networks, it is often necessary to address numerous, or maybe even all stations simultaneously and in order to achieve this, the broadcast transmission was introduced. Broadcast transmission involves the simultaneous delivery of a data packet from a single sender to all stations in the network.

For certain applications, multi-destination transmission accross different networks is essential and as early as 1978 a means of achieving this task had already proposed [48]. This led to the design of IP multicast.

The advantages of a unicast transmission are obvious: Given that most applications work according to the host-to-host communication model, if more than a single destination exists, the corresponding number of parallel connections has to be created. Low server system efficiency leads to a potential bottleneck during this process, resulting in a slowing down of transmission rates and an overloading of the network.

For example, if *n* video transmissions of 5 Mbps each occur at the same time, a 100 Mbps connection processes a theoretical maximum of 20 streams. If the video is broadcast, the server system has to stream it only once into the network. Therefore, the load is removed from the server and its network connection to the network itself, which is responsible for transmitting the data to all connected hosts. Additionally, an overload situation may occur at the end systems, which receive and analyze all the data-flow packets, dropping those that are unnecessary.

Considerations such as those provided in the above example, have led to the development of IP multicast. The principle of a multicast transmission is to send data packets only once into the network, which is then responsible for the duplication of these packets at all the required and optimum places, and the delivery of the data to all interested receivers.

In general, one can make a distinction between protocols and technologies used for intra-and inter-domain multicast routing [92], [112]. End systems employ the IGMP protocol (internet group management protocol, [51]) in order to tell the directly connected routers which multicast groups they are interested in. Routing mechanisms in the domain ensure the transmission of multicast packets to all the participants of the particular group.

Multicast routing protocols [34] are divided into two basic classes called dense-mode and sparse-mode protocols Dense-mode protocols work according to the push principle. All the data are distributed to all systems until they declare that they are not interested in receiving the traffic (e.g. if no active receiver exists). In the case of sparse-mode protocols, which operate according to the pull principle, multicast data flows are requested by an explicit announcement. The two classes differ in terms of scalability and complexity.

In the early days of IP multicast, a logical network overlying the internet was created and given the name MBone. The structures of this network are only used for multicast transmissions. Using tunnels between multicast enabled routers, the single multicast networks were interconnected. Nowerdays, these first structures have been replaced by a natively multicast routing [8]. Issues regarding the security and the scalability of the system were addressed by new developments such as IGMPv3 [38] or SSM (source specific multicast, [111], [24], [25]).

IP multicast has been used intensively over the last several years for various applications. Initially multimedia applications were installed [121], [127] such as video conferencing between the directors of the computing centers in Bavaria (project TKBRZL, [107]) and TV broadcasts, which distributed recorded videos of lectures (project Uni-TV, [62]). Other examples include the transmission of stock exchange rates [136] or the synchronization of clocks using NTP (network time protocol, [146]).

Due to the high complexity and the incomplete or possibly unavailable implementations, the operation of IP multicast is still very expensive for ISPs [61]. In addition, only a few people have the knowledge to operate large multicast networks. IP multicast users require a high quality of service, arising from the fact that most contemporary real-time multimedia applications make substancial demands on the available quality of service levels. A ’catch 22’ situation appears: The ISPs are not willing to invest large sums of money for network equipment and employee training until a suitable high consumer level is reached. On the other hand, customers cannot use the currently available networks for their required new IP multicast applications because these networks cannot provide the required quality of service.

Today, it is generally considered that the solution of this problem lies in the development of automatisation procedures designed to rapidly locate the problem areas in the IP multicast network. The primary goal is a fully automatic test of the multicast functionality including a current network quality of service analysis, with the possibility of an expected service quality forecast [67]. Since 1999, several appropriate tools have been developed [6]. Some examples will be mentioned here: MHealth [131], which comprises the mtrace tool [80] and a graphic frontend, the multicast reachability monitor (MRM, [11]), and the multicast beacon [41] which is a project of the NLANR (national laboratory for applied network research). All these tools allow one to measure the quality and connectivity in a multicast network. Unfortunately, all of these methods are insufficiant, as not all the relevant quality of service parameters are examined. In general, only simple accessibility tests and low level failure location methods are available.

The goal of this work is to describe a novel approach which avoids the incompleteness of the available implementations based on a completely new analysis of the problem. The primary goal is to test the multicast infrastructure starting with the requirements of the most important applications. The functionality, i.e., the connectivity within the network is measured as well as the quality of service. With the gathered information and measurement results, it will be possible to predict the quality of scheduled applications.

The chapters are organized as follows: First, an introduction to the basic mechanisms of IP multicast including an analysis of potential problems and a summary of the aims of this work is provided. Section 2 "Multicast Infrastructure" describes the currently used multicast routing algorithms and strategies for intra- and inter-domain multicast routing. Additionally, well-known multicast networks are analyzed and discussed. The campus network of the University of Erlangen-Nuremberg, the Bavarian university network (BHN), the German research network (G-WiN), and the global multicast routing in the internet are used as examples. An overview of typical applications such as audio and video tools, collaboration applications, and others including the typical multicast services, broadcasts and conferences is provided.

Section 3 "Quality of Service in Multicast Networks" describes the most important quality of service parameters for data transmissions over the internet. In particular, the reachability, the packet loss ratio, the delay, and the variation of the delay are all discussed in detail. A number of working groups led by the IPPM WG (IP performance measurement working group) of the IETF (internet research task force) are investigating the possible ways to measure these parameters. Measurement tools for IP unicast connections are being developed as well as metrics to evaluate the results and to estimate the quality of service properties of a network connection. Additionally, an analysis and test of initial approaches and concepts to measure the quality in an IP multicast environment are provided. Finally, an overview of the methodologies to either increase or guarantee the quality of service for particular transmissions over the internet is provided. The integrated services architecture and the differentiated services architecture (both developed at the IETF), layered multicast transmission, and the development of protocols allowing a reliable data transmission of multicast packets are taken as examples.

Measurement methods allowing the analysis of the behavior of multicast applications transmitting multimedia content are provided in section 4 "QoS Requirements of IP Multicast Applications". Where the results of sample measurements are also shown. The different measurement methods are here defined and distinguished: The objective tests and the subjective valuations, as well as between tests in a real world network and such in a well-controlled lab environment. As the measurement equipment is very expensive and sometimes insufficiently implemented, a new IP based impairment tool has been developed.

The presentation of the new approach to measure and to analyze the quality of service parameters in an IP multicast network is divided into three main topics of interest. Section 5 "Modeling IP Multicast Networks and Services" discusses an object oriented approach to represent an IP multicast network including the employed multicast services. The model, which has been specified in UML (uniform modeling language), allows one to incorporate static properties of network components as well as dynamically measured information. The structure of the object model is based on the TCP/IP layer model. The model allows to integrate routing algorithms in order to calculate optimum paths through the network for a particular service. Usage scenarios and the capabilities of the prototypical implementation in JAVA are shown.

In section 6 "Definition of a Metric for Multicast Services", an overview of a metric which allows the comparison between the conditions in a multicast network and the quality of service requirements of a particular application is provided. Another goal of this metric is to create so called SLAs (service level agreements). Such SLAs, defined in a contract between the ISP and the customer, state minima for the offered quality of service. The customers can rely on the availability of the agreed services. Parameters of the metric and some basic calculation methods are discussed as well.

Section 7 "Multicast Quality Monitor (MQM)" describes a novel approach to measure the quality of service in a multicast network. Based on the analysis of already existing methods, and on the model discussed in section 5, new measurement methodologies are specified and implemented allowing the supervision of large and complex multicast networks. After a description of the structure and the basic principles, the individual measurement methods, the test of the connectivity and the measurement of the quality of service, are discussed. Although some of these methods have already been implemented in other systems, some, such as the one-way delay in IP multicast networks, and measurement equipment communication are completely new.

Final considerations about the state of this work are summarized in section 8 "Summary". To conclude with an outlook to further activities is provided.

2 Multicast Infrastructure

In order to study the mechanisms to measure service quality parameters, an analysis of the basic technologies and concepts of IP multicasting is required. In this introduction, a summary of the available contemporary multicast routing protocols is provided as well as an overview over the existing multicast networks. This is followed by a discussion of multicast applications and associated examples.

2.1 Multicast Routing Protocols and Strategies

To identify potential bottlenecks in IP multicast networks, it is necessary to study the essential concepts underlying the mechanisms of the most popular multicast routing protocols [152], [153]. Ramalho [168] provides an excellent overview in his paper "Intra- and Inter-Domain Multicast Routing Protocols: A Survey and Taxonomy" which also covers complexity problems. Another summary is given by Sahasrabuddhe and Mukherjee [176].

In the next few subsections, protocols used for group management, intra- and inter-domain multicast routing are introduced [145]. Particular attention is paid the possible service quality leakage resulting from protocol design.

2.1.1 Group Management

The group management is responsible for two main tasks:

- communication between the end system and local router, and
- selection of a feasible group address.

The first one, the communication between the end system and the local router, is done by using IGMP (Internet Group Management Protocol). Unfortunately, there is to date no specific criterion for the selection of the group address. Anyway, some known approaches are analyzed.

2.1.1.1 Internet Group Management Protocol (IGMP)

The Internet Group Management Protocol [50], [51] is responsible for the communication between end systems and locally connected routers. The routers require information about the group membership of their directly connected hosts to compute their multicast forwarding tables and to provide a network wide multicast connectivity.

There are three types of communication in IGMP:

- maintain a group: the router performs a so-called query-response process every 60 sec. to update its information about current multicast memberships
- join a group: a host informs its routers that it wants to receive traffic for a particular multicast group

- leave a group: a host informs its routers that it is no longer interested in traffic for a particular multicast group.

Figure 2.1 shows the query-response process of IGMP. A selected router acts as the IGMP querier. In IGMPv1 no definition of an election mechanism is provided to select one router as the querier. This problem is solved by the multicast routing protocol. The designated router (DR) of the multicast routing protocol is also elected as the IGMP querier. In IGMPv2 an election mechanism exists, particularly to allow also non-multicast routing devices such as multicast capable LAN-switches to act as IGMP querier.

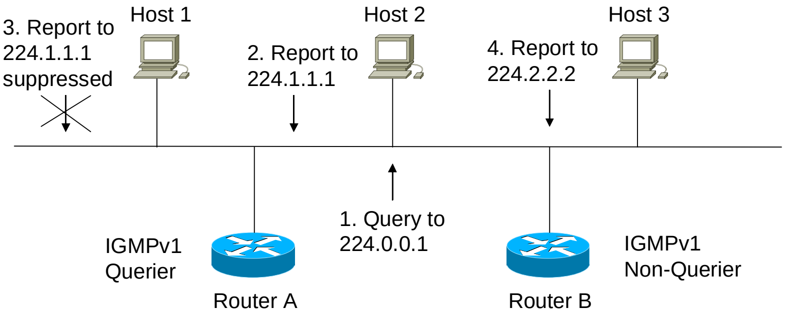


Figure 2.1 - IGMP Query-Response Process

(1) Router A (IGMP querier) sends membership query messages to all multicast hosts (224.0.0.1). This action is repeated periodically (every 60 sec.).

(2) Host 2 first responds first by sending a membership report for group 224.1.1.1 for which it wants to receive traffic.

(3) Host 1 who is also a member in group 224.1.1.1 receives this report and suppresses any additional report for this particular group.

(4) Host 3 reports to 224.2.2.2.

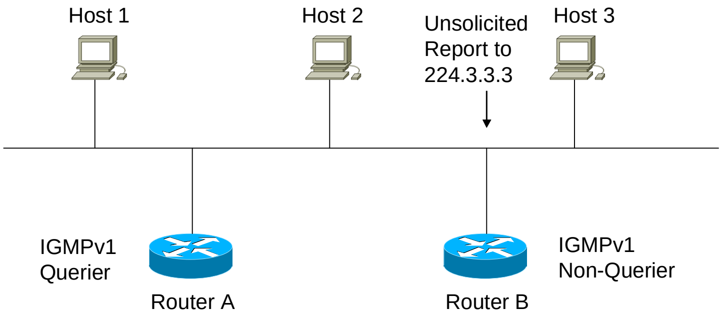


Figure 2.2 - IGMP Membership Report

Figure 2.2 shows the join mechanism. Technically, a join is just the same membership report which is also used in the query-response process. Due to the fact that everyone says "join a multicast group", an unsolicited report is called a join.

The design of the host model [51] specifies that a join is only required to receive multicast traffic. To send multicast packets, there is no requirement for joining the group. Unfortunately, this results in connectivity problems between sparse and dense mode networks (see section 2.1.2). In order to avoid this kind of problem, most of the multicast applications join a multicast group before sending data even if they do not need to receive any packets.

The time between the submission of the membership report and the arrival of the first packets in a multicast group is called the join latency. The join latency is very important for applications such as TV broadcasts. If a user tunes into a channel, he wants to receive the signal as fast as possible. If the router is already receiving packets for that group, the join latency can be very low, otherwise, the join latency depends on both the multicast routing protocol and the distance from the sender.

There is no leave message in IGMPv1. The router holds a timer for each active group. Every 60 sec. the query-response process is initiated. If the router gets no answer after 3 queries, it times out the group and stops the transmission of packets. So the leave latency may be as long as 3 minutes. This is a severe problem. Using the same TV broadcast analogy, if a user zaps through the channels, the network will be quickly overloaded. Typical reasons for this are the high bandwidth usage for each channel, the capacity of the network towards to the user, and the high leave latency. This problem has been solved by the introduction of IGMPv2 [83]. Figure 2.3 shows the leave mechanism of IGMPv2.

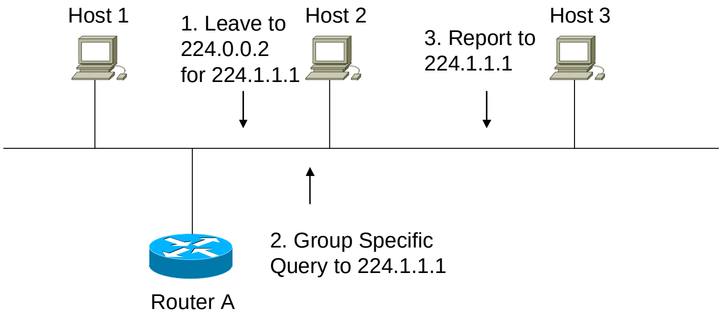


Figure 2.3 - IGMPv2 Leaving a Group

(1) Host 2 sends a leave message for group 224.1.1.1 to 224.0.0.2.

(2) The router answers with sending a group specific query. The group specific query has been introduced to prevent the full query-response process for each received leave message.

(3) Host 3 is still a member in 224.1.1.1, so it sends a membership report to the router.

However, one problem still exists: Each member of a group receives all the traffic to this group. Because there is no centralized group reservation mechanism, more than one application may choose a particular group for its transmission. Unfortunately, this also opens a security hole. Attackers who want to disturb active sessions may do so by sending a large number of packets to the multicast group used by that application. Regrettably, there are many kids around the world who enjoy giving offense. The next version of IGMP, IGMPv3 [38], attempts to solve this problem. Join and leave messages are extended by (S,G)-pairs. This allows the support for source filtering. In IGMPv3, hosts confer to the router, which multicast group containing the traffic they want to receive as well as which senders they are interested in [91]. Therefore, IGMPv3 is also the basis for Source-Specific Multicast (SSM, [110]).

Each multicast capable end system is required to listen on multicast address 224.0.0.1 (all-systems.mcast.net) to make the query-response process work. The routers are required to listen to all multicast groups. The reason is very simple. First, a membership report is sent to the multicast group for which the host wants to receive multicast traffic. The router has to receive it to rebuild its multicast forwarding tables. Secondly, multicast senders are not required by the standard to join the multicast group first, so they may start to transmit data to a particular group at any time and the router is required to instantly forward these packets.

The first implementations of IP multicast had large performance problems due to this mechanism. Today, most manufacturers move the task of analyzing the multicast packets in special ASICs, but there are still software-only routers which are about to stop to perform IGMP actions if there are some high bandwidth multicast streams active.

2.1.1.2 Multicast Addressing

A common problem of all multicast applications is how to choose the multicast group address. This address has to be selected very carefully. If more than one application uses the same group, both data streams may interfere, or they may unnecessarily stress the network connections as well as the end systems. Three different types of address allocation are defined [208]:

- static allocation,
- scope-related allocation, and
- dynamic allocation.

For some well-known applications, the Internet Assigned Numbers Authority (IANA) has reserved single static addresses [2] and a few multicast applications have chosen other statically defined multicast groups on their own which are no longer available to other applications [151].

address range	scope
239.0.0.0-239.255.255.255	administratively scoped multicast address space
224.0.0.0-224.0.0.255	link-local scope
224.0.1.0-238.255.255.255	global scope

Table 2.1 - Administrative Regions in Multicast Address Space [208]

Additionally, the multicast address range (class D, 224.0.0.0/24) has been divided into a number of administrative regions shown in table 2.1.

Packets with a link-local address will not be forwarded by any router. Administratively scoped addresses are used in a single organization [142] and only packets with an address in the global scope range are forwarded through the entire internet.

In order to have unique multicast addresses available for individual organizations and routing domains, several approaches have been initiated. One example is the multicast-scope zone announcement protocol (MZAP, [98]). This protocol allows a dynamic allocation of multicast address space.

The term session is used to describe an application involving one or more multicast groups for its data communication. Most multimedia applications, such as video conferencing, use the session announcement protocol (SAP, [97]) in conjunction with the session directory (SDR, [93]) to choose and to announce multicast groups for their transmissions. The SDR tool chooses some unused multicast groups to initiate a session. This session, for example a video conference can be joined by any user. The shared usage of a single multicast address by more than one application can be prevented by using this mechanism. Another benefit of using the SDR is its capability to provide a set of data about each session. Examples are the name and the purpose of the session, the time when the session becomes active and the type of media streams used in this session. The SDR announces all this information in addition to the chosen multicast addresses by using the session description protocol (SDP, [96], [100]).

The current approach is to provide a set of protocols to choose globally unique multicast addresses. The multicast address allocation architecture [195] consists of local multicast address allocation servers (MAASs) and three protocols [208]:

- a host-server protocol, which allows a client to request an address from its local MAAS: multicast address dynamic client allocation protocol (MADCAP, [101]),
- an intra-domain-server protocol, which is used to claim addresses and to inform local peer MAASs about used multicast addresses: multicast address allocation protocol (AAP, [95]), and
- an inter-domain protocol, used to allocate multicast address sets to domains: multicast address-set claim (MASC, [125], [167]).

2.1.2 Intra-Domain Multicast Routing

Intra-domain routing means routing in a limited sized network. The expectations for such routing protocols are very different from those for inter-domain routing protocols. Typically, the best path between two nodes is defined by numerous criteria including the number of inter-node hops, the most available bandwidth, and the smallest. In inter-domain routing, a best path means the path with the lowest cost.

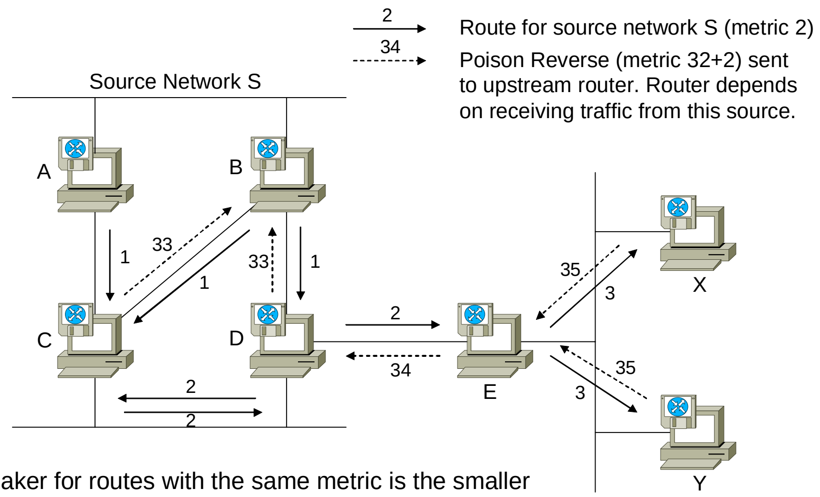
Multicast routing protocols can be divided into two types: the dense-mode protocols and the sparse-mode protocols. The next few subsections introduce the working principles of both classes and the mechanisms of the most widely used protocols. In addition, a short overview of a very different kind of multicast forwarding protocol, XCAST, is given.

2.1.2.1 Dense-Mode Protocols

Dense-mode protocols work according to the push principle: Multicast traffic is forwarded through the network until routers declare that they are not interested in receiving that traffic. This action is called pruning. The process of flooding and pruning repeats periodically. Dense-mode protocols are easy to implement and require only few resources in routing devices. The problem with these protocols is the working principle. Due to the repeating flood of packets which are not required by any receiver, network links may overload or, at least, become unnecessarily stressed.

2.1.2.1.1 Distance Vector Multicast Routing Protocol (DVMRP)

The first dense-mode protocol, which also was the first multicast routing protocol ever, was the distance vector multicast routing protocol (DVMRP, [197]). DVMRP is a distance vector protocol like the unicast routing information protocol (RIP, [104]). DVMRP periodically updates the routing information (every 60 sec.). It also maintains its own multicast routing table beside the unicast routing information. It is a classless routing protocol and, like all distance vector routing protocols, defines a special TTL as the infinity (32 hops). The flood-and-prune mechanism recurs every 2 minutes.


2
Route for source network S (metric 2)
34
Poison Reverse (metric 32+2) sent to upstream router. Router depends on receiving traffic from this source.
Source Network S
A
B
C
D
E
X
Y
1
33
1
33
1
2
2
2
34
35
3
35
3

Tiebreaker for routes with the same metric is the smaller IP address (Router D). In multi-access networks, one router is elected ad the designated forwarder (Metric + smaller IP address) (Router D, E).

Figure 2.4 - DVMRP Truncated Broadcast Tree

Initially, a DVMRP neighbor ship discovery process is started. All the DVMRP routers send probe messages on all DMVRP enabled interfaces. Figure 2.4 shows the so called truncated broadcast tree. This tree is created by exchanging DVMRP route reports between all neighboring routers. Poison reverse messages are sent upstream to help the upstream routers to build their multicast forwarding tables. If more than one router has the same metric for a particular multi-access network, then the one with the smaller IP address is elected as the designated forwarder.

The final pruned state (Figure 2.5) is the result of an initial network-wide flooding and subsequent pruning. The figure shows the truncated broadcast tree as well as the real packet flow for the (S,G) transmission. Already pruned branches of the multicast distribution tree can be reactivated using so called graft messages.

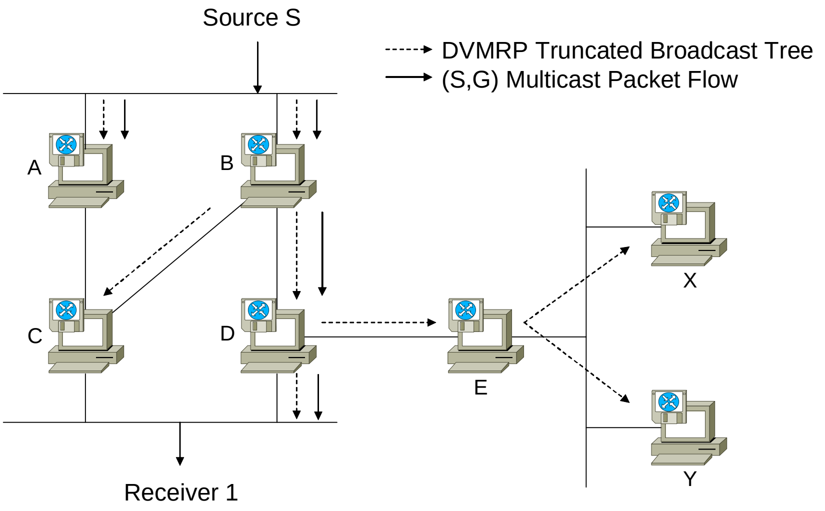


Figure 2.5 - DVMRP final pruned state

The first global multicast network was the MBone [73]. This network was completely based on DVMRP. Because only a few routers have understood multicast routing protocols, a virtual network was created consisting of tunnel connections (IP over IP) between all multicast enabled network clouds.

After a few years, the MBone grew to a very large network leading to severe scalability problems. The design of DVMRP was very simplistic and no one thought of multicast routing tables with up to 50’000 entries. The typical limitations of distance vector protocols apply also to DVMRP. Due to the significant time delay involved when updating 50’000 routes every 60 sec., the network gets severely congested.

In addition, routing tables typically require more than one cycle of route reports (depending on the size of the network) to achieve convergence. This results in potential routing loops or split networks for some minutes. In large networks, routing changes appear to occur very often, which makes DVMRP inapplicable for these installations.

Concerning the quality of service relevant topics of DVMRP, a few disadvantages have to be discussed. Due to the push working principle, network links may unnecessarily become overloaded. Therefore used applications may become inoperable. Another point is the high latency in routing updates. During such updates, little or no functional multicast forwarding is possible.

2.1.2.1.2 PIM Dense-Mode

Another approach is PIM dense-mode (PIM-DM, [57]). PIM is an abbreviation for protocol independent multicast. The term protocol independent means that PIM does not build its own routing table. It reuses the unicast routing table to create the multicast forwarding table. It is a classless routing protocol in the case that a classless unicast routing protocol is in use. The flood-and-prune mechanism repeats every 3 minutes.

Similar to most routing protocols, PIM first creates a neighborhood relationship to other PIM enabled routers. This is done by sending PIM hello messages to the multicast address 224.0.0.13 (All-PIM-Routers) respectively 224.0.0.2 (All-Routers) in PIMv1. This mechanism is also used to elect a designated router (DR), which is responsible for several actions in multi-access networks. If more than one router exists, the highest IP address is used as the tiebreaker. Unlike DVMRP, which builds the minimum spanning tree (DVMRP terminology is truncated broadcast tree) based on its own multicast routing table and the poison reverse mechanism, PIM-DM uses its neighborhood information. An initial shortest path tree (SPT) is built with the input interface toward the source and all other neighbors for destinations. This initial SPT is also known as the broadcast tree.

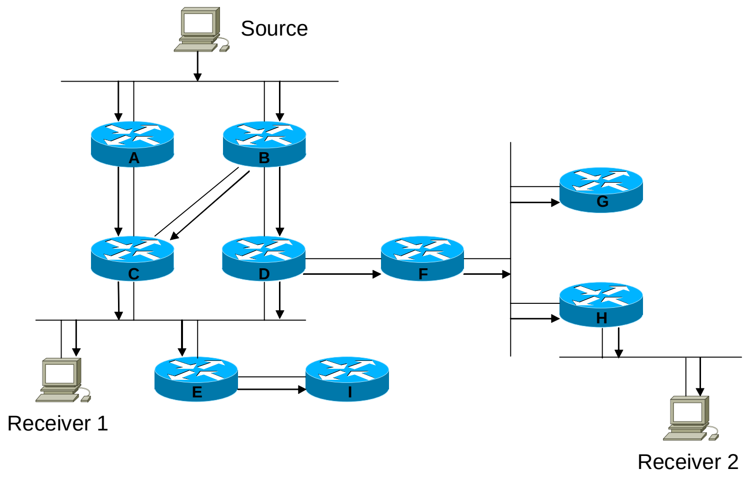


Figure 2.6 - PIM-DM initial flooding

Figure 2.6 shows such a broadcast tree. The source transmits packets into the network and the routers forward them out to all their outgoing interfaces. The problem arises when duplicated packets are created if there is more than one upstream router. An example of this behavior is shown in figure 2.6 at router C. So the tree is cut back gradually.

PIM-DM defines several conditions for this pruning process:

- traffic arrived at a non-RPF interface (router C)
- leaf router without directly connected receivers (router I and G)
- non-leaf router which has received a prune over a point-to-point link (router E)
- non-leaf router which has received a prune over a LAN segment and no other neighbor has overwritten the prune

To prevent an unnecessary pruning in multi-access networks, pruning messages may be overwritten by other attached PIM routers (router H). A special mechanism, named assert, exists for multi-access networks with more than one upstream router. A forwarder is elected by the best metric toward the source using the highest IP address as the tiebreaker (router C and D). Figure 2.7 shows the distribution tree after pruning.

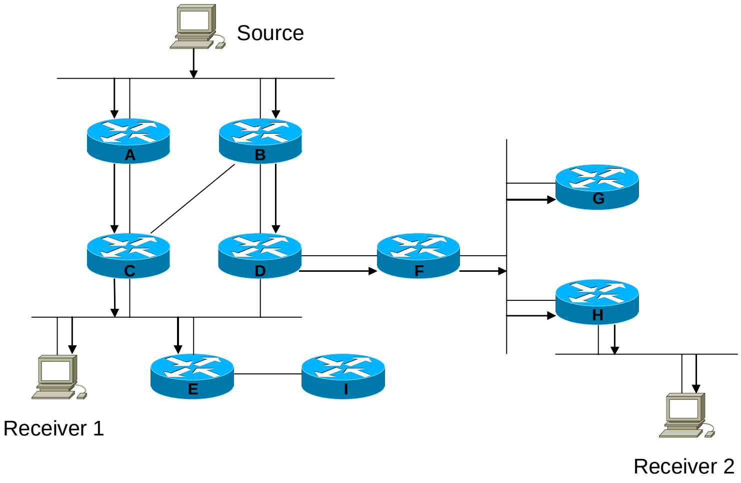


Figure 2.7 - PIM-DM after pruning

PIM dense-mode solves some of the limitations of DVMRP such as the scalability problem by using the unicast routing table instead of its own. Therefore, routing protocols [26] such as OSPF (open shortest path first, [148], [149]) for intra-domain routing or BGP (Border Gateway Protocol, [170]) for inter-domain routing may be used. But PIM-DM still suffers from the flood-and-prune mechanism. Unwanted traffic is flooded through the whole network until all the routers cut back the broadcast tree. Core networks may get overloaded due to this working principle.

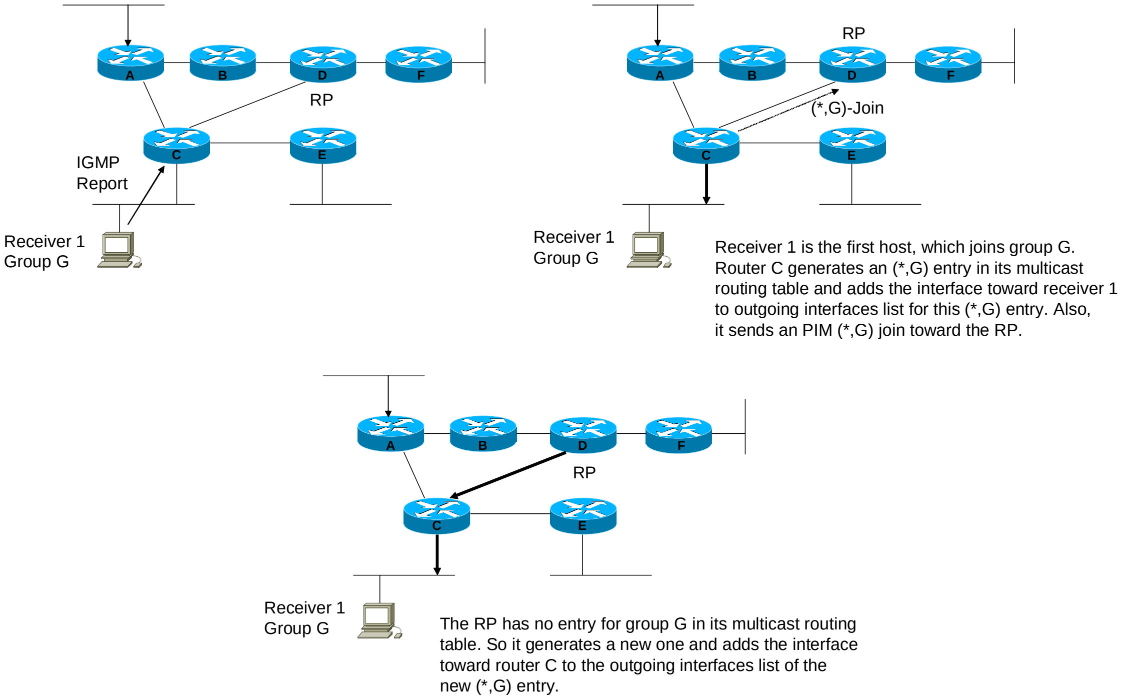
In conclusion, PIM-DM finally shows the same disadvantages concerning the quality of service as DVMRP except for the high latency of network convergence after routing updates depending on the working principles of the configured unicast routing protocol.

2.1.2.2 Sparse-Mode Protocols

In contrast to the dense-mode protocols, sparse-mode protocols follow the pull principle. An explicit join is required in order to receive multicast traffic for a particular group. Typically, sparse-mode protocols implement a central core, which is responsible for maintaining a list of active multicast groups (groups for which either senders or receivers exist) and a list of active senders. In addition to that, the core router is the root of the so called shared trees. Multicast traffic only flows down this tree towards to the receivers. Sometimes the same tree is used to get packets from the sender to the core, sometimes other mechanisms are used. The most common sparse-mode protocols are PIM sparse-mode (PIM-SM, [74], [82], [99]) and CBT (core based tree, [20], [21]). As PIM-SM is the most widely deployed multicast routing protocol today, the basic mechanisms are explained in detail.

2.1.2.2.1 PIM Sparse Mode

Like PIM dense-mode, PIM sparse-mode does not maintain its own multicast routing table, but uses the unicast routing table to build the multicast forwarding table. PIM-SM supports two types of multicast distribution trees. The first one is the shared tree via a central core named RP (rendezvous-point). This tree is also called RP-tree (RPT). The second one is the shortest path tree (SPT), which uses an optimum path (based on the unicast routing table) between the source and the actual router.



Receiver 1 is the first host, which joins group G. Router C generates an (*,G) entry in its multicast routing table and adds the interface toward receiver 1 to outgoing interfaces list for this (*,G) entry. Also, it sends an PIM (*,G) join toward the RP.

The RP has no entry for group G in its multicast routing table. So it generates a new one and adds the interface toward router C to the outgoing interfaces list of the new (*,G) entry.

Figure 2.8 - Shared Tree Join

Figure 2.8 shows the process of building a shared tree toward the RP. First, a receiver appears and signals to its local router that it wants to receive traffic of a particular group G using IGMP.

Next, the router C creates a new entry in its multicast routing table (*,G) and sends a (*,G)-join toward the RP. Finally, the RP receives this join message and creates its own (*,G)-entry. The shared tree for group G is set up and ready to let multicast packets flow down the tree.

The other tree used in PIM sparse-mode is the shortest path tree. As usual, the same mechanisms are used to build the RPT and the SPT. Join messages towards to the upstream router create the tree. Prune messages are used to signal to the upstream router that the traffic is no longer needed. The creation process of a SPT is shown in figure 2.9. The local router E to receiver 1 sends a (S_1,G)-join messages to the sender S_1. Router C creates a new entry for (S_1,G) in its multicast routing table and continues to send a (S_1,G)-join upstream. Router A, which is directly connected to sender S_1, modifies its multicast routing table and starts to transmit packets down the shortest path tree (S_1,G).

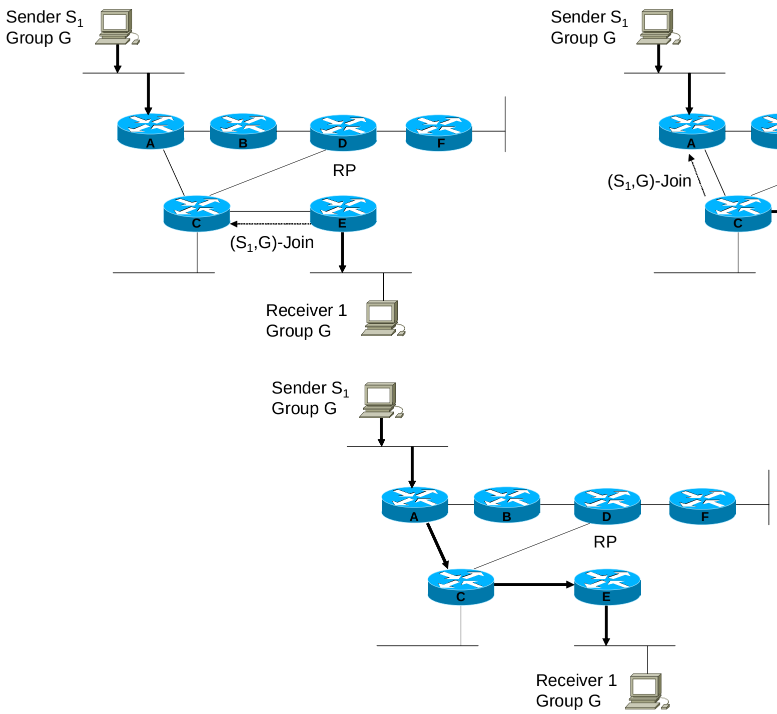


Figure 2.9 - Shortest Path Tree Join

The advantages of SPTs are:

- the usage of a direct (optimum) path between source and destination,
- the minimization of the latency, and
- the minimization of the load of the RP.

Disadvantages are:

- the number of required (S,G) entries (entries in the multicast forwarding table) may become very large, and
- it requires much more resources in the network.

In view of so many advantages, the question "What is the need of the RPT?" comes to mind. The problem is to find active senders in the network. Due to that knowledge, the SPT should always be used. PIM-SM uses the RPT to get multicast packets to all receivers, also informing the last hop router of the existence of new senders. A special process called source registration is used to inform the RP of the existence of new active senders as well as to get the first few packets of the multicast transmission to the RP. This process is shown in figure 2.10. After receiving the register message, the RP creates an SPT toward the sender and informs the first hop router A to stop the register messages. At this time, Router C receives traffic from sender S_1 and can initiate the SPT switch-over process by establishing a SPT toward the sender and pruning the RPT toward the RP for sender S_1. The register messages, which are unicast from the first hop router to the RP, also contain the multicast traffic from the sender.

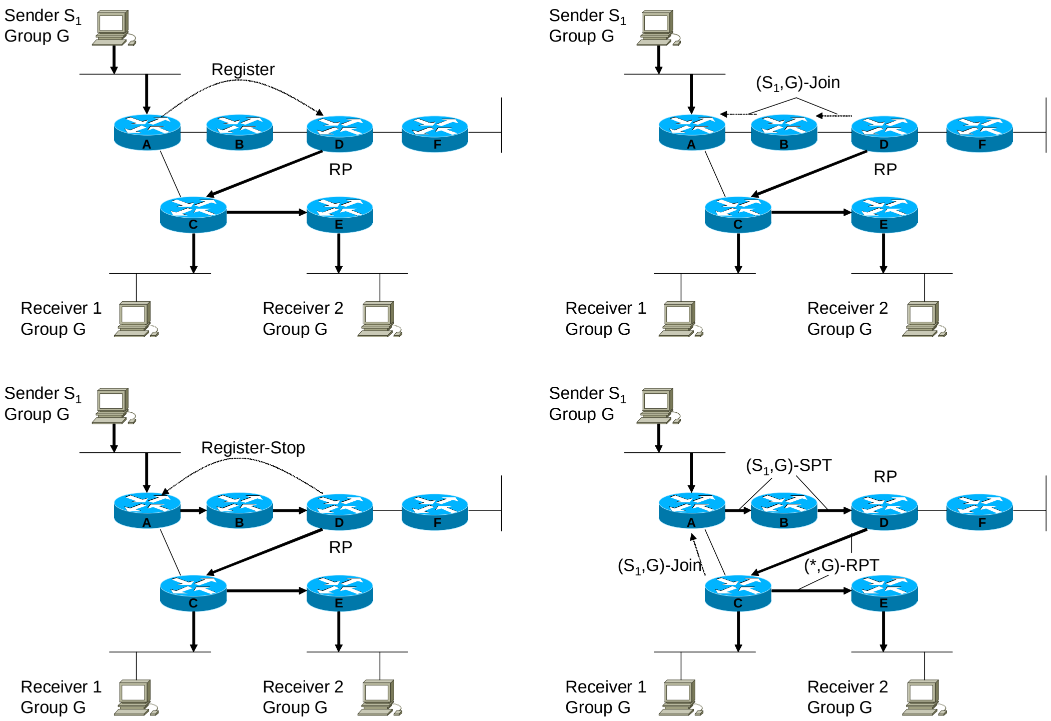


Figure 2.10 - Source Registration and SPT switch-over

The last property of PIM-SM, to mention here is the discovery of the RP. The RP can be configured statically, but there are also two mechanisms to locate it dynamically to provide some more redundancy. Cisco developed a mechanism called Auto-RP and in PIMv2 the bootstrap router (BSR, [81]) mechanism has been introduced. The latter is the current standard for PIM sparse-mode networks. Further concepts for the localization and the relocation of the RP have been developed [17].

Due to the working principle of PIM-SM, an overloading of the network caused by unwanted multicast traffic cannot occur. Therefore QoS related questions on PIM-SM are limited to the join latency. Depending on the size of the network and the location of the RP, it may take some time to create the state for a group in all involved routers. Additionally, the RP may get overloaded if there are too many high bandwidth multicast streams using the RP at a given moment. Finally, the SPT switch-over process may take another few seconds. The join latency is a very important parameter in sparse-mode networks and should be examined carefully by network management tools and network wide measurements.

2.1.2.2.2 Source Specific Multicast (SSM)

To improve some of the drawbacks in PIM sparse-mode such as the requirement of a functioning RP or the high join latency, a new protocol is currently being standardized. Source specific multicast (SSM, [111], [24], [25]) is a simplified version of PIM-SM. Only shortest path trees are used and the task of finding active sources has been moved to mechanisms other than the multicast routing protocol. By using SSM, the join latency can be dramatically reduced to the time it takes to build the SPT from the last hop router toward the sender. Furthermore, the security of multicast transmissions is improved because it becomes much more difficult to start denial of service (DoS) attacks to an active multicast application just by sending jam to the same multicast address. Only packets from explicitly joined sources are transmitted through the network.

2.1.2.3 Explicit Multicast (Xcast)

Xcast [28] is not really a multicast routing protocol but it enhances the standard IP routing by multicast features, e.g. it defines the possibility for IP packets to have more than one destination address. The list of destination addresses is examined by each router along the paths. They will duplicate the packet if required. The address list is modified according to the receivers along the new path. Using this mechanism, Xcast does not have to deal with any kind of multicast routing protocols or with choosing a multicast address. Instead, separate mechanisms have to be installed to supply information about the session membership to each participant. This implies that the worst possible join latency cannot be predicted. It depends on the technique used to synchronize the session information to all clients.

Even if IP multicast routing is not available in all the parts of the internet, it is the preferred technology for one-to-many connections. Xcast has no significance in the global internet routing.

2.1.3 Inter-Domain Multicast Routing

For the inter-domain multicast routing, different protocols are used [54]. Today, most networks are running PIM sparse-mode or DVMRP which have to be interconnected. Unicast routing protocols such as the border gateway protocol (BGP-4, [170]) allow the implementation of metrics based on real costs of links instead of on the available bandwidth or the number of hops.

Routes are exchanged between different networks, which are called autonomous systems (AS). MBGP (multiprotocol extensions for BGP-4, [23]) does the same for multicast. Since the transport networks, typically, also run PIM-SM for multicast routing, it becomes very easy to interconnect multiple sites. Only one problem needs to be solved. There is no global RP to inform all the attached routers of active sources. A new protocol has been initiated to fill this gap: the multicast source discovery protocol (MSDP, [143]). Using MSDP, the RPs of different networks inform each other about active sources. Finally, the PIM-SM mechanisms to build a shortest path tree can be used to build the multicast distribution trees [56]. Because such deployments are not so easy to install and to maintain, special instructions are given on how this task can be accomplished [137].

Regarding the quality of service in the multicast network, there are two main parameters to review in inter-domain multicast routing. The first and most important one is the availability because there are a large number of possible misconfigurations. Secondly, due to the mechanism of MSDP, it takes a relatively long time to inform all routers about the existence of new sources. This results in a very high join latency, which has to be examined in order to complete description of the quality of service.

2.2 Analysis of well-known Multicast Networks

Having examined the working principles and the quality of service-related properties of multicast routing protocols, the following section provides an analysis of the well-known contemporary multicast networks. This section concentrates on the practical deployment of the routing protocols and examines four different multicast networks and their infrastructure:

- the global multicast routing in the internet,
- the German research network (G-WiN, "Gigabit Wissenschaftsnetz"),
- the Bavarian university network (BHN, "Bayrisches Hochschulnetz"), and
- the campus network of the University of Erlangen-Nuremberg.

A short historical overview of the IP multicast routing evolution is provided including a discussion of several deployment issues which are still relevant today.

2.2.1 Global Multicast Routing in the Internet

Before the construction of the MBone (multicast backbone) in the early nineties, only a few very local deployments of multicast routing existed. The MBone is a virtual network overlaying the internet which interconnects all these small multicast enabled clouds. This interconnectivity has been achieved using tunnel structures and an IP in IP encapsulation. The basic infrastructure is shown in figure 2.11. DVMRP has been used as the IP multicast routing protocol for the MBone.

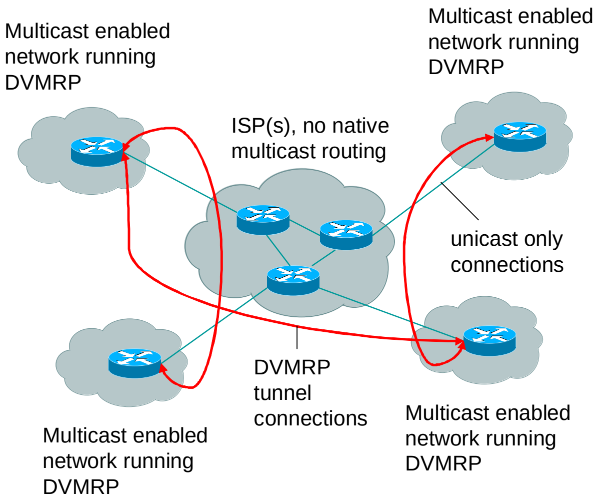

Multicast enabled network running DVMRP
Multicast enabled network running DVMRP
ISP(s), no native multicast routing
unicast only connections
DVMRP tunnel connections
Multicast enabled network running DVMRP
Multicast enabled network running DVMRP

Figure 2.11 - The MBone

Unfortunately, there are a number of limitations in this installation, which are based either on the routing protocol DVMRP or on the tunnel structure. As the MBone matured, the number of entries in the multicast routing table increased as well. DVMRP updates its routing table periodically. Not only did the process of updating 50’000 routes every 60 sec. become problematic, but also, the distance vector protocols required a special hop count to define infinity. In DVMRP this value has been defined to 32. Therefore, no paths longer than 31 hops may exist in a DVMRP network.

Limited by the available bandwidth, the push working principle of DVMRP became a serious bottleneck as well. With the growing number of active applications, the unnecessary traffic grew as well. Finally, the analysis and the solution of routing problems have become more and more difficult in the growing network.

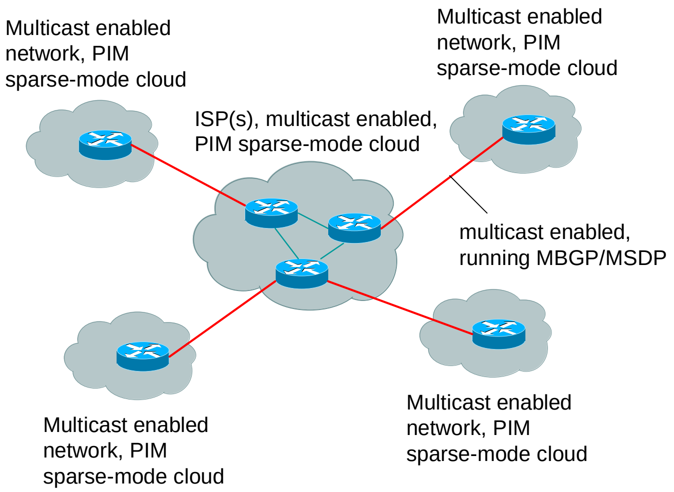

Multicast enabled network, PIM sparse-mode cloud
Multicast enabled network, PIM sparse-mode cloud
ISP(s), multicast enabled, PIM sparse-mode cloud
multicast enabled, running MBGP/MSDP
Multicast enabled network, PIM sparse-mode cloud
Multicast enabled network, PIM sparse-mode cloud

Figure 2.12 - Global Multicast Routing

Today, the old MBone has been replaced by a natively integrated service [8]. Most carrier networks as well as most of the research networks around the world have already implemented a native multicast routing using PIM sparse-mode. The interconnectivity between these single networks can be assured using MBGP and MSDP. An example of the protocol usage in the current global multicast routing is shown in figure 2.12.

2.2.2 German Research Network

The primary task for the German research network (G-WiN, "Deutsches Wissenschaftsnetz") is to provide high speed and high quality interconnectivity between all universities and other R&D institutes. Currently, the network is in its 3rd generation, using backbone links up to 10 Gbps. Some universities including the University of Erlangen-Nuremberg are connected at OC-12 speed (622 Mbps). Meanwhile, the native routing of IP multicast has been implemented all over this network. The G-WiN, which includes most of the networks of the connected institutions, behaves as a large PIM-SM network cloud. A few RPs have been installed to provide some kind of load sharing and a higher reliability. The multicast routing to the global internet is done by using MBGP in conjunction with MSDP.

As to the quality of service in this large multicast domain, at least the large join latency has to be mentioned. Due to the size of the network (about 50 routers in the backbone and countless routers at each site) it takes up to several seconds from joining a multicast group until the first packets are delivered from an active sender. Other parameters, such as the delay of a transmission, are analyzed in a following section.

2.2.3 Bavarian University Network

The Bavarian university network (BHN, "Bayrisches Hochschulnetz") has been initiated to interconnect all Bavarian universities. This association has been used to carry on negotiations with internet service providers (ISPs) or software resellers. In this context, the multicast connectivity in the BHN is discussed in more detail. Starting in 1994, a virtual network has been created consisting of tunnel connections between single multicast networks of various universities in Bavaria (figure 2.13). This network has been administrated by researchers of the University of Erlangen-Nuremberg.

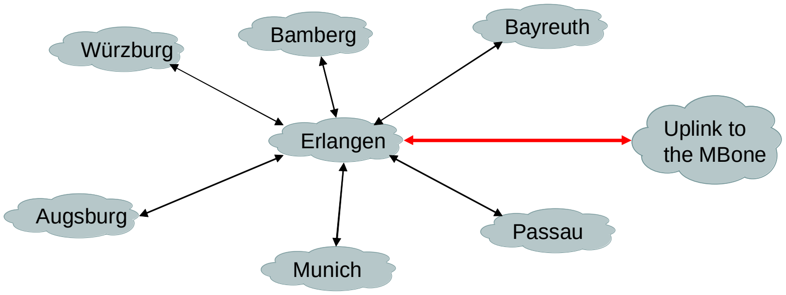


Figure 2.13 - Structure of the Virtual Multicast Network in Bavaria

In the year 2000, this virtual network was replaced by a native multicast routing provided by the German research network. The Bavarian multicast backbone has not survived as a single multicast cloud but it allowed to learn a lot about IP multicast routing in its first version. After the reorganization, the Bavarian university network can still be used for network measurements and tests of network applications.

2.2.4 Campus Network of the University of Erlangen-Nuremberg

Around 1992, the University of Erlangen-Nuremberg started to implement IP multicast in the campus network using the first DVMRP based solution. Today, native IP multicast routing using PIM sparse-mode is available over the entire campus. Especially for multimedia applications, the usage of multicast is being promoted [68].

The backbone network consists of a pure layer-3 IP network built of Cisco routers connected by gigabit ethernet networks. Attached to this backbone, individual routers support geographically or logically separated parts of the university. This structure is shown in figure 2.14. The four routers on the top build the backbone network. It is structured for a primary and a secondary (backup) usage. The routers in the middle of the figure are responsible to connect the single regions to the backbone. On the left side, the connection to the German research network is represented.

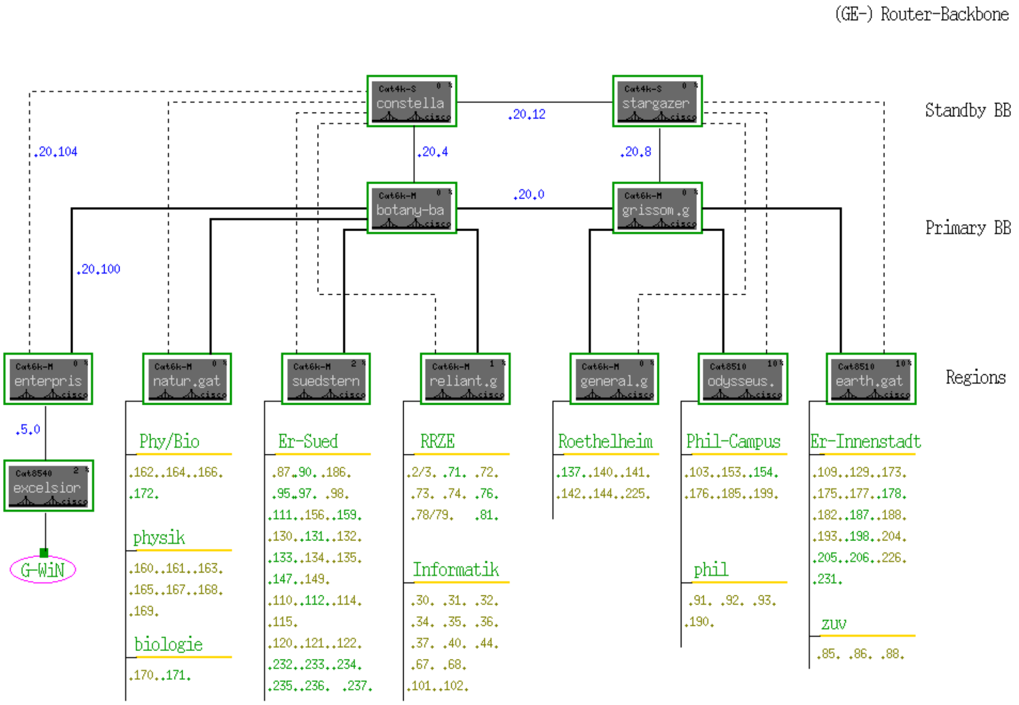


Figure 2.14 - Campus Network of the University of Erlangen-Nuremberg

The complete campus network builds a single PIM-SM cloud including its own PIM infrastructure. The RP is located on the border gateway (excelsior) towards to the German research network using MSDP to announce active senders to the MSDP peers in the G-WiN. The existence of the RP is distributed dynamically over the campus network using auto-RP. If the RP fails, the complete network will be switched to PIM dense-mode to continue the forwarding of multicast traffic.

The university operates its own RP for the following reasons:

- the maximum number of hops from a multicast sender or receiver is minimized,
- the reliability of the multicast network is improved,
- potential problems can be analyzed and repaired by local network administrators, and
- it becomes possible to deploy a test bed for new multicast applications and protocols, and to connect it easily to the global multicast routing.

The university network has been used for the developing and testing of various multicast hardware and software with the focus on quality of service measurements. A typical example is the development of the multicast quality monitor (MQM) shown in section 7.

2.3 Analysis of typical Multicast Services

In order to allow statements on the capabilities of an IP multicast network in general, the used applications have to be analyzed. In the context of this work, the applications are called services to distinguish between the tools and the application scenarios.

This subsection is split into two parts. The first part examines typical services using one-to-many transmissions, which are called broadcasts, and the second part deals with many-to-many connections. Such services mostly are conferences and distributed applications.

2.3.1 Broadcasts

Strictly speaking, the one-to-many transmission model is a special case of the many-to-many operation of IP multicast. This distinction is done because there is a number of such broadcasts active in the internet and because of the different resource usages compared with real many-to-many transmissions.

Figure 2.15 shows this scenario including a comparison of the network and server usage if multicast is preferred over single unicast transmissions. In the case of multicast, numerous resources are saved. Thus, the utilization of the server and of its network connection is minimized. Additionally, the load of single network links within the core network is reduced.

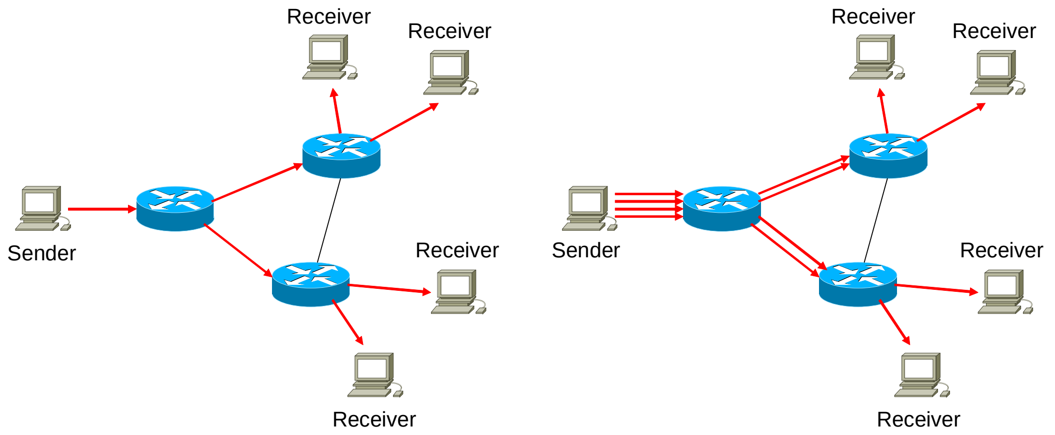


Figure 2.15 - One-to-many Transmission (left: multicast, right: unicast)

The most common kind of broadcast is a video or TV broadcast. Typically, these transmissions have a high bandwidth usage in a single direction. The sender streams the (high quality) video into the multicast network and all other participants do only receive this traffic. The information about the quality of the received data is provided back to the sender in order to allow adaptive algorithms to correct the used bit rate according to the current transmission quality over the multicast network.

Regarding the required quality of service, only the paths (the multicast tree) from the sender towards to all the receivers have to be analyzed. The backchannel usage is nearly zero.

2.3.1.1 FAU-TV

One example of such a TV broadcast is a service installed at the University of Erlangen-Nuremberg, which is named FAU-TV. This service has been initiated to allow an easy testing of the multicast infrastructure in Germany. A single TV channel is being streamed into the network at about 800 kbps (MPEG1, 10 fps). The programming of the infrastructure of this streaming server has been done in several students' projects. Due to the behavior of FAU-TV, which endlessly sends multicast packets, this service has also been used for several tests of the current quality of service of the used parts of the multicast network.

2.3.1.2 Uni-TV

Another project at the University of Erlangen-Nuremberg is Uni-TV [62]. The goal of this project is to record and produce lectures at a high quality. A German TV sender is involved to increase the quality of the production and broadcasts the recorded lectures. Uni-TV has created its own video server to offer all recordings to the students. One distribution path is made via a self-made near-VoD (Video-on-Demand) service. The students are able to select the program using a web-based front-end. The video server offers some multicast channels to stream recorded lectures into the internet. If a channel is available, the streaming can be started instantly. If not, the streaming is scheduled at a later time.

The behavior of these multicast transmissions differs from the FAU-TV example. There is not only a single channel sending videos, and the streaming occurs at random times, based on the requirements of the listeners. In order to achieve this, the active transmission cannot be used in general to measure the quality of the used multicast network. Instead, other mechanisms have to be used.

2.3.2 Conferences and distributed Applications

The concept of multicasting was to allow distributed applications. One of the first and most popular examples is an audio or video conference [39].

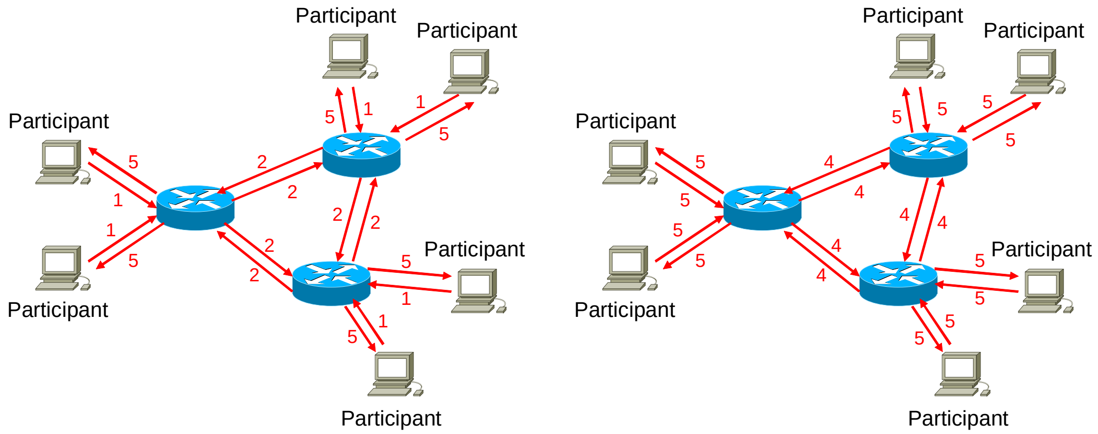


Figure 2.16 - Many-to-many Transmission (left: multicast, right: unicast)

The use of multicast for such services provides the possibility to dramatically reduce the resource requirements for each participant as well as for the used network. Figure 2.16 shows a typical service based on the many-to-many transmission model. The numbers above the connections describe the utilization of the particular links.

But not only audio or video conferences make use of the capabilities of IP multicast. More and more distributed applications appear to use multicast such as a distributed file system [89]. Some examples of multimedia services are given in the following subsections.

2.3.2.1 TKBRZL

The idea of the project TKBRZL [107] was to allow the directors of the computing centers in Bavaria to organize meetings without having to travel around. Using the capabilities of IP multicast, a video conference was set up including shared applications like a text editor to protocol the session. Since there have always been about 10 to 12 participants, the resource requirements have been very high. Every client had to receive and analyze 9 to 11 videos. Additionally, the network has to be capable of transmitting all the multicast streams with only a few, or no lost packets at a minimum delay to retain the interactivity.

Separate projects have been started to analyze the behavior of such a conference. It has been proved that such a conference requires significant support by technical engineers. First, the availability and, importantly, the current available quality of the used parts of the multicast network have to be verified before the conference gets started. Secondly, during the session, the participants often require help in using the available tools.

2.3.2.2 MBone-de

One of the first virtual meetings using IP multicast was the MBone-de session. This virtual conference room has been set up by the University of Erlangen-Nuremberg to allow different people to discuss current problems.

Mostly, the MBone-de session has been used by administrators of multicast networks. Additionally, this session is often used to test the connectivity between different multicast clouds. The problem with this kind of test is the requirement of having participants online at each site. These users also have to be synchronized using E-mail or telephone calls to locate potential multicast routing problems.

2.3.2.3 Network Time Protocol (NTP)

One of the first distributed applications using IP multicast has been the network time protocol (NTP, [146]). NTP is used to synchronize the clocks of different computers to a single source. Typically, this source is synchronized by high precise time synchronization mechanisms such as GPS (global positioning system). First, NTP uses only unicast connections for this synchronization. This requires the knowledge about available time sources and their IP addresses. In local networks, the synchronization can be done using broadcast messages as well. To minimize the administrative cost, IP multicast can be used instead. Each client joins the well-known multicast group to receive the synchronization messages. Because NTP is very tolerable to network problems such as lost packets, only the connectivity is important for this kind of service and has to be ensured by the network administrators.

2.4 Analysis of well-known Applications

The first applications using IP multicast have been multimedia conferencing tools. The development of audio and video applications has been the most interesting research topic over the last couple of years. Beside these tools, collaboration and session management appliances have been created.

2.4.1 Audio

The visual audio tool (VAT, [113]) was the first available tool to establish audio conferences. Developed in 1992 at the Lawrence Berkeley Laboratory (LBL), it allowed the users to start a audio transmission in different qualities. The quality of a conference depends on the used bandwidth and the selected audio codec. In 1995, a new tool for audio transmissions appeared: the robust audio tool (RAT, [102], [103]) developed at University College London (UCL). This application shows a much more robust behavior in the case of a low transmission quality. Adaptive algorithms allow for adjustment of the transmission rate in order to achieve the best possible quality. Additionally, the RAT includes a series of codecs which allow audio transmissions in a wide range of qualities and resulting bit rates from CD quality (Stereo, 192 kbps, PCM G.711 encoded [217]) down to a still understandable speech transmission at about 16 kbps (Mono, G.726 encoded [212]).

2.4.2 Video

For conferencing, the most popular tool is the video conference tool (VIC, [138]) developed at the Lawrence Berkeley Laboratory (LBL). Like the audio applications, it includes a number of different codecs to encode a video in different qualities and at different bit rates. For low bandwidth environments, there exist a number of available codes such as H.261 [218] and H.263 [219]. Using MPEG1 [214] at about 2 Mbps, it is possible to have a video connection using IP multicast at TV quality. In contrast to the audio streaming, the video transmission depends much more on the available quality of service of the network (see also section 4.1).

Another application using IP multicast for video transmissions is the video server from Cisco, Inc. named IP/TV [213]. This server has been designed to offer a near-VoD program based on schedules provided by the administrator or requests by a remote user. Additionally, life transmissions can be started. The quality starts at MPEG1 at a transmission rate of less than 1 Mbps (low TV quality) and increases up to MPEG2 [215] at more than 6 Mbps (DVD quality). Using the Cisco IP/TV server, it is possible to initiate high quality video transmissions if the used multicast network provides a high quality of service.

Beside these particular tools, approaches have been developed allowing a scalable feeback control [29].

2.4.3 Collaboration and Session Management

Last but not least, a set of tools exist to allow a proper session management and for collaboration beside a video conference. The session directory tool (SDR, [93]) has been developed to allow a global session management including the announcement of active and forthcoming sessions. Additionally, it provides information about the sessions such as the used multicast addresses, the used codecs or information about the originator of a particular session. The SDR is used by nearly every multimedia application to choose a free (unused) multicast group address in order to prevent failures due to address conflicts.

The whiteboard tool (WB, [114]) was the first approach to offer a shared notepad to the participants of a conference. This tool allows each user to draw or write on a virtual sheet of paper at the same time. The result can be seen at each site with only a short delay. The drawback of the tool is its inability to handle text in a feasible fashion.

To advance this tool for environments which requires a text-only tool with the capability to allow every participant to modify the written text independently, the development of the network text editor (NTE, [94]) has been initiated at University College London (UCL). The NTE works like a simple text editor but allows a shared usage over an IP multicast enabled network. This tool is commonly used to protocol a virtual meeting using video conference applications.

3 Quality of Service in Multicast Networks

This section provides an overview of the quality of service (QoS) parameters and measurement methods in conjunction with IP and IP multicast networks. Definitions of terms such as real-time and the different QoS values are given in the first few subsections Application requirements and measurement methodologies are discussed in section 4 and section 7.3. Next, the real-time transport protocol (RTP) is examined. This protocol is used for most multimedia transmissions over the internet especially if IP multicast is being used [128].

Measurement tools are used to provide information about the current quality of service or even to predict the QoS for upcoming multimedia sessions. The IPPM WG (IP performance measurement working group) of the IETF (internet engineering task force) defined some concepts outlining how to measure different QoS parameters which are explained in few words. The main part of this section concentrates on the examination of existing measurement tools for IP multicast networks. Finally, an overview describing the approaches used to enhance or even to guarantee quality of service values for single applications is provided.

3.1 Real-Time and QoS in Communication Networks

Mercer [140] defines a real-time system as the entire collection of components that are designed to solve some physical world problems; the system includes the physical environment in which this problem is to be solved, the application software, the operating system software and the underlying hardware. Figure 3.1 shows a schematic of a time constrained computation. Each computation has a ready time at which the computation becomes available for scheduling. At some point after the ready time, the computation will be scheduled and will start processing. The computation will terminate at a later time. A deadline is typically associated with the computation as well, and the aim is to complete the computation before the deadline.

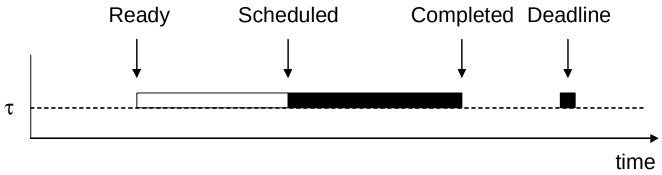


Figure 3.1 - Schematic of a Time Constrained Computation [140]

It is important to distinguish between the following definitions of deadlines [140]:

- A hard deadline means a deadline after which the value of completing a task becomes 0 or becomes -∝; The task can be ignored (if the value becomes 0) or the system experiences a catastrophic failure (if the value becomes -∝).
- A soft deadline characterizes a deadline after which some value remains in completing the associated task.

Steinmetz [191] defines real-time as follows: real-time operation of a system means that programs to compute available data have to be operational so that the results of these computations are ready within a defined time slice.

Another definition of real-time systems can be found on http://www.faqs.org. Here, a real-time system is one in which the correctness of the computations not only depends upon the logical correctness of the computation, but also upon the time at which the result is produced. If the timing constraints of the system are not met, a system failure has occurred.

Even if a single, all-embracing definition for real-time systems cannot be found, typical properties of real-time systems can be summarized as follows:

- Real-time systems require predictable fast processing of time-critical events. All contingencies have to be considered.
- A high degree of accurate scheduling has to be performed. This schedulability defines the maximum utilization of a resource if time limits are obligatory.
- Real-time systems should show a stable system behavior in case of overload. So it has to be guaranteed that deadlines of time critical processes are met even if the system is overloading.

In classical real-time applications a calculation has to be finished at a given time. If the result is not available at this time it is worthless. If the transmission of data over a computer network has to be examined, the definition of real-time changes slightly. Real-time network applications not only require a guaranteed maximum transmission delay, but also, and more importantly, a very low variation of the delay. This is especially true in the case of multimedia applications. Additionally, the error rate has to be nearly zero and the availability should be at 100%. In addition, a predictable behavior of the clients is required [84].

Recapitulating, it can be said that the network has to be able to guarantee a number of requirements of real-time applications. The term quality of service (QoS) has been introduced to describe the requirements of the applications as well as the provisions of the network. The individual QoS parameters are characterized in the following subsection.

3.2 Quality of Service Parameters

In order to analyze the quality of service of a particular multicast network, the single QoS parameters have to be defined and explained. In order to prevent the objective from being lost, only those parameters are described, which are imperative in multicast and multimedia environments. The first QoS parameter is the reachability. Particularly in the case of complex multicast networks, full connectivity between all interconnected sites cannot be presumed. Real-time multimedia applications depend predominantly on the delay of a transmission. Above all, the one-way delay is important in multimedia environments. In case of a bidirectional communication the round-trip-time has a large impact as well. Streaming audio and video transmissions require a low variation of the delay, and nearly every application depends on a low packet loss ratio. In every transfer using IP, especially in IP multicast environments, reordered and duplicated packets may appear. The applications are required to handle such events, but it is desirable to prevent such failures.

3.2.1 Reachability and Reliability

Connectivity between two end systems means that it is possible to transmit data between these two sites with at least a best effort behavior. Typical reliability measurements are based on this simple metric [130]. Every QoS test should start with the examination of the connectivity. The principle of the reachability between several clients is shown in figure 3.2. In this example, the connectivity test between Host A and C and between B and C is positive. No reachability between Host A and B is provided.

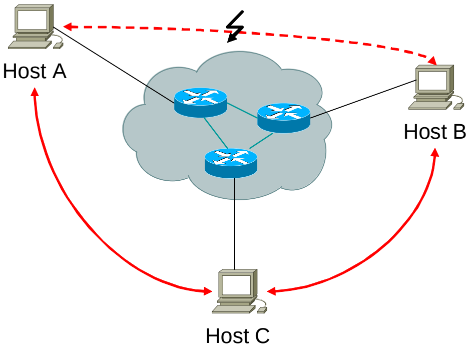


Figure 3.2 - Reachability between several Clients

Normally, the reachability should be maintained by the simplest possible mechanisms. Redundancy is provided in typical backbone networks. Even if the provisioning of the connectivity sounds easy, especially in IP multicast environments this cannot be presumed. The multicast reachability suffers from the complexity of the multicast routing protocols and the lack of experience of network administrators declining with these mechanisms. Another problem is the still miserable interoperability between devices of different manufacturers and, partially, the incomplete implementation of the protocol stacks.

Using the results of reachability measurements over a period of time, the reliability of the network can be calculated. High availability systems require a reliability of nearly 100%.

Therefore, reachability means connectivity at a certain point of time and reliability stands for the percentile reachability over a period of time.

3.2.2 Delay

The delay, or more precisely the absolute delay, describes the latency between the transmission of a packet and its successful reception at the receiving site. In a broadcast scenario, a large but constant delay leads to a delayed playback of multimedia content, but does not reduce the overall quality of the transmission. For bidirectional conferences, for example, it has been shown that a delay larger than 200 ms reduces the interactivity dramatically [36].

Measuring the delay requires the distinction between two possible results: the one-way delay and the round-trip time. Figure 3.3 shows this principle. Both delay measurements are described in the following subsections in more detail.

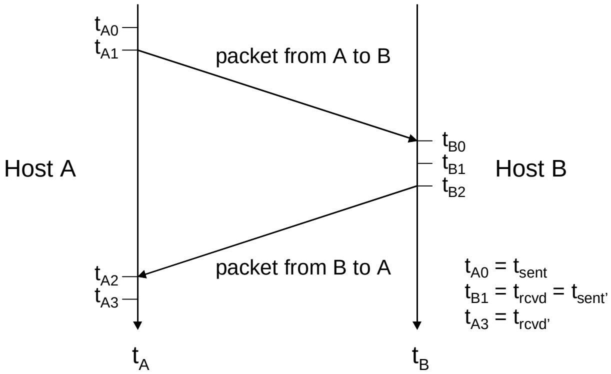


Figure 3.3 - Delay between two Hosts

Additionally, the behavior of the system has to be examined in order to get correct results. For example, it takes a small amount of time between sending a packet at the program level and the transmission of this packet out of the network interface of this machine. The reason for this additional delay is the working design of the computer. The computer moves the packet first through different layers of the operating system software until it reaches the hardware interface. Some layers include a queuing mechanism to prevent locks in the application. Unfortunately, each queue introduces an additional queuing delay.

3.2.2.1 One-Way Delay (OWD)

One of the most important values for real-time multimedia communication is the one-way delay (OWD), because every transmission of audio or video signals flows unidirectional from one host to another. Even in bidirectional video conferences, the one-way delay is very important. Due to synchronization problems between the clocks of each client, the measurement of the one-way delay is a non-trivial task. For exact measurements, it is required that both clocks are highly synchronized.

An example is given in figure 3.3. Each host has its own independent time line (t_A and t_B). At time $t_{A0} = t_{sent}$, the application decides to send a packet to host B. A short delay is introduced by the application, the operating system, and the networking hardware, so that the packet is actually sent at t_{A1}. Host B receives the packet at t_{B0} and a timestamp is taken by the application at t_{B1} = t_{rcvd} (there is an additional delay by the system between t_{B0} and t_{B1}). If the timestamp t_{A0} has been included in the packet, it is possible to compute the one-way delay (Δt_{OWD}) as follows:

$$\Delta t_{OWD} = t_{rcvd} - t_{sent} \qquad \text{Equation 3.1}$$

The problem of synchronizing hosts interconnected by computer networks is an intensively discussed research topic. Interesting ideas and approaches can be found, for example, in papers from Mills [146] and Awerbuch [18].

3.2.2.2 Round-Trip Time (RTT)

Especially for applications which require a fast query-response behavior, the round-trip time (RTT) is an important value. It is also known as the bidirectional or two-way delay. Examples of such applications are video conferences and remote controls.

The example in figure 3.3 also shows the principles of the measurement of the round-trip time. The timestamps $t_{A0} = t_{sent}$, t_{A1}, t_{B0}, and $t_{B1} = t_{rcvd}$ are the same as those used by the one-way delay measurement. t_{B1} is also used as $t_{sent'}$ included in the response packet. At t_{A2} the packet is received by host A and the last timestamp is taken at $t_{A3} = t_{rcvd'}$. Using all these timestamps, it is possible to compute the round-trip time (Δt_{RTT}) as follows:

$$\Delta t_{RTT} = \Delta t_{OWD'} + \Delta t_{OWD} = t_{rcvd'} - t_{sent} \qquad \text{Equation 3.2}$$

Because the measurement of the round-trip time depends only on $t_{rcvd'}$ and t_{sent}, which are timestamps at host A, only the clock of one host is involved. Therefore, no synchronization between the clocks of each host is required. This is the reason, why the estimation of the RTT is one of the standard measurements in computer networks. Additionally, the behavior of the network in terms of the available quality of service can be rated using this single parameter.

3.2.3 Jitter (Delay Variation)

The variation of the inter-arrival time of packets at the receiving site is known as the delay variation, also referred to as the jitter [59]. Based on the mechanisms used to measure the jitter, its definition can vary slightly. There are two basic approaches to measure and to define the jitter.

The variation of the delay. To estimate the jitter, delay measurements are taken over a period of time. The jitter is computed as the maximum variance of the delay around its mean value. Additionally, the 90th percentile or the 95th percentile can be calculated by ignoring the top 10% or 5% of the highest delays respectively. The concept behind such percentiles is to remove single, exceptionally high delay values.

Typically, the variation of the delay is computed using the OWD measurements, because the jitter in the round-trip time is ostensibly meaningless for multimedia transmissions. Unfortunately, the same problem appears as in the calculation of the one-way delay. The clocks of all involved systems have to be synchronized.

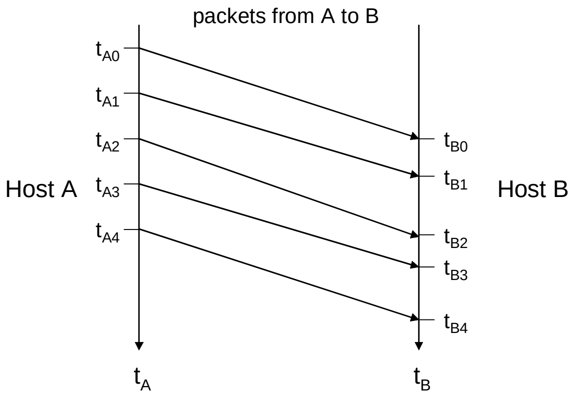


Figure 3.4 - Jitter (Delay Variation)

An example is shown in figure 3.4. Host A sends periodically packets to host B. It includes a transmission timestamp in each packet in order to allow the estimation of the OWD at host B. The variation of the delay can be calculated as:

$$\Delta t_{\mathrm{jitter}} = \max_{k}\left(\left|\frac{\sum_{i=0}^{n} \Delta t_{\mathrm{OWDi}}}{n+1} - \Delta t_{\mathrm{OWDk}}\right|\right) \qquad \text{Equation 3.3}$$

The variation of the interarrival time. Another method is to analyze the variation of the interarrival time of packets. A constant flow of packets with a well-defined inter-packet distance is required for this estimation. The receiver measures the distances between the packets when it receives them. The jitter can be calculated as the maximum variance (90th percentile, 95th percentile) of the interarrival time and its mean value over a period of time.

$$\Delta t_{\mathrm{interarrival\ n}} = t_{\mathrm{Bn}} - t_{\mathrm{Bn\text{-}1}} \qquad \text{Equation 3.4}$$

$$\Delta t_{\mathrm{jitter}} = \max_{k}\left(\left|\frac{\sum_{i=1}^{n} \Delta t_{\mathrm{interarrival\ i}}}{n} - \Delta t_{\mathrm{interarrival\ k}}\right|\right) \qquad \text{Equation 3.5}$$

Using this kind of definition of the jitter, the problem of unsynchronized clocks between the involved systems disappears because only timestamps at the receiver are used for this computation. This method is also used for the proposed MQM (section 7.3.3).

It is typical of Multimedia applications to use RTP (real-time transport protocol, [182]) to exchange data between each participating site. This protocol allows the receiver to measure the current jitter based on timestamps written in each packet. The application is required to introduce a playback buffer to deal with the actual delay variation. If a packet arrives after a maximum time (defined by the size of the buffer), it is too late. Therefore, the application has to drop it. Consequently, it is possible to reduce the problems introduced by a large jitter to them given by a constant delay (jitter smaller than allowed by the size of the playback buffer) or those introduced by a non-zero packet loss ratio (if packets are dropped). Nevertheless, the jitter is a parameter which should be carefully measured in order to predict the behavior of an application depending on the current quality of service of the network.

3.2.4 Packet Loss Ratio

The packet loss ratio defines the amount of packets which were lost in the last time slice [130]. The size of the mentioned time slice depends on the specific measurement method.

Packet loss is very common in the internet. The IP protocol instructs devices on the path to drop packets if they discover any problems. One example is the lack of resources on oversubscribed interfaces. Therefore, most transmissions over longer paths through the network show a remarkable packet loss ratio.

Even if different encoding algorithms for multimedia content are available which allow the applications to deal with a small packet loss ratio, the knowledge about the packet loss ratio is useful for several reasons [13]:

- Some applications do not perform well if the end-to-end packet loss between hosts is high in relationship to some threshold value.
- Excessive packet losses may make it difficult to support certain real-time applications (where the precise threshold of ´excessive´ depends on the application).
- The larger the number of packet losses, the more difficult it is for transport-layer protocols to sustain high bandwidths.
- The sensitivity of real-time applications and of transport-layer protocols according to packet losses becomes important, especially when very large delay-bandwidth products must be supported.

Particularly in supporting multimedia streaming, it is not possible to use a reliable transport protocol for retransmissions of lost packets. Other applications may use such protocols but this always results in large delays if lost packets have to be repaired. Therefore, the packet loss ratio is a very important measurement in order to provide an overview of the current capabilities of the network to transport data without loss.

In order to measure the packet loss ratio, a packet stream which includes sequence numbers is required. Using the RTP protocol for measurements of the packet loss ratio is a typical approach, since most multimedia applications use this protocol which already includes sequence numbers.

Another typical QoS parameter is the bit error rate. This value is often used in the context of physical or link layer protocols. In an IP network, this parameter can be reduced to the packet loss ratio, because each bad packet is being dropped due to a checksum check at a lower layer. Therefore it cannot be seen at the IP layer.

3.2.5 Ratio of reordered or duplicated packets

The last quality of service parameters to be examined are the ratio of reordered packets and the ratio of duplicated packets. Even if the final result of these two ratios is nearly identical as shown in the following, the factors determining each ratio may be very different. This means that both ratios have to be measured in order to allow a qualified analysis of the network behavior and, possibly, to enable a proper modification of the involved components by the network administrators.

Typically, duplicated packets are the results of routing loops or just multiple paths to a destination in multicast networks. These loops may appear, for example, while pruning occurs in dense-mode networks or due to misconfigured routing protocols. The effect is shown in figure 3.5. A number of packets (P1, P2, P3) is simultaneously transmitted over two different network paths. At the destination the packets may arrive in an unexpected order, for example in the sequence P1, P2, P1, P2, etc.). Another reason for duplicated packets is an installation using more than one router in order to achieve a high redundancy. If these routers are not properly configured, duplicated packets may be generated in every IP multicast transmission.

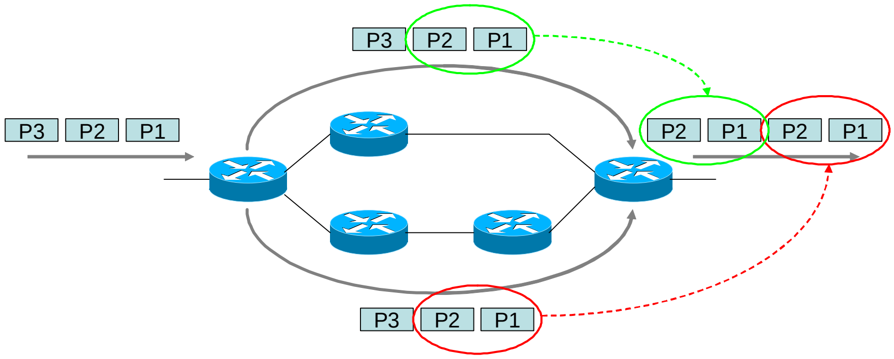


Figure 3.5 - Duplicated Packets due to multiple Paths

The main reasons for reordered packets are the queuing and scheduling mechanisms in the routers along the path. If only a single queue exists and the standard FIFO (first-in, first-out) scheduling mechanism is used, no reordering can appear. Today, in order to implement more intelligence in the network (typically to improve the quality of service for single transmissions), much more difficult mechanisms are used. Therefore, if packets of one packet stream are put into different queues or the employed scheduling algorithm selects packets of the same packet stream out of order, these packets may be reordered. Figure 3.6 shows this effect. Another reason is the existence of multiple paths towards to a destination and the networks use load-sharing mechanisms. The delay on each of the paths may be different and packets traversing the longer path may arrive out-of-order.

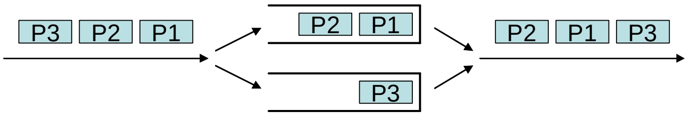


Figure 3.6 - Reordered Packets due to multiple Queues

The applications deal with the appearance of reordered or duplicated packets predominantly by using sequence numbers to identify them. Nevertheless, such packets are relevant for most applications, especially when network support for real-time media streams is assessed. The extent of reordering may be sufficient to cause a received packet to be discarded by functions above the IP layer [147]. It would require a reasonable buffer to handle such packets at the application layer while introducing an additional delay depending on the size of the buffer.

3.3 Real-Time Transport Protocol (RTP)

Most of the current applications which exchange multimedia content or other data with real-time characteristics use the real-time transport protocol. The protocol is defined in RFC 1889 [182]. A new version is in progress at the IETF to enhance the capabilities of RTP [185].

Not only multimedia applications use RTP. Some QoS measurement tools, including the approach presented in section 7 "Multicast Quality Monitor (MQM)", are based on RTP as well. Therefore, an overview of the concepts of RTP and its fellow RTCP (RTP control protocol) are provided in the following. The basic definitions and the header formats can be found in appendix A.

Usually, UDP (user datagram protocol, [163]) is used in conjunction with RTP. UDP is a connection-less transport protocol, which itself is not reliable, i.e. it does not guarantee the delivery of the sent packets. In order to allow a real-time communication, RTP includes features like timestamping and sequence numbering. The applications can calculate the quality of the current transmission based on these mechanisms. For example, the sequence numbers can be used in order to determine the packet loss ratio or to detect duplicated and reordered packets. On the other hand, the time stamps may be used to calculate the one-way delay and the jitter.

It is not only the receiver who is interested in determining the current quality of service from the source towards to the destination. For multimedia transmissions, it is much more important for the sender to use adaptive algorithms in order to deal with the currently available quality.

Therefore, some feedback process is required to inform the sender of the measured quality at the receiver. RTCP has been developed to achieve this task [85]. Using RTCP, the receivers send so called receiver reports (RR) towards the source. These reports inform the sender about the existence of this particular receiver and include information about the reception quality such as the number of lost packet, or the jitter. Sender reports (SR) are used to send out-of-band information about the data stream being sent. For example, a SR includes the number of bytes and the number of packets which have already been sent.

According to RFC 1889, RTCP performs four functions:

(1) The primary function is to provide feedback on the quality of the data distribution. This is an integral part of the RTP's role as a transport protocol and is related to the flow and congestion control functions of other transport protocols. The feedback may be directly useful for the control of adaptive encodings. In IP multicast networks, it is also critical to get feedback from the receivers to diagnose faults in the distribution. Sending reception feedback reports to all participants allows the observer to evaluate whether any problems are local or global. With a distribution mechanism like IP multicast, it is also possible for an entity such as a network service provider who is not otherwise involved in the session to receive the feedback information and act as a third-party monitor to diagnose network problems.

(2) RTCP carries a persistent transport-level identifier for an RTP source called the canonical name or CNAME. Since the SSRC identifier may change if a conflict is discovered or a program is restarted, receivers require the CNAME to keep track of each participant. Receivers also require the CNAME 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. Each participant has sent its control packets to all the others, so that each of them can independently observe the number of participants. This number is used to calculate the rate at which the packets are sent.

(4) An optional function is to convey minimal session control information, for example participant identification to be displayed in the user interface. This is most likely to be useful in ´loosely controlled´ sessions where participants enter and leave without membership control or parameter negotiation. RTCP serves as a convenient channel to reach all the participants, but it is not necessarily expected to support all the control communication requirements of an application. A higher-level session control protocol may be needed.

RTP is a very flexible protocol which can easily be implemented and enhanced to support new demands. Each RTP packet contains a payload type identifier, which describes the profile which has been used to submit the data stream. Many of these profiles are already defined by the IETF, e.g. for different audio and video encodings [162], [185], but it is possible to create new ones in order to use RTP to transport other information.

Another protocol, RTSP (real-time streaming protocol, [184]) has been defined in order to control active multimedia streams, which may use RTP as the transport protocol. Such a protocol is required to allow both the client and the server, to maintain the state of different parallel connections. Typically, RTSP is used to initiate multimedia streamings.

3.4 Metrics for QoS Measurement

The quality of service in the internet has become a very important resource. Several groups of the IETF are working to enhance the standard best effort service to guarantee or, at least, to improve the QoS of data transmissions [35]. The IPPM WG (IP performance measurement working group) mainly searches for mechanisms to describe and measure the QoS of unicast connections. The working group created several measurement methods and metrics describing single QoS parameters. Most of the tools to measure QoS parameters in an IP network are based on these definitions. One example is the multicast quality monitor, which is described in section 7. The next few subsections examine the metrics for:

- connectivity,
- one-way delay,
- delay variation,
- one-way packet loss, and
- reordering.

Unfortunately, no there is no document specifying a metric for the round-trip time. Additionally, the IPPM WG created documents defining a framework for IP performance metrics and providing some applicability statements as well as common one-way measurement protocol requirements. A short overview of these documents precedes the analysis of the single metrics.

3.4.1 Framework for IP Performance Metrics

Several documents have been written by the IPPM WG of the IETF in order to define a generally accepted framework for IP performance measurements [161], [188], [196]. These definitions can be divided into functional requirements: definitions of packet formats and common recommendations.

3.4.1.1 Functional Requirements

The protocols should provide the ability to measure, record, and distribute the results of measurements. To facilitate the broadest possible use of obtained measurement results, the protocols should allow any necessary post-processing. Since measurement session setup and the actual measurement are two different tasks, the test protocol should be separated from the control protocol. Finally, different packet formats should be supported by the measurement setup.

3.4.1.2 Packet Formats

A fundamental property of many internet metrics is that the value of the metric depends on the type of IP packets used for the measurement. The term type-P is used in these metrics to be replaced in the actual measurement by something like "a packet with a payload of B octets". A standard IP packet is defined as a correct IP packet (conforming to the definition of IP [164]) with a TTL sufficient to travel towards its destination and no IP options set unless explicitly noted.

3.4.1.3 Recommendations

Measurement samples: The number of samples needs being clearly defined. The frequency of test packets should be large enough that one can see effects on the link but low enough that the regular traffic on the link is not affected.

Packet size: In all the cases the packet size should be smaller than the MTU to avoid effects resulting from fragmentation and reassembly. Before running the actual test, it should be analyzed if the results depend on the packet size. If this appears to be the case, the size of the packets containing typical user data should be analyzed and used for the test. When test packets larger than the minimum size required by the measurement are sent, the remainder of the packets should be padded with random bits in order to avoid compression being applied to any measurement packets.

3.4.2 Metrics for Measuring Connectivity

The IPPM WG distinguishes between a instantaneous one-way connectivity and an instantaneous two-way connectivity. The definitions of both metrics (defined in [130]) are straight forward.

Metric Name:

Type-P-Instantaneous-Unidirectional-Connectivity

Metric Parameters:

- Src, the IP address of a host)
- Dst, the IP address of a host)
- T, a time

Definition:

Src has Type-P-Instantaneous-Unidirectional-Connectivity to Dst at time T if a type-P packet transmitted from Src to Dst at time T will arrive at Dst.

Metric Name:

Type-P-Instantaneous-Bidirectional-Connectivity

Metric Parameters:

- A1, the IP address of a host)
- A2, the IP address of a host)
- T, a time

Definition:

> Addresses A1 and A2 have Type-P-Instantaneous-Bidirectional-Connectivity at time T if address A1 has Type-P-Instantaneous-Unidirectional-Connectivity to address A2 and address A2 has Type-P-Instantaneous-Unidirectional-Connectivity to address A1.

Beside the instantaneous connectivity, a more useful definition for practical measurements exists, called the two-way temporal connectivity, which is defined in [130] as well. Most of the tools to estimate to connectivity between two hosts rely on this metric.

Metric Name:

> Type-P1-P2-Interval-Temporal-Connectivity

Metric Parameters:

- Src, the IP address of a host
- Dst, the IP address of a host
- T, a time
- dT, a duration

Definition:

> Address Src has Type-P1-P2-Interval-Temporal-Connectivity to address Dst during the interval [T, T+dT] if there exist times T1 and T2, and time intervals dT1 and dT2, such that:
>
> - T1, T1+dT1, T2, T2+dT2 are all in [T, T+dT]
> - T1+dT1 <= T2
> - at time T1, Src has Type-P1-Instantaneous-Connectivity to Dst
> - at time T2, Dst has Type-P2-Instantaneous-Connectivity to Src
> - dT1 is the time taken for a Type-P1 packet sent by Src at time T1 to arrive at Dst
> - dT2 is the time taken for a Type-P2 packet sent by Dst at time T2 to arrive at Src

In order to help develop measurement tools with comparable results, the IETF recommends some initial values for the test of the connectivity. The interval for the measurement should be 60 sec. and a waiting time (W) of 10 sec. for a reply packet is suggested to be useful. 20 packets should be sent during this interval at random times in the interval, uniformly distributed over [T, T+dT+W].

3.4.3 One-Way Delay Metric

The metric for one-way delay measurements is defined as follows [12]:

Metric Name:

Type-P-One-Way-Delay

Metric Parameters:

– Src, the IP address of a host

– Dst, the IP address of s host

– T, a time

Definition:

For a real number dT, "the Type-P-One-Way-Delay from Src to Dst at T is dT" means that Src sent the first bit of a Type-P packet to Dst at wire-time T and that Dst received the last bit of that packet at wire-time T+dT.

"The Type-P-One-Way-Delay from Src to Dst at T is undefined (informally, infinite)" means that Src sent the first bit of a Type-P packet to Dst at wire-time T and that Dst did not receive that packet.

The one-way delay measurement depends on the time stamps taken at different hosts. The IETF document [12] distinguishes between four notions of clock uncertainty:

- Synchronization measures the extent to which two clocks agree on what time it is. For example, the clock on one host might be 5.4 ms ahead of the clock on a second host.
- Accuracy measures the extend to which a given clock agrees with UTC. For example, the clock on a host might be 27.1 ms behind UTC.
- Resolution measures the precision of a given clock. For example, the clock on an old UNIX host might tick only once every 10 ms, and thus has a resolution of only 10 ms.
- Skew measures the change of accuracy, or of synchronization, with time. For example, the clock of a given host might gain 1.3 ms per hour and thus be 27.1 ms behind UTC at one time and only 25.8 ms an hour later. In this case it is said that the clock of the given host has a skew of 1.3 ms per hour relative to UTC, which threatens accuracy. The clock might also have a skew relative to another clock, which threatens synchronization.

Today, a high accuracy and synchronization of the clocks can be achieved using directly connected GPS receivers or, at least, GPS-based NTP (network time protocol) servers.

The IETF recommends to first ensuring the connectivity between the two hosts in order to measure the one-way delay. Nevertheless, most approaches use the same packets for the test of the connectivity and the (two-way) delay measurement.

3.4.4 IP Delay Variation Metric

The IPPM WG defined an IP packet delay variation metric [59], which has been first published by Demichelis [58], for measurements of the jitter in IP networks. The document defines two separate metrics as shown below. The same statements regarding the clock accuracy and synchronization, described in the last subsection, also apply to the measurement of the delay variation.

Metric Name:

Type-P-One-Way-IPDV

Metric Parameters:

- Src, the IP address of a host
- Dst, the IP address of s host
- T1, a time
- T2, a time
- L, a packet length in bits. The packets of a type P packet stream from which the singleton IPDV metric is taken must all be of the same length.
- F, a selection function defining unambiguously the two packets from the stream selected for the metric.
- I1,I2, times which mark the start and the end of the interval in which the packet stream occurs.

Definition:

We are given a Type P packet stream and I1 and I2 such that the first Type P packet to pass measurement point MP1 after I1 is given index 0 and the last Type P packet to pass measurement point MP1 before I2 is given the highest index number.

Type-P-One-Way-IPDV is defined for two packets from Src to Dst selected by the selection function F, as the difference between the value of the Type-P-One-Way-Delay from Src to Dst at T2 and the value of the Type-P-One-Way-Delay from Src to Dst at T1. T1 is the wire-time at which Src sent the first bit of the first packet, and T2 is the wire-time at which Src sent the first bit of the second packet. This metric is derived from the one-way delay metric.

Therefore, for a real number ddT "The Type-P-One-Way-IPDV from Src to Dst at T1, T2 is ddT" means that Src sent two packets, the first at wire-time T1 (first bit), and the second at wire-time T2 (first bit) and the packets were received by Dst at wire-time dT1+T1 (last bit of the first packet), and at wire-time dT2+T2 (last bit of the second packet), and that dT2-dT1=ddT.

"The Type-P-One-Way-IPDV from Src to Dst at T1,T2 is undefined" means that Src sent the first bit of a packet at T1 and the first bit of a second packet at T2 and that Dst did not receive one or both packets.

Metric Name:

Type-P-One-Way-IPDV-Poisson-Stream

Metric Parameters:

- Src, the IP address of a host
- Dst, the IP address of a host
- T0, a time
- Tf, a time
- lambda, a rate in reciprocal seconds
- L, a packet length in bits. The packets of a Type P packet stream from which the sample IPDV metric is taken must all be of the same length.
- F, a selection function defining unambiguously the packets from the stream selected for the metric.
- I(i),I(i+1), i >= 0, pairs of times which mark the start and end of the intervals in which the packet stream from which the measurement is taken occurs. I(0) >= T0 and, assuming that n is the largest index, I(n) <= Tf.

Definition:

A pseudo-random Poisson process is defined such that it begins at or before T0, with average arrival rate lambda, and ends at or after Tf. Those time values T(i) greater than or equal to T0 and less than or equal to Tf are then selected for packet generation times.

Each packet falling within one of the sub-intervals I(i), I(i+1) is tested to determine whether it meets the criteria of the selection function F as the first or second of a packet pair needed to compute IPDV. The sub-intervals can be defined such that a sufficient number of singleton samples for valid statistical estimates can be obtained.

The triples defined above consist of the transmission times of the first and second packets of each singleton included in the sample, and the IPDV in seconds.

Because packets can be lost, duplicated or reordered, each test packet should be marked with a sequence number. For duplicated packets only the first copy should be considered.

3.4.5 One-Way Packet Loss Metric

Another metric defined by the IETF describes the measurement of the packet loss ratio [13].

Metric Name:

Type-P-One-Way-Packet-Loss

Metric Parameters:

- Src, the IP address of a host
- Dst, the IP address of a host
- T, a time

Definition:

"The Type-P-One-Way-Packet-Loss from Src to Dst at T is 0" means that Src sent the first bit of a Type-P packet to Dst at wire-time T and that Dst received that packet.

"The Type-P-One-way-Packet-Loss from Src to Dst at T is 1" means that Src sent the first bit of a type-P packet to Dst at wire-time T and that Dst did not receive that packet.

Thus, Type-P-One-Way-Packet-Loss is 0 exactly when Type-P-One-Way-Delay is a finite value, and it is 1 exactly when Type-P-One-Way-Delay is undefined.

The last definition suggests that the packet loss can be calculated using only the measurement of the one-way delay. Nevertheless, the measurement of the packet loss ratio can be easier because no synchronization of the clocks of the involved hosts is required.

The Type-P-One-Way-Packet-Loss metric defines a single test. In order to support a sample of measurement packets, an additional metric is defined as follows [13]:

Metric Name:

Type-P-One-Way-Packet-Loss-Poisson-Stream

Metric Parameters:

- Src, the IP address of a host
- Dst, the IP address of s host
- T0, a time
- Tf, a time
- lambda, a rate in reciprocal seconds

Definition:

Given T0, Tf, and lambda, we compute a pseudo-random Poisson process beginning at or before T0, with average arrival rate lambda, and ending at or after Tf. Those time values greater than or equal to T0 and less than or equal to Tf are then selected. For each of the times in this process, we obtain the value of Type-P-One-way-Packet-Loss at this time. The value of the sample is the sequence made up of the resulting <time, loss> pairs. If there are no such pairs, the sequence is of length zero and the sample is said to be empty.

A few more definitions have been created by the IETF [122] in order to allow a more general understanding of the behavior of packet loss:

- Bursty loss: The loss involving consecutive packets of a stream.
- Loss Distance: The difference in sequence numbers of two consecutively lost packets which may or may not be separated by successfully received packets.

3.4.6 Reordering Metric

The last metric to be mentioned in this context is the measurement of the ratio of reordered packets. The definition of this metric according to the IPPM WG [147] is as follows:

Metric Name:

Type-P-Non-Reversing-Order

Metric Parameters:

- Src, the IP address of a host
- Dst, the IP address of s host
- SrcNum, the packet sequence number applied at the Src, in units of messages or bytes.
- NextExp, the next expected sequence number at the Dst, in units of messages, time, or bytes.

Definition:

In-order packets have sequence numbers greater than or equal to the value of NextExp. Each new packet which is "in-order" will increase NextExp (typically by 1 for message numbering, or the payload size plus 1 for byte numbering). The next expected value cannot decrease, thereby specifying non-reversing order as the basis to identify reordered packets.

A reordered packet outcome occurs when a single IP packet at the Dst measurement point results in the following: The packet has a Src sequence number lower than the next expected (NextExp), and therefore the packet is reordered. The NextExp value does not change on the arrival of this packet.

3.5 QoS Measurement Tools for Multicast Networks

Multicast has been deployed very slowly in the internet. Because it started as a virtual network, it was not possible to provide any kind of quality. With the beginning change to natively routed multicast it became important to control and measure the current QoS [179], [180]. But even at the outset of IP multicast routing, the requirement of connectivity and quality measurements started to increase due to the number of possible interconnectivity problems.

These measurements in an IP multicast environment require separate tests for the connection between each sender and each receiver [129]. The network administrator is required to locate proper places within the network to place some monitoring probes. Ideally, each participating client of a particular multicast application is used for the measurements. The result is a matrix, which allows identification of the network parts that fail to forward IP multicast properly. Therefore, due to the principles of IP multicast routing, it is required to have many measurement probes. Especially, this applies to services like conferences which are working according to the many-to-many transmission model.

The next subsections introduce current approaches to measure the reliability and the quality of service of an IP multicast network [181]. Each is capable of providing information about a particular network. Two of these tools are able to measure different QoS parameters in a multicast network: the multicast reachability monitor and the multicast beacon. Both of these approaches are described in more detail. Additionally, both tools have been tested and the results are provided after the mentioned description.

Unfortunately, none of these ideas examine the problem where to place the measurement probes. In section 5 "Modeling IP Multicast Networks and Services", an approach is shown, which allows to find optimum places for the probes to prevent the easily resulting scaling problem.

3.5.1 ping / traceroute

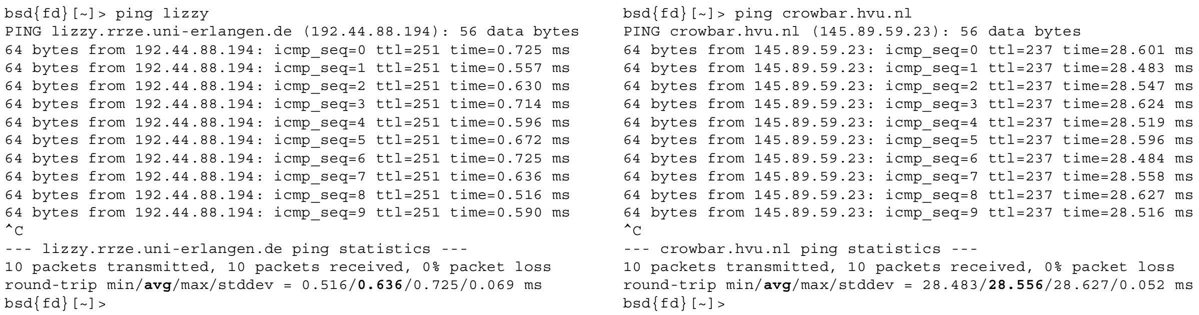
Although it only works in unicast scenarios, ping is still one of the most commonly used tools to check the bidirectional connectivity as well as to measure the round-trip time. It uses ICMP (internet control message protocol, [165]) messages in order to request an response packet from a destination.

ICMP defines a type field as well as a control field in the header of each ICMP packet. An excerpt of the type values is shown in table 3.1 [72]. Ping employs the echo request and the echo reply messages in order to test a network connection.

Type Field	Description
0	Echo Reply
3	Destination Unreachable
5	Redirect
8	Echo Request
11	Time Exceeded

Table 3.1 - Type Values of ICMP

After a successful reception to the response, ping calculates the RTT. Two examples are shown in figure 3.7. Depending on the distance between the nodes and the available resources in the network, the delay oscillates around some average value. The delay within the campus network at the University of Erlangen-Nuremberg (3 hops between both nodes), shown on the left, is much smaller (0.6 ms) than the delay between two distant hosts (18 hops between both nodes), shown on the right in the figure (28.5 ms).



```
bsd{fd}[~]> ping lizzy
PING lizzy.rrze.uni-erlangen.de (192.44.88.194): 56 data bytes
64 bytes from 192.44.88.194: icmp_seq=0 ttl=251 time=0.725 ms
64 bytes from 192.44.88.194: icmp_seq=1 ttl=251 time=0.557 ms
64 bytes from 192.44.88.194: icmp_seq=2 ttl=251 time=0.630 ms
64 bytes from 192.44.88.194: icmp_seq=3 ttl=251 time=0.714 ms
64 bytes from 192.44.88.194: icmp_seq=4 ttl=251 time=0.596 ms
64 bytes from 192.44.88.194: icmp_seq=5 ttl=251 time=0.672 ms
64 bytes from 192.44.88.194: icmp_seq=6 ttl=251 time=0.725 ms
64 bytes from 192.44.88.194: icmp_seq=7 ttl=251 time=0.636 ms
64 bytes from 192.44.88.194: icmp_seq=8 ttl=251 time=0.516 ms
64 bytes from 192.44.88.194: icmp_seq=9 ttl=251 time=0.590 ms
^C
--- lizzy.rrze.uni-erlangen.de ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.516/0.636/0.725/0.069 ms
bsd{fd}[~]>
```

```
bsd{fd}[~]> ping crowbar.hvu.nl
PING crowbar.hvu.nl (145.89.59.23): 56 data bytes
64 bytes from 145.89.59.23: icmp_seq=0 ttl=237 time=28.601 ms
64 bytes from 145.89.59.23: icmp_seq=1 ttl=237 time=28.483 ms
64 bytes from 145.89.59.23: icmp_seq=2 ttl=237 time=28.547 ms
64 bytes from 145.89.59.23: icmp_seq=3 ttl=237 time=28.624 ms
64 bytes from 145.89.59.23: icmp_seq=4 ttl=237 time=28.519 ms
64 bytes from 145.89.59.23: icmp_seq=5 ttl=237 time=28.596 ms
64 bytes from 145.89.59.23: icmp_seq=6 ttl=237 time=28.484 ms
64 bytes from 145.89.59.23: icmp_seq=7 ttl=237 time=28.558 ms
64 bytes from 145.89.59.23: icmp_seq=8 ttl=237 time=28.627 ms
64 bytes from 145.89.59.23: icmp_seq=9 ttl=237 time=28.516 ms
^C
--- crowbar.hvu.nl ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max/stddev = 28.483/28.556/28.627/0.052 ms
bsd{fd}[~]>
```

Figure 3.7 - Examples of using ping

In order to get information about the path through the network which is used for packets between two hosts traceroute can be used. Traceroute uses the TTL value to achieve information about the employed path. By sending packets with an increasing TTL (starting with TTL=1), the routers along the path are required to drop the test packet when TTL=0 is reached (the TTL is decreased at each hop by 1) and to reply with a special ICMP message informing the originator of the packet that the TTL is too small to reach the destination. Additionally, traceroute computes the RTT to each hop. The behavior is shown in figure 3.8. The full path between two hosts was traced. The same destination was used as in the previous example. As expected, traceroute has measured nearly the same round-trip times between the same hosts (28.8 ms) as ping.

```
bsd{fd}[~]> traceroute crowbar.hvu.nl
traceroute to crowbar.hvu.nl (145.89.59.23), 64 hops max, 40 byte packets
 1  reliant.gate.uni-erlangen.de (131.188.3.1)  0.512 ms  0.510 ms  0.500 ms
 2  suedstern.gate.uni-erlangen.de (131.188.21.65)  0.702 ms  0.684 ms  0.530 ms
 3  botany-bay.gate.uni-erlangen.de (131.188.20.117)  0.342 ms  0.424 ms  0.329 ms
 4  enterprise.gate.uni-erlangen.de (131.188.20.102)  0.300 ms  0.341 ms  0.346 ms
 5  excelsior.gate.uni-erlangen.de (131.188.5.1)  3.238 ms  3.285 ms  3.272 ms
 6  ar-erlangen1.g-win.dfn.de (188.1.36.1)  0.324 ms  0.361 ms  0.323 ms
 7  cr-erlangen1.g-win.dfn.de (188.1.72.1)  0.521 ms  0.366 ms  0.333 ms
 8  cr-hannover1.g-win.dfn.de (188.1.18.118)  21.241 ms  21.089 ms  21.259 ms
 9  cr-frankfurt1.g-win.dfn.de (188.1.18.181)  21.233 ms  21.262 ms  22.224 ms
10  ir-frankfurt2.g-win.dfn.de (188.1.80.38)  21.396 ms  21.003 ms  21.057 ms
11  dfn.de1.de.geant.net (62.40.103.33)  21.414 ms  21.654 ms  21.524 ms
12  de.nl1.nl.geant.net (62.40.96.54)  27.867 ms  27.841 ms  27.695 ms
13  PO6-0.BR1.Amsterdam1.surf.net (62.40.103.98)  27.673 ms  27.794 ms  27.876 ms
14  PO12-0.CR1.Amsterdam1.surf.net (145.145.166.1)  27.865 ms  27.980 ms  27.695 ms
15  PO14-0.CR2.Amsterdam1.surf.net (145.145.160.22)  28.019 ms  30.098 ms  28.469 ms
16  PO0-0.AR5.Utrecht1.surf.net (145.145.163.50)  28.631 ms  28.611 ms  28.471 ms
17  hvu-router.Customer.surf.net (145.145.16.2)  28.649 ms  28.442 ms  28.459 ms
18  msfc-itsupport.wan.hvu.nl (145.89.1.10)  28.776 ms  28.692 ms  28.861 ms
19  crowbar.hvu.nl (145.89.59.23)  28.846 ms  28.622 ms  28.835 ms
bsd{fd}[~]>
```

Figure 3.8 - Example of using traceroute

To distinguish between failures of the multicast routing and the network itself, each measurement of the connectivity within the multicast network should be accompanied by a simple test of the unicast connectivity, for example using the ping mechanism.

3.5.2 mrinfo

The first tool to be discussed is mrinfo. This tool is used to get information about a multicast router. Especially during the time of the old MBone, which consisted mainly of tunnel connections in order to build a virtual multicast network using the unicast IP infrastructure, the loss of connectivity in the network was often a result of interoperability problems between different multicast routers. Mrinfo has been developed in order to acquire information about the neighborship relations of single multicast routers. It shows all the multicast enabled interfaces of a router including its neighbors and the used multicast routing protocol on each link. An example is shown in figure 3.9. A router in Erlangen was queried. Mrinfo has discovered all its tree multicast neighbors. On all links PIM is employed as the multicast routing protocol.

```
bsd{fd}[~]> mrinfo enterprise
131.188.5.2 (enterprise.gate.uni-erlangen.de) [version 12.1]:
  131.188.5.2 -> 131.188.5.1 (excelsior.gate.uni-erlangen.de) [1/8/pim/querier]
  131.188.20.102 -> 131.188.20.101 (botany-bay.gate.uni-erlangen.de) [1/8/pim/querier]
  131.188.20.106 -> 131.188.20.105 (constellation.gate.uni-erlangen.de)
    [1/8/pim/querier]
bsd{fd}[~]>
```

Figure 3.9 - Example of using mrinfo

Even today mrinfo is still useful. It can be used to query all multicast routers in order to generate a map of the multicast infrastructure. Additionally, connectivity problems can be analyzed by using mrinfo.

3.5.3 mtrace

Mtrace is one of the most well-known tools to test the reachability including a full trace of the used multicast path between two systems. A special feature has been built into the multicast routers [79] in order to be able to trace the multicast path. It is not possible to use the same mechanism, ICMP (internet control message protocol, [165]) which is used for the unicast traceroute. A first example of using mtrace is shown in figure 3.10. The multicast path between two hosts in Erlangen was analyzed. Mtrace has detected four routers along the path. Additionally, mtrace tries to acquire as much information about the available QoS as possible. First, it measures the round-trip time and secondly, the current packet loss ratio is estimated.

```
lisa{root}[~]# mtrace lizzy
Mtrace from 192.44.88.194 to 131.188.3.150 via group 224.2.0.1
Querying full reverse path...
  0  lisa.rrze.uni-erlangen.de (131.188.3.150)
 -1  reliant.gate.uni-erlangen.de (131.188.3.1)  PIM  thresh^ 8
 -2  suedstern.gate.uni-erlangen.de (131.188.21.65)  PIM  thresh^ 8
 -3  botany-bay.gate.uni-erlangen.de (131.188.20.117)  PIM  thresh^ 8
 -4  fritz.gate.uni-erlangen.de (131.188.20.134)  PIM  thresh^ 8
 -5  lizzy.rrze.uni-erlangen.de (192.44.88.194)
Round trip time 77 ms

Waiting to accumulate statistics... Results after 10 seconds:

  Source        Response Dest    Packet Statistics For     Only For Traffic
192.44.88.194   131.188.3.150    All Multicast Traffic     From 192.44.88.194
     v       __/  rtt    8 ms    Lost/Sent = Pct  Rate       To 224.2.0.1
192.44.88.1
131.188.20.134  fritz.gate.uni-erlangen.de
     v     ^      ttl    8        0/0    = --%   0 pps     0/0    = --%   0 pps
131.188.20.133
131.188.20.117  botany-bay.gate.uni-erlangen.de
     v     ^      ttl    9        0/402  =  0%  40 pps     0/0    = --%   0 pps
131.188.20.118
131.188.21.65   suedstern.gate.uni-erlangen.de
     v     ^      ttl   10        0/416  =  0%  41 pps     0/0    = --%   0 pps
131.188.21.66
131.188.3.1     reliant.gate.uni-erlangen.de
     v      \__   ttl   11          416         41 pps       0            0 pps
131.188.3.150   131.188.3.150
  Receiver      Query Source

lisa{root}[~]#
```

Figure 3.10 - Example of using mtrace

Despite the fact that mtrace is a very useful tool to check the multicast connectivity and routing, it also has its shortcomings. Therefore it may happen that mtrace does not trace the path successfully even if the multicast forwarding works very well, just because there are routers in the path which have not implemented the mtrace features or where the feature has been disabled administratively. Such an effect is shown in figure 3.11 where mtrace stops at the border gateway of the German research network even if the multicast connectivity towards the traced destination was provided.

```
lisa{root}[~]# mtrace crowbar.hvu.nl
Mtrace from 145.89.59.23 to 131.188.3.150 via group 224.2.0.1
Querying full reverse path...
  0  lisa.rrze.uni-erlangen.de (131.188.3.150)
 -1  reliant.gate.uni-erlangen.de (131.188.3.1)  PIM  thresh^ 8
 -2  suedstern.gate.uni-erlangen.de (131.188.21.65)  PIM  thresh^ 8
 -3  botany-bay.gate.uni-erlangen.de (131.188.20.117)  PIM  thresh^ 8
 -4  enterprise.gate.uni-erlangen.de (131.188.20.102)  PIM  thresh^ 8
 -5  excelsior.gate.uni-erlangen.de (131.188.5.1)  PIM  thresh^ 0
 -6  ar-erlangen1.g-win.dfn.de (188.1.36.1)  PIM  thresh^ 32
 -7  cr-erlangen1.g-win.dfn.de (188.1.72.1)  Unknown protocol code 8  thresh^ 0
 -8  cr-hannover1.g-win.dfn.de (188.1.18.118)  Unknown protocol code 8  thresh^ 0
 -9  cr-frankfurt1.g-win.dfn.de (188.1.18.181)  Unknown protocol code 8  thresh^ 0
-10  ir-frankfurt2.g-win.dfn.de (188.1.80.38)  Unknown protocol code 8  thresh^ 0
-11  dfn.de1.de.geant.net (62.40.103.33)  PIM  thresh^ 1  Unknown error code 8
Round trip time 182 ms

Waiting to accumulate statistics... Results after 10 seconds:

  Source        Response Dest    Packet Statistics For     Only For Traffic
   * * *        131.188.3.150    All Multicast Traffic     From 145.89.59.23
     v       __/  rtt   37 ms    Lost/Sent = Pct  Rate       To 224.2.0.1
0.0.0.0
62.40.103.33    dfn.de1.de.geant.net Unknown error code 8
     v     ^      ttl    1          11882     1188 pps       0            0 pps
62.40.103.34
188.1.80.38     ir-frankfurt2.g-win.dfn.de
     v     ^      ttl    2      -46/11882=  0%1188 pps     0/0    = --%   0 pps
188.1.80.37
188.1.18.181    cr-frankfurt1.g-win.dfn.de
     v     ^      ttl    3     -746/1107 =-66% 110 pps     0/0    = --%   0 pps
188.1.18.182
188.1.18.118    cr-hannover1.g-win.dfn.de
     v     ^      ttl    4     -619/1676 =-36% 167 pps     0/0    = --%   0 pps
188.1.18.117
188.1.72.1      cr-erlangen1.g-win.dfn.de
     v     ^      ttl    5       15/2401 =  1% 240 pps     0/0    = --%   0 pps
188.1.72.2
188.1.36.1      ar-erlangen1.g-win.dfn.de
 v     ^      ttl   32      708/708  =100%  70 pps     0/0    = --%   0 pps
131.188.1.1
131.188.5.1     excelsior.gate.uni-erlangen.de
     v     ^      ttl   33     -718/0    = --%   0 pps     0/0    = --%   0 pps
131.188.5.2
131.188.20.102  enterprise.gate.uni-erlangen.de
     v     ^      ttl   34        0/718  =  0%  71 pps     0/0    = --%   0 pps
131.188.20.101
131.188.20.117  botany-bay.gate.uni-erlangen.de
     v     ^      ttl   35        0/718  =  0%  71 pps     0/0    = --%   0 pps
131.188.20.118
131.188.21.65   suedstern.gate.uni-erlangen.de
     v     ^      ttl   36        0/718  =  0%  71 pps     0/0    = --%   0 pps
131.188.21.66
131.188.3.1     reliant.gate.uni-erlangen.de
     v      \__   ttl   37          718         71 pps       0            0 pps
131.188.3.150   131.188.3.150
  Receiver      Query Source

lisa{root}[~]#
```

Figure 3.11 - Example of using mtrace

Often, mtrace shows such effects when connections are traced which involve the backbone networks of different ISPs. In addition to the traced path, mtrace is able to compute some packet statistics. Because several consecutive packets are sent in order to trace the path and each packet is required to be responded by a router, the number of lost packets can be calculated.

3.5.4 MHealth

Another tool, MHealth [131], [132], has been designed to provide a graphical overview of a particular multicast tree. Actually, MHealth is just a graphical front-end for mtrace. Therefore, it uses mtrace to trace the path from active sources in a particular multicast group of all receivers. In order to detect the active participants, it joins a given multicast group and listens for RTCP packets. Analyzing these packets allows identification of all active members in this group.

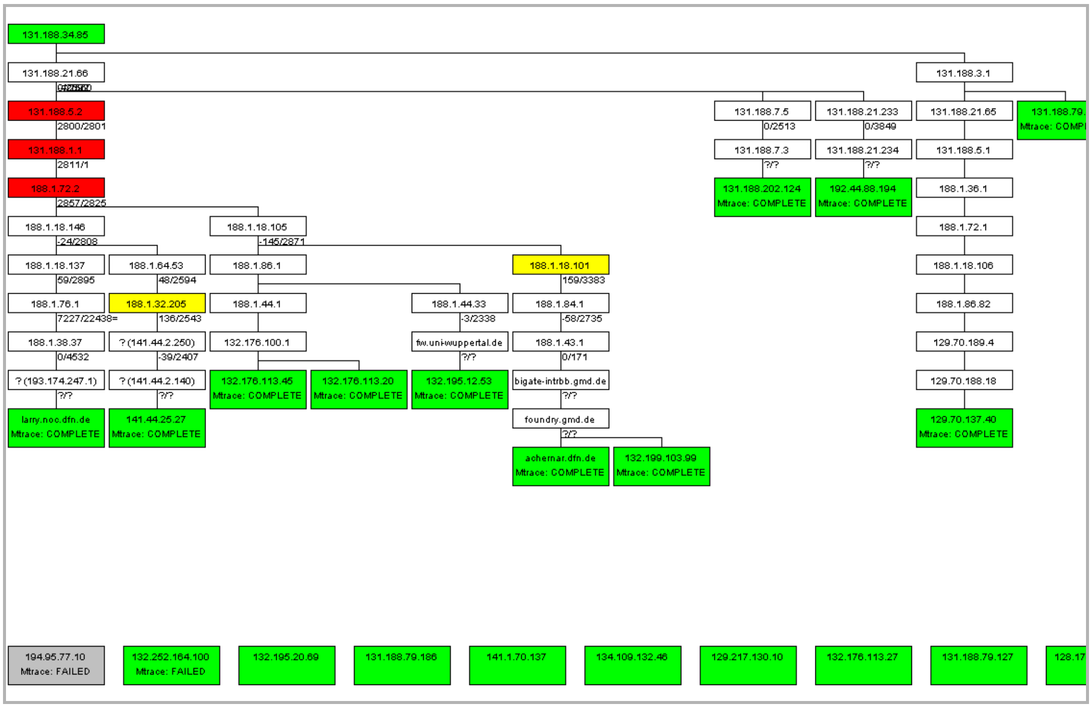


Figure 3.12 - Example of using MHealth

Figure 3.12 shows an example of using MHealth. This measurement has been taken during the world soccer championship in Korea. A server at the University of Erlangen-Nuremberg (131.188.34.85, shown on the top of the screenshot) was broadcasting the games using the multicast tools vic and vat for the video and the audio transmission respectively. Due to the popularity of soccer, it was possible to have quite a few participants receiving this session. MHealth was used to calculate and draw a map of this session. The color of each represented host specifies the last measured packet loss ratio to this destination. Unfortunately, there were too many members in this group, so that MHealth was not able to draw a complete map. The remaining hosts are shown in the bottom of the screenshot. With using a separate RTCP monitor, a list of all participants has been created. 102 participants from all over Germany were recorded.

MHealth does not introduce any new kind of QoS measurement but it provides an attractive graphical front-end for mtrace allowing the creation of a connectivity map for all the members of a particular session.

3.5.5 mlisten

Mlisten [4] was developed to allow a network engineer to get an overview of all multicast traffic in the internet which can be received at a particular site. In order to receive all this traffic, mlisten starts to listen for SAP announcements and tunes into all active multicast sessions. For each participant sending packets to any of these sessions mlisten creates a single record. After some running time, the user of mlisten is able to produce a graph including all active sessions and all active participants. The group behavior can be determined [5], [9].

It should be mentioned that mlisten does not really provide any kind of quality of service measurement but it is able to produce a unidirectional connectivity graph. Actually, this graph is a tree with sources at each leaf and the destination at its root. Even if only currently active sources can be included, the tree allows a first approximation of the connectivity between distributed sites in the internet and the local host.

According to the working principle of mlisten, this tool should be used carefully. Mlisten tries to receive all multicast traffic from the internet. This can be much more than the internet connection that particular site is able to carry. Therefore, the user of mlisten can easily overload his own internet connection.

In order to evaluate the functionality of mlisten, we let mlisten run for 24 hours. The result is impressive. We received about 1.6 TB (1’600 GB) of multicast data during this time. The result of this measurement is shown in figure 3.13. A first conclusion is that it seems as if nearly all the received traffic has been initiated by sources which stream continuously. This is quite an interesting result because such data streams can be used for quality of service measurements as shown in a following section.

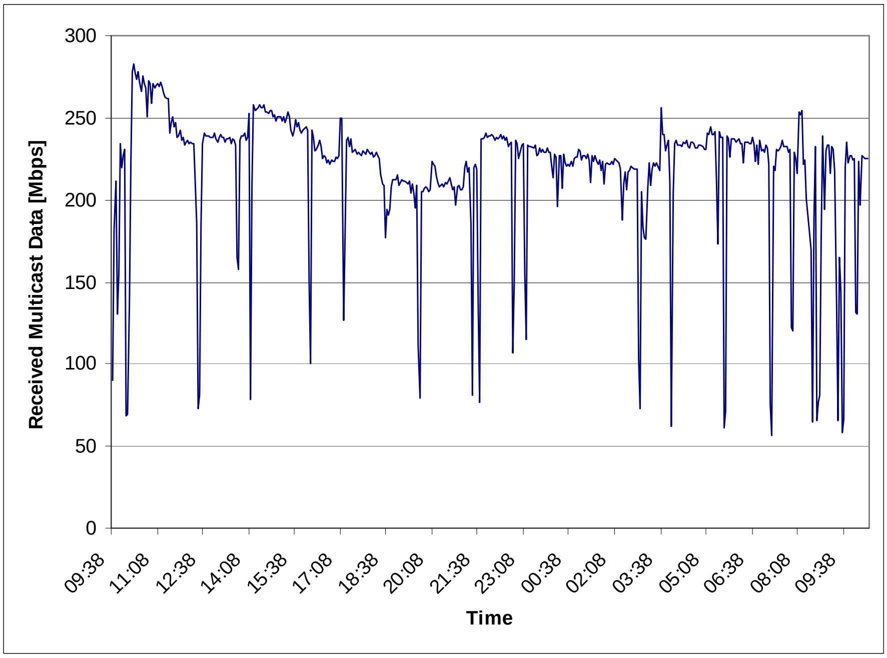


Figure 3.13 - Received Multicast Data using mlisten

Another result, shown in figure 3.14, is that the number of active sessions and senders remain the same. Even the number of receivers does not change. This is interesting, because it was expected that most receivers tune in and out of the sessions more frequently, showing a peak at the business hours.

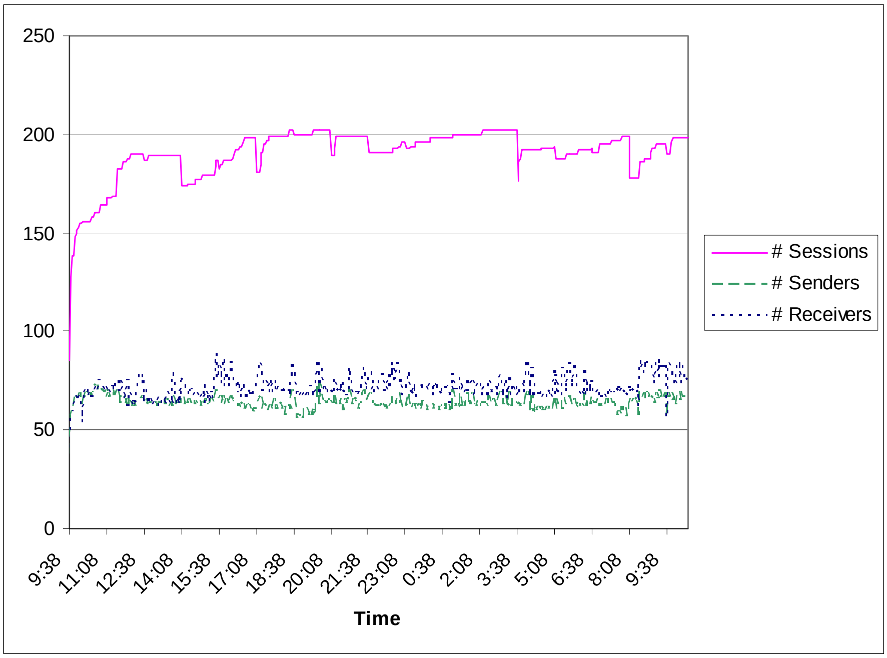


Figure 3.14 - Distribution of Active Multicast Sessions, Senders and Receivers

During the test we recorded 37 different senders and 91 different receivers participating in 92 sessions. The total number of senders and receivers, including the different sessions in which they simultaneously participated, was 68 and 212 respectively.

3.5.6 Multicast Reachability Monitor (MRM)

The multicast reachability monitor (MRM, [10]), formerly known as the multicast route monitor, has been developed to allow a centralized reachability management based on probes located all over the multicast network. End systems can be used as probes as well as the multicast routers themselves. The MRM started as an IETF draft [11], but the IETF decided to stop the work on the MRM because it interfered with current activities in the SNMPv3 specification. Nevertheless, the MRM is a very interesting tool, which was a starting point for further developments such as the MQM described in section 7.

The multicast reachability monitor defines three different processes, the MRM manager, the test sender and the test receiver, shown in figure 3.15. The manager process controls the measurement by starting and stopping the test-sender and test-receiver processes. Once started, each test-sender creates a packet stream and sends it into the network. The test-receiver processes are responsible for receiving and analyzing these data packets and for reporting the acquired information to the manager process.

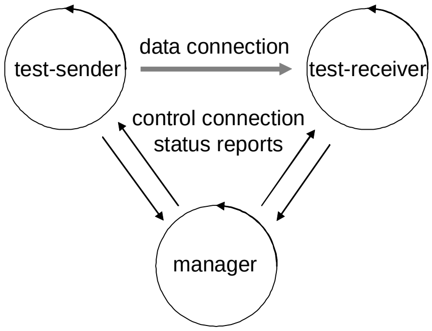


Figure 3.15 - MRM Processes

Cisco Systems, Inc., who included the functionality into the operation system (IOS) of their routers [216], have provided the first implementation of the MRM. Another implementation has been developed for Sun Solaris systems [178].

Figure 3.16 provides an overview of the interoperating parts of the MRM. The figure shows a simple network consisting of three routers and two end systems. The manager process is running on a router and multiple test sender and receiver processes are configured on the other routers or end systems. According to the specification of the MRM, each device in the multicast network can run one or more of the MRM processes. Controlled by the manager, the multicast reachability monitor is able to create a configurable packet flow at each test sender. Using the received packets, the test receivers are able to compute measurement results, such as the packet loss ratio, which provide a good estimation of the reliability of the network. The MRM clients inform the manager process, therefore the latter can provide the measured data to the network administrator for further processing [7].

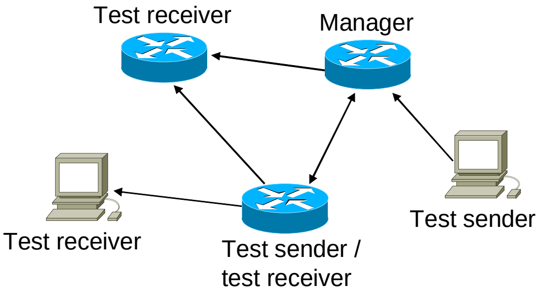


Figure 3.16 - Active Parts of the MRM

The MRM uses RTP as the transport protocol for the measurement packets. This protocol already defines fields for sequence numbers and time stamps in the packet headers. In addition, it is used by nearly every application transmitting multimedia content over the internet. Using packet flows originated by the test senders as well as other RTP streams originated from active IP multicast services, the test receiver can compute the packet loss ratio.

The rate of the generated packet stream is configurable by defining an inter-packet delay between 50 ms and a couple of hours. In order to refresh the forwarding state in all the involved routers and to prevent an impact on other applications, packets should be sent at least once a minute (typically, the state in the routers is timed out after 3 minutes). The maximum data rate depends on the available resources and on the current load of the network.

The definition of the multicast reachability monitor already includes mechanisms to measure various types of QoS parameters such as the packet loss ratio and the delay. Unfortunately, the currently available implementations allow the measurement of the packet loss ratio only. Other restrictions based on the definition itself include missing methods for such important measurements as the one-way delay or the jitter. The status reports are sent using IP multicast and the unreliable transport protocol UDP as well. This may lead to incomplete information at the manager station. Because all the measurement probes are built as easily as possible, a global view on the functionality and quality of the multicast network becomes impossible.

The functionality of the MRM was tested in the campus network of the University of Erlangen-Nuremberg. The test environment is shown in figure 3.17.

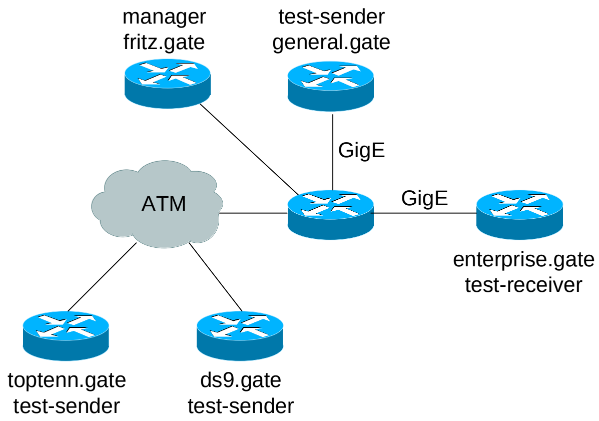


Figure 3.17 - Test Environment for the Evaluation of the MRM

Several routers (test-senders) were configured to send packets to a single destination router (test-receiver). The manager is running on a different router. Even if the figure shows only a few MRM stations, the setup included about 10 routers, most of which are interconnected by a gigabit ethernet backbone. Only a few are attached to an ATM backbone using interfaces from E3 up to OC-12.

The distribution of the lost packets over the time is shown in figure 3.18 and figure 3.19 for the slightly loaded routers and a heavily loaded router respectively.

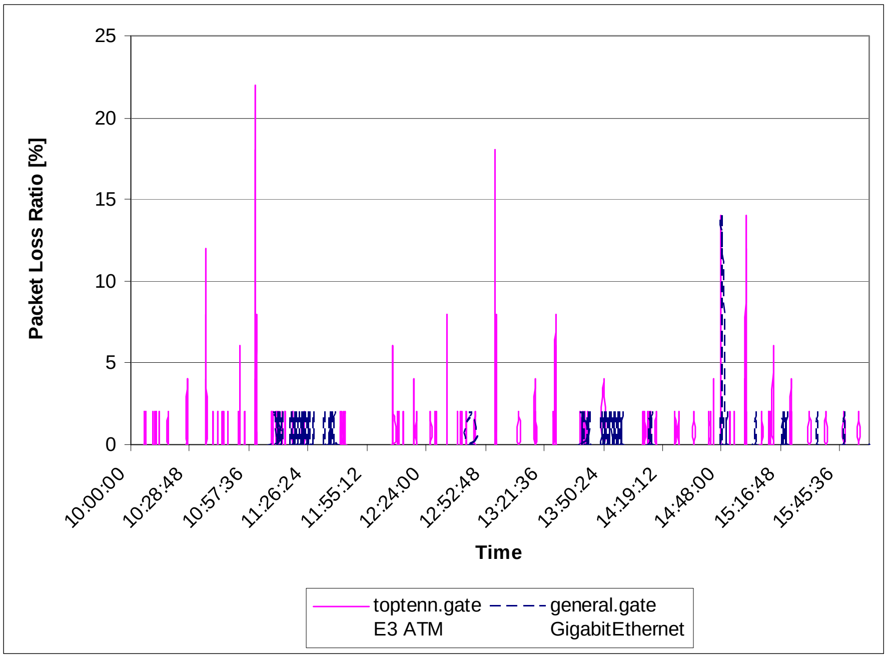


Figure 3.18 - MRM Test Results from slightly loaded Routers

During the measurement, the MRM has only shown isolated lost packets. The mean packet loss ratio within the test window is between 0.8% and 1.5% from the more slightly loaded routers and about 2.6% from a heavily loaded router.

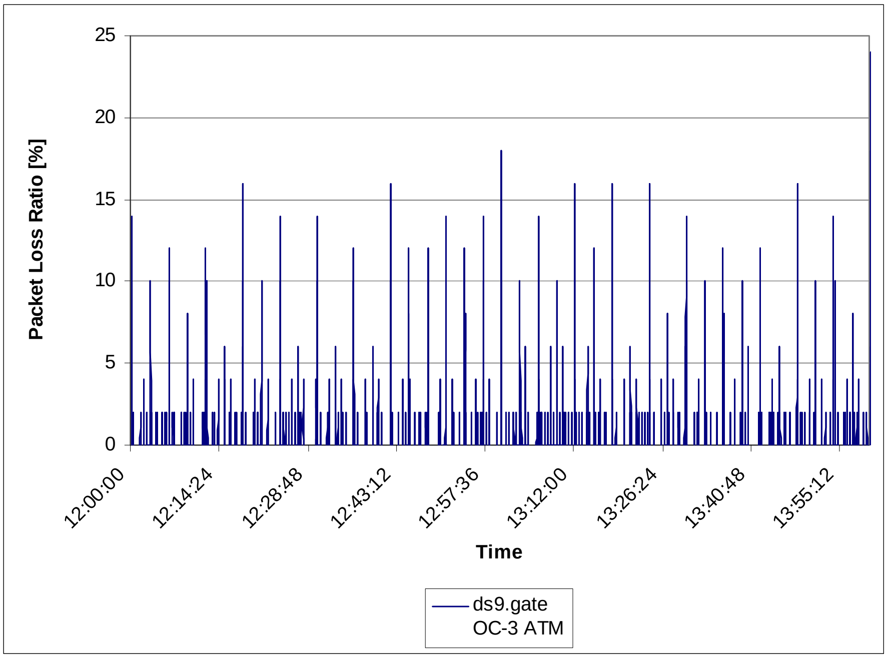


Figure 3.19 - MRM Test Result from a heavily loaded Router

It has already been shown that the multicast reachability monitor is a useful tool to supervise the current packet loss behavior of a particular network. Due to the available end host implementation, it is not necessarily required to configure the MRM instances on the routers in the network. Nevertheless, the MRM is very limited with regards to the type of available implemented measurements. Additionally, it is not possible to simulate the behavior of a typical application in order to examine the current available quality of service in relation to the requirements of that particular application.

3.5.7 Multicast Beacon

The multicast beacon [41] is the result of a research project from the NLANR (National Laboratory for Applied Network Research). Currently, there is an implementation in JAVA for the beacon clients available, which should run on nearly every end system with an installed JVM (java virtual machine). The beacon server consists of a perl program. The associated components and their interaction is shown in figure 3.20.

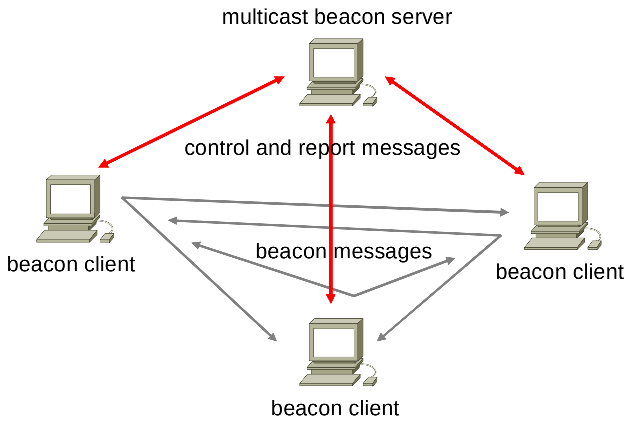
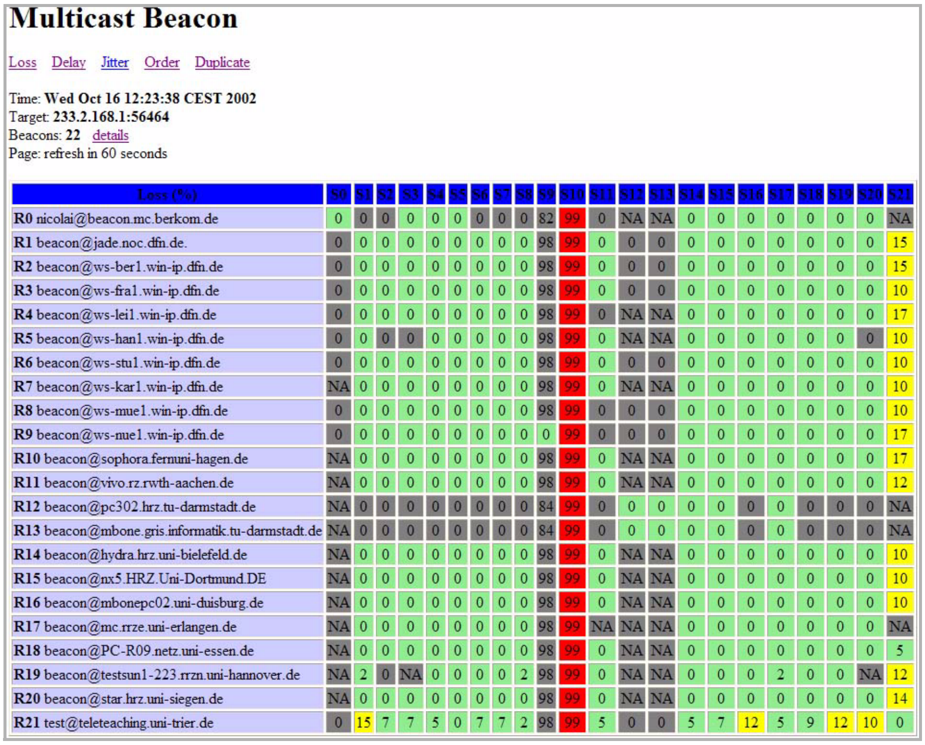


Figure 3.20 - Components of the Multicast Beacon

The principles of the multicast beacon and the MRM are very similar. The definition of the multicast beacon includes a server computing the QoS parameters from measurement results and the clients, beacons, which are sending and receiving the measurement packets. All the beacons interact directly with each other by constantly sending IP multicast packets using the RTP protocol to an administratively configured multicast group. Each beacon client reports its measured data, i.e. the results of received packets (beacons) to the server. The server calculates a matrix including each active client and allows these results to be accessed via a web gateway.

An example of the representation of the measured data is shown in figure 3.21. It can be seen that the loss ratios from sender 9 and 10 to all other participants is much too high (98-99%). All the green field describe a perfect behavior of the network, thus, the packet loss ratio is very small.

The multicast beacon uses the received measurement packets to calculate the packet loss ratio (this information can be used to create a simple connectivity matrix as well), the one-way delay, the jitter and the ratio of reordered and duplicated packets. For the one-way delay and the jitter measurements, the multicast beacons assume the clock of each beacon to be synchronized.



Multicast Beacon

Loss Delay Jitter Order Duplicate

Time: **Wed Oct 16 12:23:38 CEST 2002**
Target: **233.2.168.1:56464**
Beacons: **22** details
Page: refresh in 60 seconds

Loss (%)	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21
R0 nicolai@beacon.mc.berkom.de	0	0	0	0	0	0	0	0	0	82	99	0	NA	NA	0	0	0	0	0	0	0	NA
R1 beacon@jade.noc.dfn.de.	0	0	0	0	0	0	0	0	0	98	99	0	0	0	0	0	0	0	0	0	0	15
R2 beacon@ws-ber1.win-ip.dfn.de	0	0	0	0	0	0	0	0	0	98	99	0	0	0	0	0	0	0	0	0	0	15
R3 beacon@ws-fra1.win-ip.dfn.de	0	0	0	0	0	0	0	0	0	98	99	0	0	0	0	0	0	0	0	0	0	10
R4 beacon@ws-lei1.win-ip.dfn.de	0	0	0	0	0	0	0	0	0	98	99	0	NA	NA	0	0	0	0	0	0	0	17
R5 beacon@ws-han1.win-ip.dfn.de	0	0	0	0	0	0	0	0	0	98	99	0	NA	NA	0	0	0	0	0	0	0	10
R6 beacon@ws-stu1.win-ip.dfn.de	0	0	0	0	0	0	0	0	0	98	99	0	0	0	0	0	0	0	0	0	0	10
R7 beacon@ws-kar1.win-ip.dfn.de	NA	0	0	0	0	0	0	0	0	98	99	0	NA	NA	0	0	0	0	0	0	0	10
R8 beacon@ws-mue1.win-ip.dfn.de	0	0	0	0	0	0	0	0	0	98	99	0	0	0	0	0	0	0	0	0	0	10
R9 beacon@ws-nue1.win-ip.dfn.de	0	0	0	0	0	0	0	0	0	0	99	0	0	0	0	0	0	0	0	0	0	17
R10 beacon@sophora.fernuni-hagen.de	NA	0	0	0	0	0	0	0	0	98	99	0	NA	NA	0	0	0	0	0	0	0	17
R11 beacon@vivo.rz.rwth-aachen.de	NA	0	0	0	0	0	0	0	0	98	99	0	NA	NA	0	0	0	0	0	0	0	12
R12 beacon@pc302.hrz.tu-darmstadt.de	NA	0	0	0	0	0	0	0	0	84	99	0	0	0	0	0	0	0	0	0	0	NA
R13 beacon@mbone.gris.informatik.tu-darmstadt.de	NA	0	0	0	0	0	0	0	0	84	99	0	0	0	0	0	0	0	0	0	0	NA
R14 beacon@hydra.hrz.uni-bielefeld.de	NA	0	0	0	0	0	0	0	0	98	99	0	NA	NA	0	0	0	0	0	0	0	10
R15 beacon@nx5.HRZ.Uni-Dortmund.DE	NA	0	0	0	0	0	0	0	0	98	99	0	NA	NA	0	0	0	0	0	0	0	10
R16 beacon@mbonepc02.uni-duisburg.de	NA	0	0	0	0	0	0	0	0	98	99	0	NA	NA	0	0	0	0	0	0	0	10
R17 beacon@mc.rrze.uni-erlangen.de	NA	0	0	0	0	0	0	0	0	98	99	NA	NA	NA	0	0	0	0	0	0	0	NA
R18 beacon@PC-R09.netz.uni-essen.de	NA	0	0	0	0	0	0	0	0	98	99	0	NA	NA	0	0	0	0	0	0	0	5
R19 beacon@testsun1-223.rrzn.uni-hannover.de	NA	2	0	NA	0	0	0	0	2	98	99	0	NA	NA	0	0	0	2	0	0	NA	12
R20 beacon@star.hrz.uni-siegen.de	NA	0	0	0	0	0	0	0	0	98	99	0	NA	NA	0	0	0	0	0	0	0	14
R21 test@teleteaching.uni-trier.de	0	15	7	7	5	0	7	7	2	98	99	5	0	0	5	7	12	5	9	12	10	0

Figure 3.21 - Packet Loss Measurement using the Multicast Beacon

The main differences between the MRM and the multicast beacon are the capability of the multicast beacon to provide a direct access to the measurement results and the wider range of QoS measurements (packet loss ratio, delay, and jitter). On the other hand, the MRM allows one to distinguish between a test sender and a test receiver. This differentiation results in a much lower impact on the network, especially if broadcasting scenarios are the most common applications in the particular network under study. Additionally, the MRM allows the use of active RTP streams which reduces the impact on the network as well.

We tested the functionality of the multicast beacon among several universities in Germany. First, the estimation of the delay was examined. The multicast beacon depends on synchronized clocks at each host. Typically, NTP is used in order to achieve a proper synchronization. Only measurement results from such NTP synchronized hosts are provided.

Figure 3.22 shows the measurement of the delay between a host at the University of Erlangen-Nuremberg and three other sites connected to the German research network. The graph on the left shows the delay over the time whilst that on the right presents a histogram of the delay distribution over all recorded samples. All measurements show the same average delay of about 20 ms.

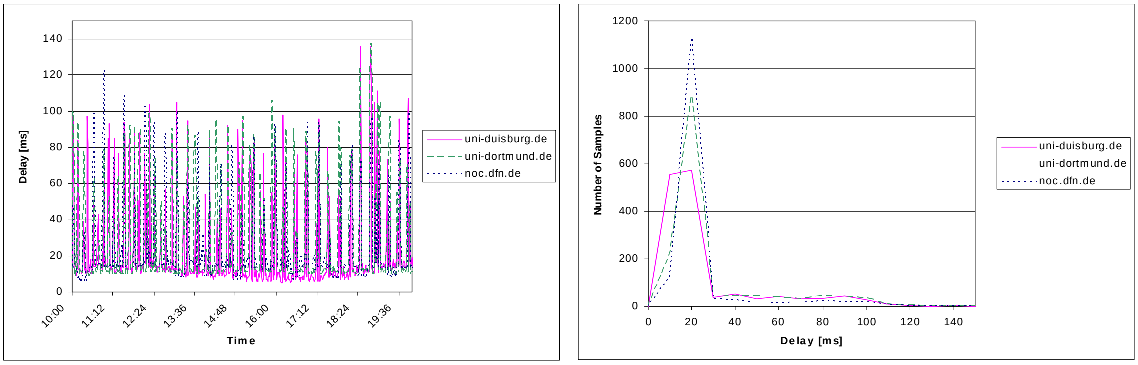


Figure 3.22 - Measurement of the Delay using the Multicast Beacon

Another example is shown in figure 3.23. The connection between Erlangen and Regensburg was tested and the measured delay (about 180 ms) is much higher than the previously shown test results. The University of Regensburg uses an old router to achieve multicast connectivity towards to the G-WiN. This router introduces the high delay. The histogram on the right in figure 3.23 shows a broad distribution of the delay values. Starting at about 20 ms, even a delay of up to 500 ms has been observed. Typically, it is not possible for an application to deal with such a high variation in the delay.

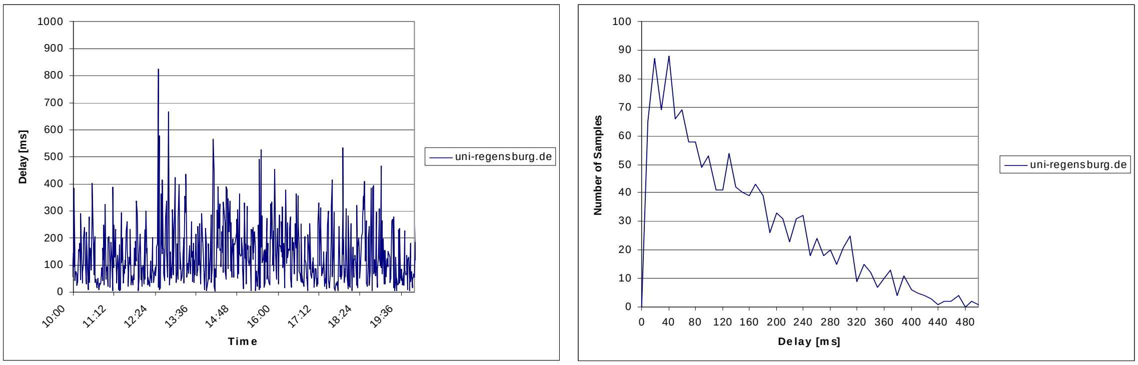


Figure 3.23 - Measurement of the Delay using the Multicast Beacon

Finally, we examined the packet loss ratio between the hosts at the participating universities. Figure 3.24 shows the results of the measurement of the packet loss ratio. Once again, a host in Erlangen sent packets to all other participating clients. It can be seen that only a few packets were lost. Additionally, single peaks can be seen in the figure. Typically, such consecutive lost packets resulted from an overload situation at a single router.

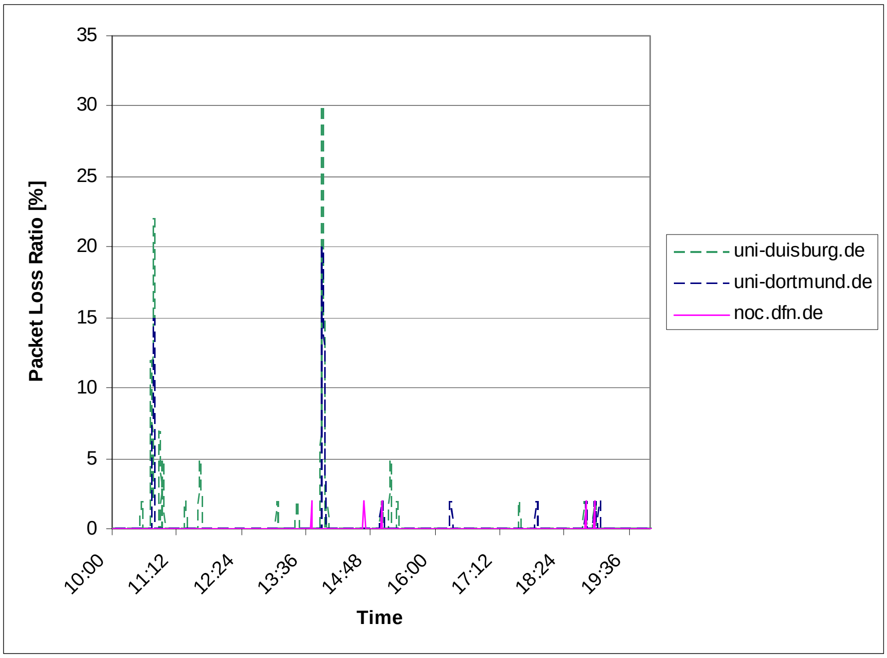


Figure 3.24 - Measurement of the Packet Loss Ratio using the Multicast Beacon

We also examined the packet loss ratio to a host in Regensburg. The measurement of this connection showed a very high loss ratio with an average at about 10%. This loss ratio combined with the high delay variation makes any multicast transmissions between Erlangen and Regensburg impossible.

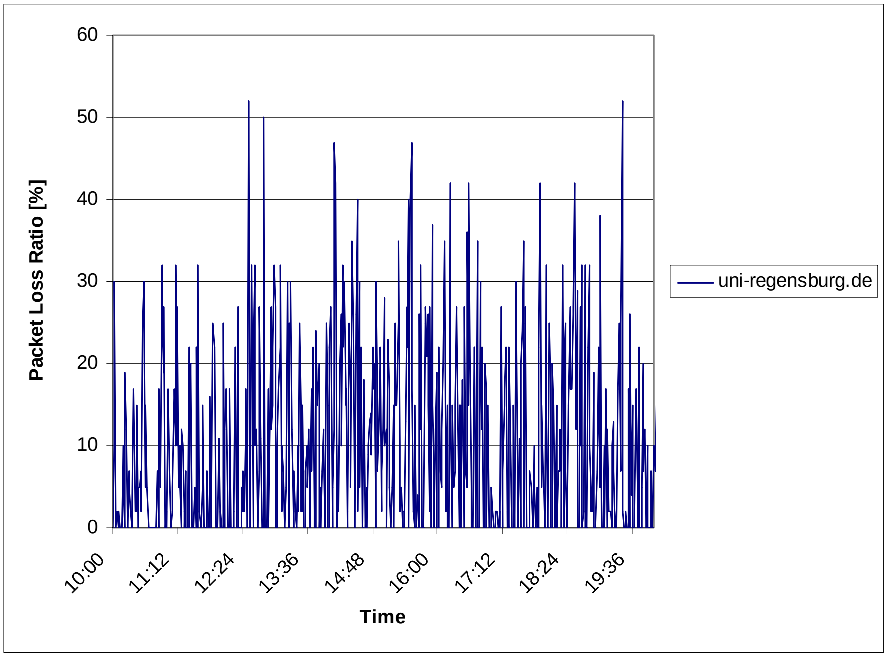


Figure 3.25 - Measurement of the Packet Loss Ratio using the Multicast Beacon

It has been shown that the multicast beacon includes features which allow the measurement of the quality of service parameters such as the delay or the packet loss ratio. The connectivity and the QoS within smaller multicast networks can be estimated.

Unfortunately, the working principle of the multicast beacon does not provide a high scalability because each participating beacon host always sends measurement packets to each other. Additionally, there is a considerable impact on the measured network. The relatively high data rate of sent beacon packets leads to a serious usage of the available bandwidth.

3.5.8 mmon

In contrast to the previously discussed approaches, the mmon (multicast monitor) uses the capabilities of SNMP (simple network management protocol) getting information about the multicast network. The mmon [135] has been developed at HP, Inc. to be included into their network management system HP OpenView. This tool allows one to query multicast specific information from the routers in the network. The goal of this approach is to create a connectivity map of the multicast network and to locate potential failures in the multicast forwarding paths.

Because SNMP is known to have security holes and because most providers do not want to open the configuration of their routers to everyone’s eyes, the multicast monitor is only applicable in a very local environment. Additionally, it requires a running HP OpenView management system which itself is very expensive and resource exhaustive. Therefore, it can be summarized that the mmon is of little importance in measuring the availability and the quality of IP multicast networks.

3.5.9 Summary

The given overview of QoS measurement approaches has shown that there are no single tools available which allow the user to measure all important quality of service parameters at once. This is especially true for IP multicast. Some of the examined tools are based on very good concepts, which should be incorporated into new approaches. Another problem is that most of the existing tools for measurements in IP multicast networks are not scalable for large measurement environments. A new solution for QoS measurements in IP multicast networks is proposed in section 7 "Multicast Quality Monitor (MQM)".

3.6 Providing QoS in IP Multicast Networks

Various working groups of the IETF are working on new approaches to provide or guarantee quality of service parameters to applications using the internet protocol. The trend is conceived to converge all data transmissions to use the same network infrastructure, e.g. IP, even if there are already kinds of networks available including such mechanisms.

One example of QoS enabled networks is ATM (asynchronous transfer mode). ATM is a connection-oriented network. All the resources required are reserved when the connection is being set up. This idea has been taken over by the IntServ WG (integrated services working group) of the IETF. Another approach is used by the DiffServ WG (differentiated services working group). No reservation is performed in order to originate a connection but mechanisms are used to give single packets or packet flows a higher priority. Therefore, DiffServ operates using the typical connection less working principle. Both mechanisms work for unicast and multicast communications and are discussed in the next few subsections.

In addition to the concepts of providing of an amount of quality directly in the network, special mechanisms are being developed to enhance the QoS of IP multicast transmissions [76], [120], [211]. Most of the current multimedia applications already use adaptive algorithms to adjust the quality of the content, for example of a video encoding, to the available quality in the network.

In a multicast environment, the quality of the transmission depends on the worst transmission quality towards all clients. In order to improve this situation, new approaches are investigated. An example is the layered multicast transmission.

Finally, some applications do not use IP multicast because there is no reliable transport protocol such as TCP for unicast. Mission critical applications, for example the transmission of stock rates, require such a reliable protocol. An approach of the IETF to offer a reliable data transfer using IP multicast is the PGM (pragmatic general multicast) protocol.

3.6.1 Integrated Services

The IntServ WG (integrated services working group) of the IETF defined an integrated services architecture (ISA, [32]). The concept of the ISA is to extend the internet architecture and protocols to provide integrated services, i.e., to support real-time as well as the current non-real-time service of IP. This means a measurement-based admission control has been proposed. This is just an extension of the internet philosophy of sending a packet and ´just seeing´ if there is enough capacity [47].

The ISA defines a service model as well as a reference implementation. The implementation reference model for routers is shown in figure 3.26. On the lower side the packet flow through a router is illustrated. The input driver receives the packets from the network interface. A classifier separates the packets to goups or flows. The scheduler provides mechanisms to select packets for forwading depending on a configured behavior of each flow. The processes shown on the upper side, such as the routing agent, the reservation setup agent and the management agent, control the described tasks.

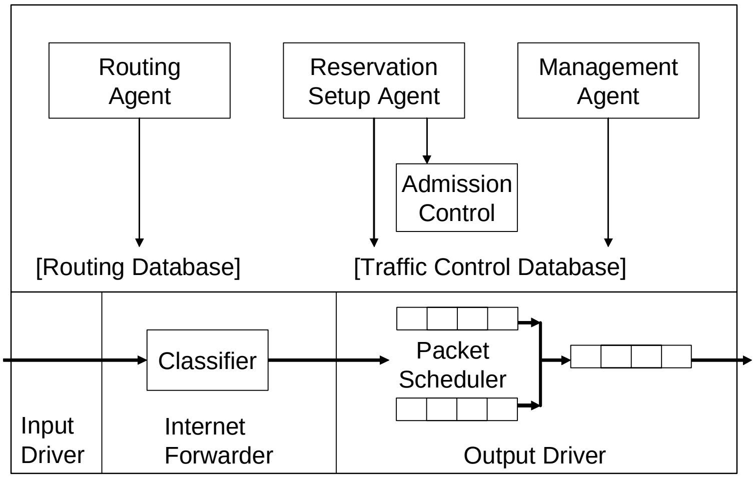


Figure 3.26 - ISA Implementation Reference Model for Routers [32]

The ISA distinguishes between real-time applications and non-real-time applications. Two new services have been defined in order to support the real-time applications: guaranteed load [189] and controlled load [210]. A controlled load service defines an agreement about the required bandwidth of a particular application, i.e. if the network behaves as in a slightly loaded situation. A guaranteed load service goes a little further. In addition to the bandwidth, a maximum end-to-end delay can be guaranteed.

The main result of the work of the IntServ WG is the standardization of a protocol to set up connections with some specific quality of service. This protocol is called resource reservation protocol (RSVP, [33], [209]). The concept of RSVP is to first set up a connection before transmitting data. During the setup phase, all the routers along the path towards to the destination check if the required resources are available and if they are, the routers reserve them. Therefore, RSVP works according to the classic connection oriented principle.

Due to this behavior, many resources are required in the routers along the path to set up and maintain all the active reservations. Scalability is an important factor and it seems that RSVP is not the approach which will work for very large networks with lots of simultaneous connections [133]. But even if it worked in a controlled environment, practically no ISP has implemented RSVP in its core networks. Beside these limitations, RSVP is still the only approach offering a guaranteed quality of service in an IP network.

3.6.2 Differentiated Services

Another approach to increase the quality of service for individual transmissions is the concept of differentiated services, which uses the TOS field (type of service) in the IP header [164] which was specified to differentiate the priority of single IP packets. The single parameters of the TOS field have the following purpose: The precedence bits are used to group packets of the same priority to classes which are processed by routers in the same way. Additionally, bits may be set for high demands to the delay, the troughtput and the reliability. This principle is being standardized by the DiffServ WG (differentiated services working group) of the IETF [27], [90].

The TOS field as shown in figure 3.27 offers the possibility to differentiate IP packets belonging to different types of data. This mechanism can be used to offer more than one class of service (CoS) to the applications. The DiffServ WG recommends some encodings of the TOS field in order to provide classes such as "premium" or "assured". These encodings do not match with the original semantics of the TOS field and are to be used in an ISP backbone network. The TOS field is called DS field (differentiated services field, [150]) at this point. The edge routers perform the task of classifying packet streams belonging to a single service class and set the TOS field according to the common representation in the particular core network.

3 bit	1 bit	1 bit	1 bit	2 bit
precedence	D	T	R	unused

D … Delay (0 = normal delay, 1 = low delay)
T … Throughput (0 = normal throughput, 1 = high throughput)
R … Reliability (0 = normal reliability, 1 = high reliability)

Precedence

000	Routine
001	Priority
010	Immediate
011	Flash
100	Flash Override
101	CRITIC / ECP
110	Internetwork Control
111	Network Control

Figure 3.27 - Composition of the TOS field [164]

Different mechanisms can be used to provide the required quality to the different service classes. Examples are the usage of special scheduling schemes, or a defined behavior in the case of overloading. These mechanisms are named PHB (per hop behavior). The characterization of the assured quality of the different data streams is specified in SLAs (service level agreements). Such SLAs exist between the customer and the ISP as well as between different ISPs.

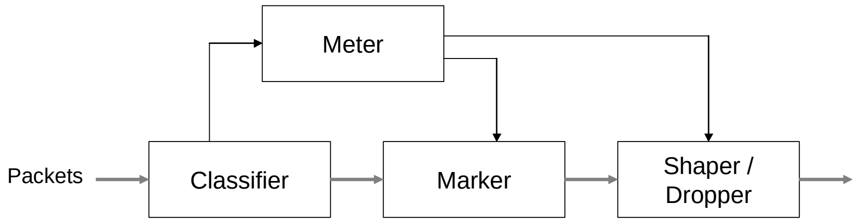


Figure 3.28 - Components of the DiffServ Architecture [27]

The basic components of the DiffServ architecture [27] are shown in figure 3.28. The classifier identifies the arriving packets in order to separate packet streams belonging to different classes. The meter measures temporal properties of the stream, the marker sets the DS field of a packet to a particular code. Finally, the shaper and the dropper try to bring the packet stream into compliance with a specified traffic profile by delaying or dropping particular packets.

The DiffServ concept requires low resources in the network components, so that it is a highly scalable approach. On the other hand, it does not guarantee any quality of service but offers several classes of service. SLAs handle the negotiability of different COSs between customers and ISPs.

Many people believe that DiffServ is the technology to enhance the quality of transmissions in the internet. Right now most of the routers have implemented the DiffServ architecture. In 1999, we tested an implementation for the popular operating system Linux [64], [106]. The principle design is shown in figure 3.29. The basic parts of the network interface are shown on the upper side. The output queuing is enhanced by the DiffServ components. Filters may be used to classify the IP packets in order to apply different queuing disciples to the classified packet flows.

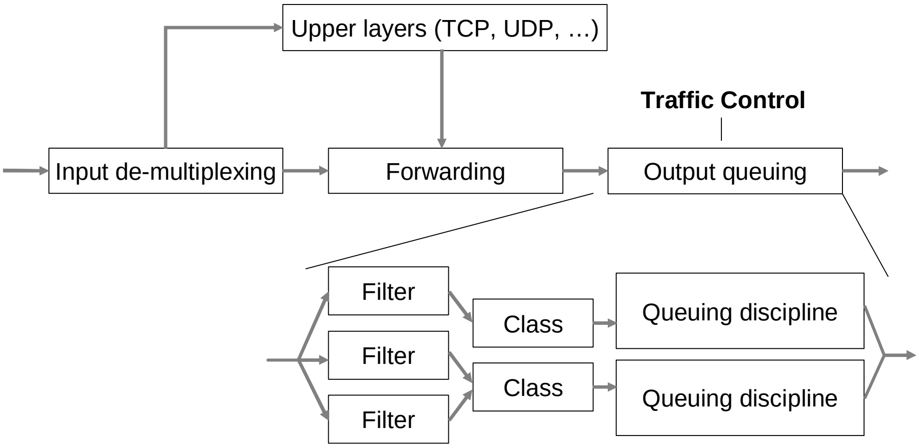


Figure 3.29 - DiffServ Components of the Linux Kernel [14]

The implementation [14], [15] allows one to specify different kinds of filters based on parts of the IP header of incoming packets, e.g. the destination IP address or the TOS field. Additionally, various scheduling schemes are available. We tested standard FIFO queuing against CBQ (class-based queuing). The test environment is shown in figure 3.30. All the connections have at least a bandwidth of 100 Mbps except the outgoing interface of the Linux router to create a bottleneck at this point and to allow the different queuing mechanisms to show effects.

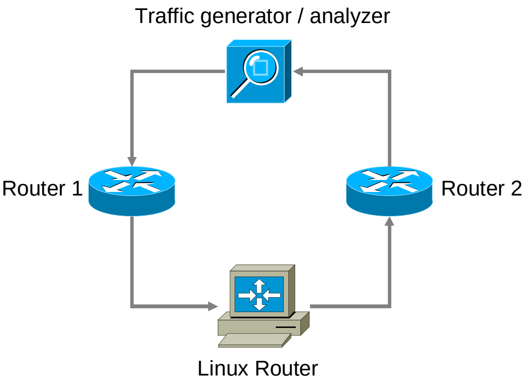


Figure 3.30 - Test Environment for the DiffServ Test using a Linux Router

In order to measure the capabilities of the Linux implementation, we used a Smartbits 6000 traffic generator / traffic analyzer. This tool allows one to configure and analyze multiple parallel packet streams with different properties. We configured four UDP packet flows whereas flow1 has the highest QoS requirements. Flow 4 simulates the best effort background traffic which increases during the tests from 8 Mbps to 50 Mbps. A summary of the definitions is shown in table 3.2.

Flow #	**Packet rate [pps]**	**Packet size [byte]**	**Bandwidth [kbps]**	**Application type**	**CBQ rate [kbps]**
1	33	240	63	Voice over IP	100
2	33	240	63	Audio Streaming	100
3	139	900	1’000	Video Streaming	1’100
4	variable	429	variable	best effort traffic	8’000

Table 3.2 - Definition of the used Packet Flows

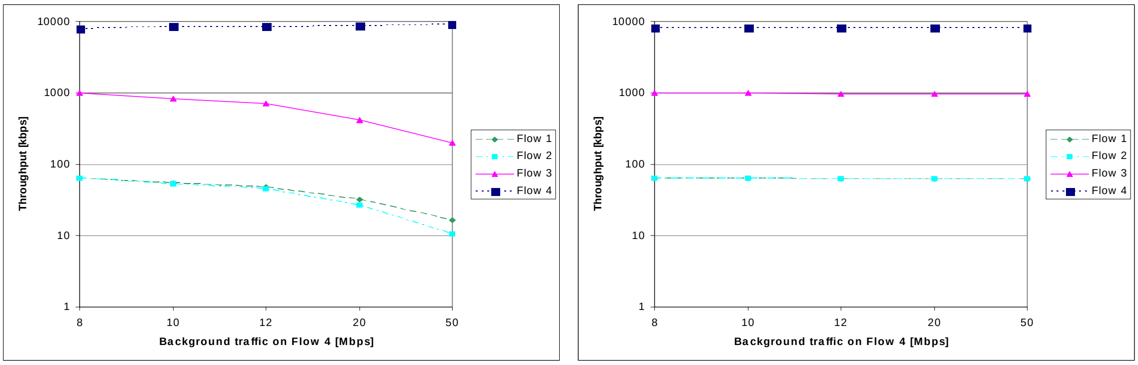


Figure 3.31 - Measurement Results using DiffServ on Linux (left: FIFO; right: CBQ)

On the left of figure 3.31, the results of the FIFO test are shown. It is obvious that the achieved level of quality for the services with higher demands is too low. On the right, the tests results for CBQ are provided. In this case the quality of the multimedia applications was not reduced. The measurement has proved that it is possible to use the differentiated services architecture to provide different qualities for different classes of applications. It is the job of the ISPs to implement the mechanisms in their backbone networks and to provide adequate SLAs to their customers.

Nevertheless, even if different classes of service are available, tools are required to test if the SLAs comply with the achieved quality. Especially for IP multicast, new concepts are required. A new approach for such measurements is presented in section 7.

3.6.3 Layered Multicast Transmission

In addition to the activities designed to enhance the achievable quality of single transmissions, several approaches started in order to build applications which allow the use of the available resources as efficiently as possible [109], [118]. One of the most interesting concepts is the layered transmission of multimedia content. The source encodes the video or audio in different qualities. The receiver, based on the QoS in the network can choose the highest possible one in order to offer the best quality to the user.

A practical realization of this concept is the receiver-driven layered multicast [139]. The server sends copies with different encoding qualities on different multicast groups, thus allowing the client to subscribe to required channels. When using this approach, the clients first start to receive the base information and activate the reception of other layers until all layers are received or lost packets are recognized. In the latter case, the reception of higher layers is stopped immediately.

Using this mechanism, which is shown in figure 3.32, every user gets the best quality which is achievable depending on the available quality of service in the network. The server is required to send all packet flows in order to allow the clients to choose the optimum ones.

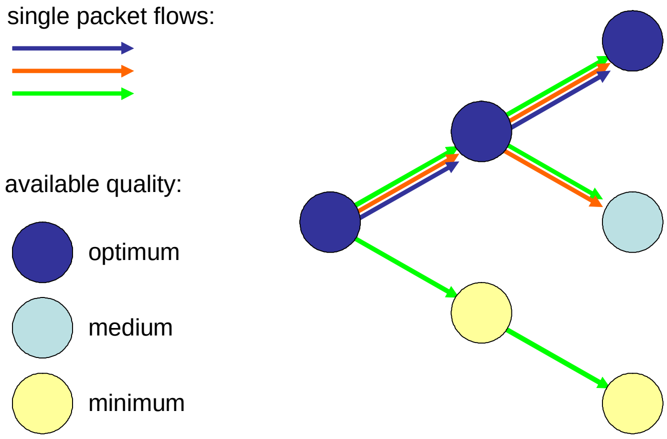


Figure 3.32 - Receiver-driven Layered Multicast [191]

Other approaches allow a communication between the client and the server to inform the server which flows are actually required. While such mechanisms are being used, another problem appears if quite a few receivers are involved, the feedback may result in an overloaded network towards to the server. New concepts of congestion control are a current research topic [87].

3.6.4 Reliable Multicast

Multicast allows an efficient transmission of data from a single source to multiple receivers in the network. The employed mechanisms do not guarantee the correct delivery of the information to each recipient. Most multimedia applications tolerate some amount of lost packets. Additionally, features such as a forward error control mechanism to repair single lost fragments of the data stream are implemented. Nevertheless, applications exist which depend on a reliable service. An example is the transmission of stock exchange rates [136]. Different multicast transport protocols have been developed [16], [46]. A taxonomy of such protocols was created by Obraczka [156].

The concepts used for unicast transmission such as TCP, which provide a fully reliable data transmission do not scale in a multicast environment. A number of mechanisms exist to provide a reliable transfer:

- ACK: The receiver is required to acknowledge the reception of each packet.
- NAK: A negative acknowledgement is sent when a packet has not been received after an amount of time.
- FEC: The packet stream is extended by additional information, allowing the receiver to repair single lost packets.

Additionally, new approaches are in progress to offer a better scalability:

- Hierarchy: ACKs or NAKs are sent upstream using a well-defined hierarchy in order to minimize the messages along the path towards to the source.
- NAK suppression: The receiver waits for a random time before sending a NAK. If a retransmission occurs in this time, the NAK is no longer required.
- Transmission windows: A number of packets may be acknowledged at once.

Criteria have been defined by the IETF to evaluate reliable multicast protocols [134]. The most advanced approach is the pragmatic general multicast protocol (PGM, [190]) which the first implementations are already available for [173]. The PGM uses a conjunction of all the mechanisms mentioned in order to minimize the impact on the network.

4 QoS Requirements of IP Multicast Applications

Even the best tool for the measurement of the current quality of service in an IP multicast network is useless, if the requirements of the applications are not known exactly. Additionally, the proposed approach, shown in section 7, requires knowledge of the expected behavior of both active and scheduled services. An analysis of the requirements of each application is provided in the next subsection. This is followed by the description of the test environment, which has been developed and installed at the University of Erlangen-Nuremberg in order to analyze the quality of service requirements of particular applications. Finally, measurement results are discussed.

4.1 Analysis of the Requirements of Applications

Numerous articles have been published concerning the quality of service requirements of single applications. The goal of this subsection is to give an overview of some of these papers as well as to provide a basis for the evaluation of QoS measurements in IP multicast networks.

4.1.1 General Taxonomy of Multicast Applications

The general taxonomy of multicast applications examines the requirements of different kinds of services. Additionally, an overview of the demands of various applications is provided. The bandwidth and synchronization requirements as well as the delay and the loss tolerance are investigated. According to RFC 3170 [166], the following selected multicast applications are analyzed.

(1) Scheduled audio/video distribution: One or more data streams are used to broadcast lectures, presentations, or meetings to an audience. Typically, such services need a high bandwidth in order to deliver high quality video signals to the receivers. Synchronization between the streams is required. Additionally, different priorities may be assigned to these different data streams. For example, it is more important to ensure an intelligible audio stream than a perfect video.

(2) Push media: The broadcast of news headlines or stock exchange rates is characterized by low bandwidth requirements, but a high reliability in transmitting such data is required.

(3) Announcements: Network time and multicast session schedules are examples of announcements. The requirements of these services vary, but, typically, such applications show high tolerance levels with regards to lost packets and a high delay.

(4) Multimedia conferences: An audio/video conference, possibly including whiteboard applications and others, has similar quality requirements as the scheduled audio/video distribution. Furthermore, coordinating issues such as determining who gets to talk at which time have to be implemented [202].

(5) Collaboration: Examples of collaboration tools include a whiteboard application or the editing of shared documents. The resource requirements vary strongly depending on the kind of data to be exchanged. Typically, collaboration tools tolerate the delay but they require a medium or high bandwidth between all participants.

(6) Resource discovery: Service location protocols and address selection mechanisms are examples of resource discovery tasks.

(7) Accounting: The collection of data is an example of an accounting mechanism. In some cases this has to be done in real-time. If many systems try to send collected data to a single host or to each other, this information can be overwhelming. Mechanisms to provide a proper scaling for such environments are required.

4.1.2 Bandwidth Requirements

Figure 4.1 shows the approximate bandwidth requirements of multicast applications. Especially the multimedia sessions have much higher demands. Typically, a video transmission works at a higher data rate than, for example, the transmission of session announcements.

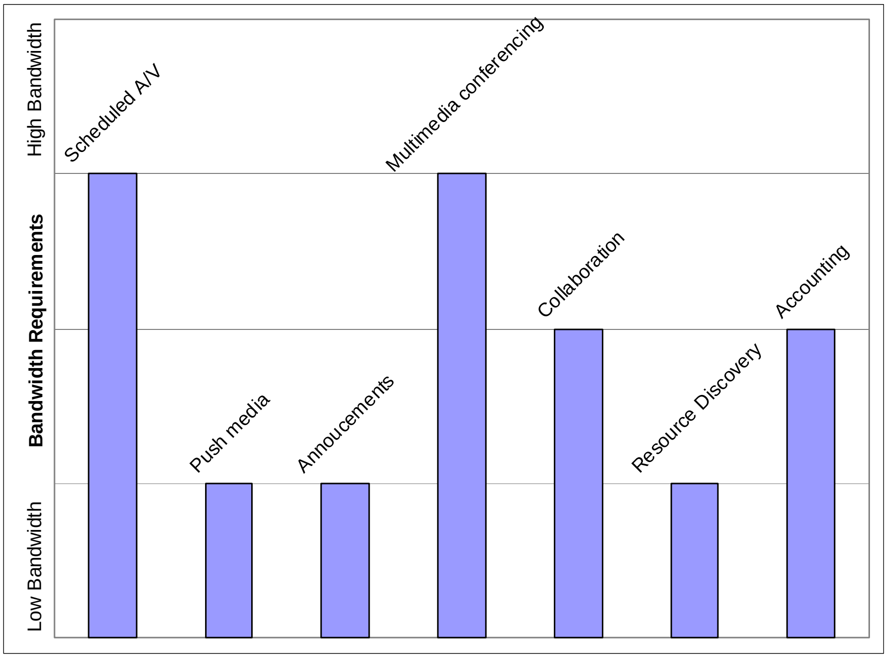


Figure 4.1 - Bandwidth Requirements of Multicast Applications

The bandwidth requirements of selected audio and video encodings are summarized in table 4.1.

Encoding	min. Througpout [bps]	max. Throughput [bps]
MP3 (audio)	64k	240k
PCM (audio)	64k	2M
H.261 (video)	64k	1M
MPEG1 (video)	1M	3M
MPEG4 (video)	64k	10M
MPEG2 4:2:0 (video)	1M	15M
MPEG2 4:2:2 (video)	15M	50M

Table 4.1 - Data Rate of Selected Audio/Video Encodings

Typically, multicast receivers have to accept traffic from more than one source. Consequently, multiple data streams, possibly at a high data rate, have to be received and analyzed. Additionally, the encoding quality selected at the sender has to be accommodated to the worst possible quality to each receiver, except for the usage of mechanisms like the layered multicast transmission.

4.1.3 Delay Tolerance

Most applications show a reasonable tolerance to an increasing delay. The only exception is a bidirectional communication. It has been shown that a maximum delay of 200 ms is admissible in order to allow a suitable conversation [36]. Figure 4.2 summarizes the delay tolerance of selected multicast applications.

Even if the absolute delay is less important to most applications, it should not exceed some application-dependent values. If the delay increases too much timeouts may appear, which make the transmission inoperative. Apart from the absolute delay, the delay variance or "jitter", is of interest to real-time applications. As these typically have an implemented buffer, the jitter can be reduced to an absolute delay, depending on the size of the buffer, or a higher packet loss ratio if the delay varies over the available buffer size.

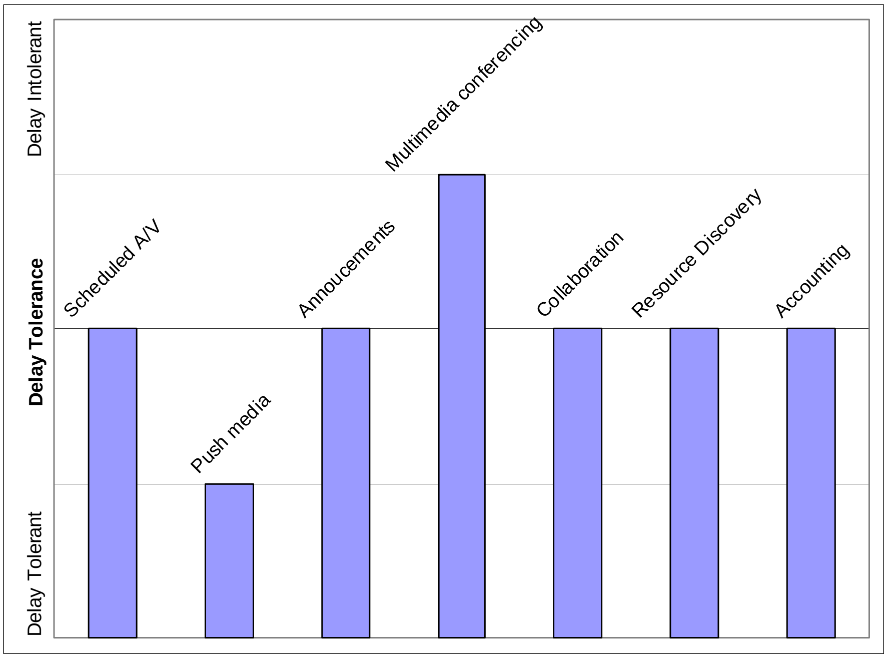


Figure 4.2 - Delay Tolerance of Multicast Applications

4.1.4 Loss Tolerance

Many multicast applications use loss tolerant encoding schemes. Therefore, the delivery of the signal remains useful, even if some of it is lost. First of all, most multimedia transmissions implement features to survive some lost data. For example, audio might have a short gap or lower fidelity but will remain intelligible despite some data losses.

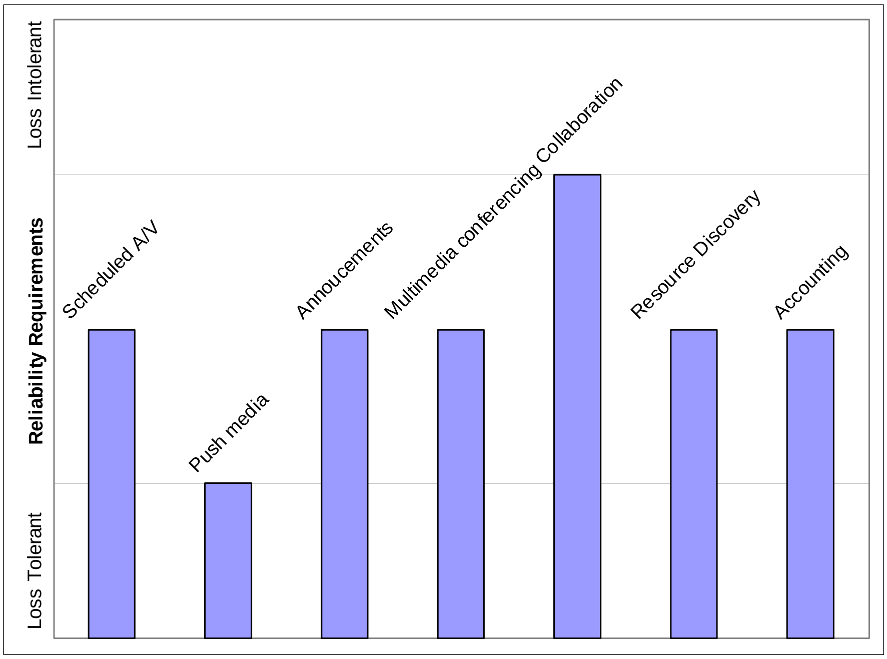


Figure 4.3 - Reliability Requirements of Multicast Applications

Figure 4.3 summarizes the loss tolerance of some multicast applications. For non-multimedia traffic, the reliability of the data transfer may be increased by using a reliable multicast transport protocol such as PGM. Such reliable transport protocols ensure the proper transmission of each packet and, if a packet is lost, its retransmission. Although the reliability is being increased, the delay is increased as well. Therefore, reliable transport protocols cannot be used for delay sensitive applications.

4.1.5 Synchronization Requirements

Multimedia services typically contain more than one data stream. Generally, at least two different transmissions exist for audio and video signals. These data streams have to be synchronized at the receiver.

Because the primary recipient is a human being, no strong synchronization requirements have to be applied. A human being does not perceive slight concurrency. Nevertheless, the adherence of time limits is significant for the quality of the received multimedia data.

Medium		**Mode, Application**	**QoS**
Video	Animation	correlated	+/- 120 ms
	Audio	lip synchronization	+/- 80 ms
	Picture	overlay	+/- 240 ms
		non-overlay	+/- 500 ms
	Text	overlay	+/- 240 ms
		non-overlay	+/- 500 ms
Audio	Animation	event-correlated	+/- 80 ms
	Audio	closely coupled (stereo)	+/- 11 µs
		loosely coupled (dialog of participants)	+/- 120 ms
		loosely coupled (background music)	+/- 500 ms
	Picture	closely coupled (music with notes)	+/- 5 ms
		loosely coupled (slide show)	+/- 500 ms
	Text	audio commentary	+/- 240 ms
	Pointer	audio in context to the pointed item	-500 / + 750 ms

Table 4.2 - Synchronization Requirements between different Media [191]

Table 4.2 shows the synchronization requirements between different kinds of multimedia content. The greates synchronization requirement is necessary for lip synchronization. To watch a particular video transmission can be difficult, if the audio and video stream are highly unsynchronized, i.e., if they are more than 80 ms apart. Approaches to achieve better results in lip synchronization are in progress [124].

Other typical examples include the synchronization between a video signal and some text information, and between audio and single pictures of a slide show. The time limit for overlaid text, e.g. subtitles to a movie, is 240 ms. For a slide show, a maximum gap of 500 ms can be considered.

4.2 Test Environment and Evaluation

The behavior of the applications can only be achieved by end-to-end measurements [1]. In order to test the QoS requirements of various applications, it is necessary to set up a test environment and to define techniques to evaluate the results. Starting with the latter requirement, it has been shown that there are two distinct kinds of measurements [65]: The first one is an objective test. The results of such a test are taken and analyzed by special hardware and software which has been designed just for these tests. The second one is a subjective test. It is often impossible to have such a special measurement environment available and it is necessary to include the subjective impression of the end users, which makes applications evaluation by human beings necessary.

Even if the application is intended for usage in the internet, it has to be questioned if a test in this network is really complete. A number of uncertainties always appear in this environment. For example, the utlization depends on the number of users which simultaneously transmit data. The results become more useful if it is possible to build a test environment in a lab. Such an environment allows the manipulation of single parameters in a well-defined form. Examples of parameters to be modified are the jitter and the packet loss ratio. To perform such tests, we have developed an IP based impairment tool, which allows the manipulation of various aspects of the network behavior.

4.2.1 Objective Tests

As previously mentioned, special devices are required in order to achieve objective measurement results. Unfortunately, such devices are only available for a limited number of application scenarios and even if one exists, of course they are typically very expensive.

An example of such a measurement tool is the RADCOM VoIP (voice over IP) analyzer. This tool is a software enhancement for the RADCOM network analyzer. This network analyzer can be used like any other network sniffer. It records all the received packets in order to allow an analysis of the data traffic. The RADCOM hardware can be attached to different kinds of network interfaces. We used the device with an ethernet interface. The recorded packets can be analyzed in order to get more information about the received network traffic. In particular, the voice over IP traffic can be analyzed to evaluate the perceived jitter, the packet loss ratio and the amount of reordered packets.

Similar tools are available to analyze other kind of data such as video traffic. Another solution is based on the applications themselves. Sometimes, the application developers include facilities to evaluate the perceived quality of service. For example, the popular MBone tools such as vic and rat show parameters such as the current packet loss ratio to the user.

4.2.2 Subjective Tests

Even if objective measurements generally provide more accurate results, they have to be combined with subjective tests in order to evaluate the results depending on the requirements of the end user [31]. Additionally, subjective tests are the only solution if no measurement devices for an objective analysis are available.

Subjective tests are much sophisticated than one might expect. The capabilities of different people to view a specific signal and to describe its quality vary extremely. Typically, a group of different persons perform the test individually. Then, the single results are compared in order to discover the final solution.

Additionally, the signal source has to be carefully selected. For example, if the quality of an audio transmission has to be analyzed, two types of audio sources should be separately tested: voice and classic music. Even inexperienced testers are able to produce comparable results using these audio transmissions.

4.2.3 Real World Measurements

Most measurements are started in a real world environment. This means that the enterprise network or the campus network is used to deploy the required applications as well as to test their behavior. The results of such tests strongly depend on the current usage of the network. Even if the tests are repeated at different times to evaluate the applications behavior in a heavily used network as well as in an optimum environment, the results cannot be used to provide a general overview of the QoS requirements of the tested application.

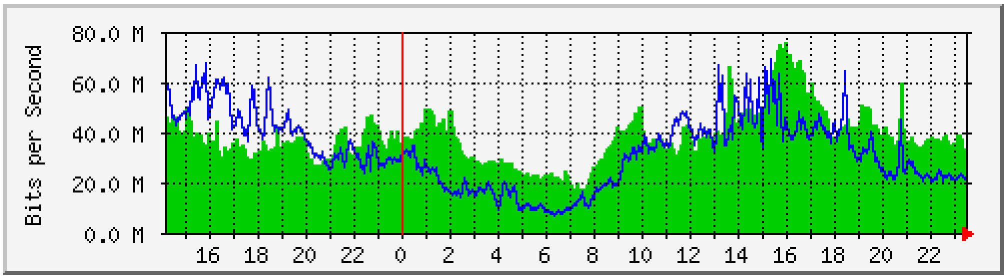


Figure 4.4 - Load of a Network Interface depending on the Time of Day

An example for the distribution of the load of a typical interface of a router over a period of time (24 h) is shown in figure 4.4. The green (filled) curve is the input ratio and the blue (line) curve is the output ratio.

If applications which depend on a minimum quality of service of the network are tested it is nearly impossible to find these minima using a production network such as the real internet. Nevertheless, it is a generally good approach to use the real world scenario in order to show the proper working of the particular application in this environment. Additionally, the unpredictable behavior of the campus network or the global internet allows one to test the application's behavior in a more comprehensible way. The exactly behavior and the specific quality requirements of the particular application can only be analyzed using a controlled lab environment.

4.2.4 Lab Tests using an Impairment Tool

Lab tests are used to test the application's behavior in a well-defined environment. Typically, the tests are started with an optimum network. Such a network only consists of devices which are really required to test the application. During the next test periods, single parameters of the lab network are manipulated. Examples of such variations are the degradation of the available quality of service, such as a decreasing bandwidth or an increasing delay, or the interference of the network by applications started in parallel. These single tests can then be combined in order to get more meaningful results.

Typically, the network's behavior is modified by using a so called impairment tool. Such an instrument allows the manipulation of different quality of service parameters of the network. At the very least, the tool should allow the introduction of an additional delay, and modification of the packet loss ratio. An impairment tool should only work on a configurable packet stream in order to manipulate the networks behavior for a particular application.

4.2.4.1 Requirements for an IP Impairment Tool

Since the first lab for quality of service measurements of different applications at the University of Erlangen-Nuremberg was built, it has been shown that there are only few devices available which implement an impairment functionality. These components are very expensive. Nevertheless, hardware monitors are required to achieve an extremely high accuracy [116].

The first impairment tool used was a GN-Nettest interWATCH 95000, which is a tool operating on ATM networks. In order to use this device for measurements in an IP based network, a conversion between IP and ATM is required. The environment for such tests is shown in figure 4.5. Unfortunately, additional effects are introduced by the two routers, which are only included to convert IP packets to ATM cells back and forth in order to allow the impairment tool to manipulate the behavior of the IP network. Even if the routers are only slightly loaded, at least a small additional delay and a small jitter due to queuing effects are introduced.

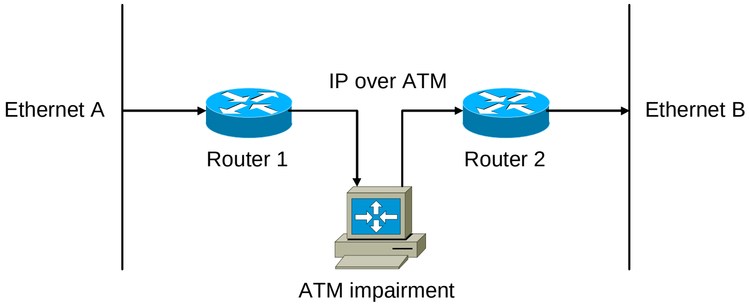


Figure 4.5 - Using an ATM based Impairment for IP based Tests

The impairment has been designed to manipulate ATM networks in order to examine ATM based applications. Because such applications are generally real-time services, they depend on a very high quality of service in the network. Therefore, the ATM impairment was designed to modify the behavior of the network over a very small range. For example, it is not possible to introduce a delay of up to one second. IP based applications are much more tolerant to a slightly reduced quality of the network, therefore the tool is of only little use for measurements of IP based applications.

All these restrictions lead to an increasing demand for new tools. Therefore, we started our own attempt to develop an impairment tool at the University of Erlangen-Nuremberg. The new tool is designed for the usage in IP networks and it should:

- include support for various kinds of local networks, at least for ethernet,
- run on a cheap hardware, preferably on a standard PC, and
- it is required to be easily configurable and maintainable.

Figure 4.6 shows the new scenario. No "converters" for different kinds of local area networks are required.

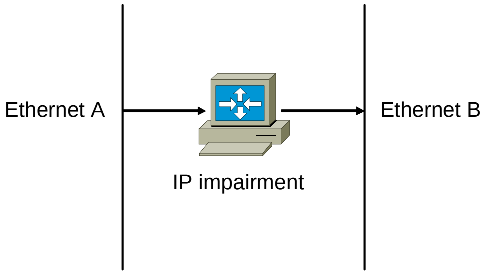


Figure 4.6 - IP Impairment

In order to modify the behavior of the network, the IP impairment tool should be able to modify the packet stream by:

- introducing a static delay to each packet,
- introducing a variable delay based on a configurable probability function, such as a Gaussian function, in order to increase the jitter of the transmission, and
- increasing the packet loss ratio by dropping packets. The selection of the packets should also be based on a configurable probability function.

The impairment tool must not modify, duplicate, or change the order of any packets.

4.2.4.2 Developing an IP Impairment Tool

We developed an IP based impairment tool, named DNDF (dummynet with distribution function), based on the dummynet driver [172] of the FreeBSD operating system. The implementation was undertaken by a student during his master thesis.

The dummynet driver already allows the manipulation of a given packet stream in order to introduce a static delay or to increase the packet loss ratio by configuring a loss priority. The concept of dummynet is using the firewall code of FreeBSD in order to separate packets of individual streams. These packets are put into different queues. The standard FIFO scheduler of FreeBSD is enhanced by dummynet to add a configurable impairment to each queue. If a delay is to be included, all the packets of this queue are delayed by using internal timestamps. In the case of a required non-zero packet loss ratio, an internal probability function is used to mark each packet as either to be dropped or forwarded. An overview of the configuration of dummynet is provided in appendix C.1.

Our implementation enhances the dummynet code to modify its delay function as well as its function to select packets for forwarding or for dropping. This is done by adding new kernel-internal tables to perform a quick lookup for probability values calculated by a separate user-level tool. Using this probability function, it becomes possible to include a variable delay or a predictable behavior of the packet loss functionality. Information about the configuration of the new DNDF module can be found in appendix C.2.

4.2.4.3 Verification of the IP Impairment Tool

In order to validate the functionality of the IP impairment tool, a special lab environment was created. A Smartbits 6000 traffic generator / analyzer was used to create packets, send them out of one interface, and receive them at a second interface. This test setup is shown in figure 4.7. Using timestamps and sequence numbers, it is possible to measure the delay of each packet and to detect lost packets.

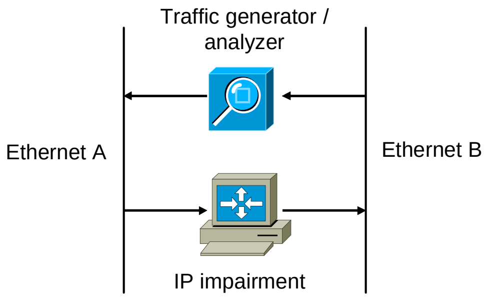


Figure 4.7 - Lab Setup to Test the IP Impairment Tool

For all the tests we configured several packet flows with a similar behavior as known for typical applications in the internet. An overview of the used flows is provided in table 4.3. The complete configuration of DNDF is shown in appendix C.3.

Flow #	**Packet rate [pps]**	**Packet size [byte]**	**Bandwidth [kbps]**	**Application type**
1	34	240	64	Voice over IP
2	68	300	160	MP3 Streaming
3	93	1.100	800	MPEG4 Streaming
4	596	1.100	5.000	MPEG2 Streaming
5	3055	429	10.000	best effort

Table 4.3 - Definition of the Packet Flows used for Testing DNDF

The complete summary of all the test results can be found in appendix C.4. At this point, only a short overview of these results is provided. Additionally, appendix C.4 includes some reference measurements. These tests are required in order to evaluate the measurements of the impairment functions.

We started the tests of the impairment tool by configuring dummynet to introduce a constant delay to all forwarded packets. The results of the measurement are shown in figure 4.8. The requested delay of 20 ms results in a complete transmission delay between 19.5 ms and 21.5 ms. The resulting delay strongly depends on the packet size. The forwarding of larger packets requires much more time in all involved components. Therefore, the minimum additional delay can be easily explained and is completely tolerabled.

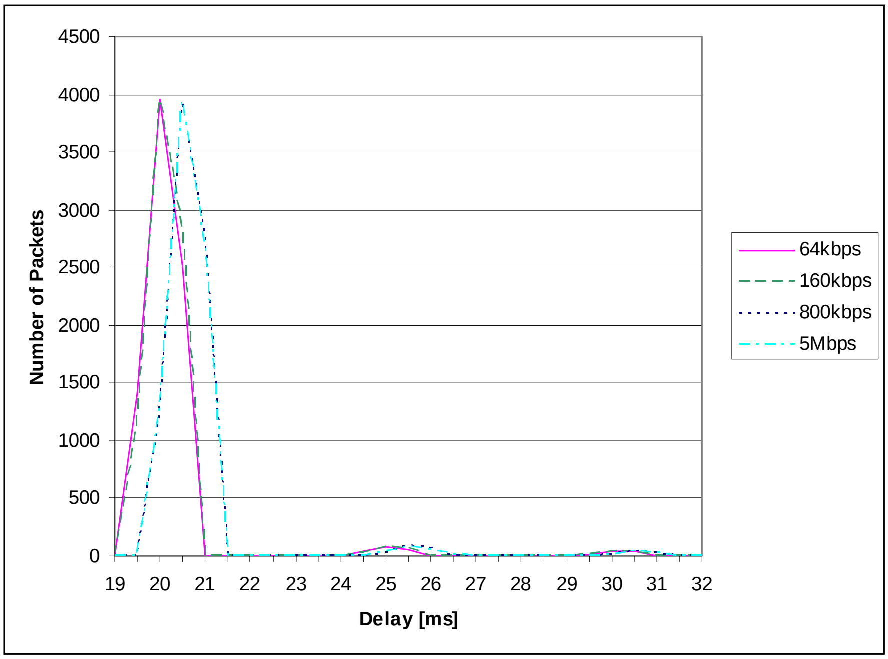


Figure 4.8 - Constant Delay of 20 ms

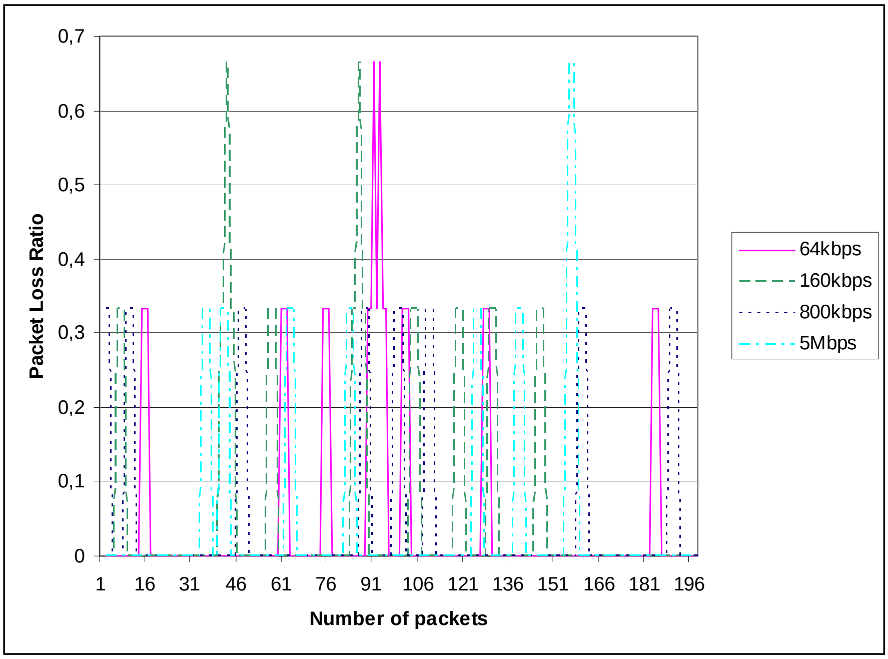


Figure 4.9 - Packet loss of 5%

The second function of the IP impairment to be tested was the introduction of a configurable packet loss ratio. Dummynet allows to independently configuring a particular packet loss ratio for each flow. Figure 4.9 shows the measured loss ratio for a preconfigured value of 5%. The average loss for each flow is exactly 5%. Nevertheless, the selection function of dummynet which is responsible for choosing the single packets which have to be dropped has not been perfectly implemented. In each case, consecutive packets are discarded instead of single packets. The resulting behavior can be only used for streams with a high data rate such as the examination of the behavior of video transmission applications.

Finally, DNDF was configured to produce a variable delay. We prepared a table containing values for a Gaussian distribution function with a minimum delay of 0 ms, a maximum delay of 500 ms, an average of 150 ms with a variance of 50. This table was used to introduce a jitter to the tested packet flow. The resulting distributions of the delay are shown in figure 4.10. The histogram shows that the impairment works well for low data rates.

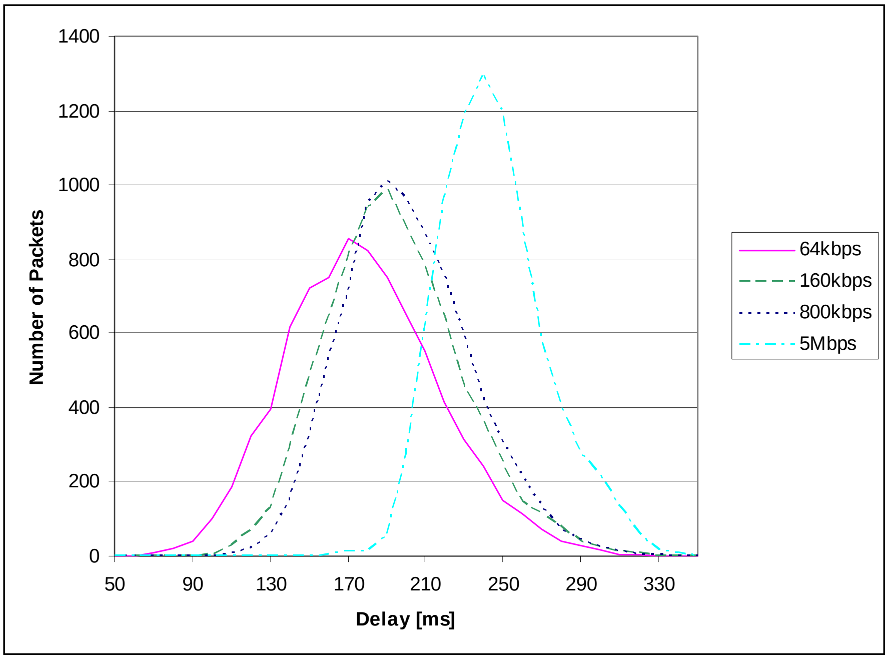


Figure 4.10 - Variable Delay (min 0 ms, max 500 ms, avg 150 ms, var 50)

The resulting graph for the 5 Mbps video transmission is shifted to a higher average delay. The primary reason for this shift is the additional queuing delay in all the involved network components. The higher the packet rate, the higher the average queue length is. Additionally, the impairment tool is required to keep the order of the packets. Therefore, it is possible that if a first packet has to be delayed for a long time, a second one cannot be sent at its calculated delivery time. It has to be delayed until the first packet has left the system.

An example of such an effect is shown in figure 4.11. A first packet arrives at t_{A0}. The delay function selects a value of dt_0 for this packet. Therefore, it will be sent at $t_{D0} = t_{A0} + dt_0$. The second packet arrives at t_{A1} and will be sent at $t_{D1} = t_{A1} + dt_1$. Finally, the third packet has to be sent at $t_{D2} = t_{A2} + dt_2$. Unfortunately, t_{D2} is smaller than t_{D1}. Therefore, this packet will also be sent at t_{D1} including an additional delay of $dt_{plus} = t_{D1} - t_{D2}$.

A solution for this would be an adaptive algorithm for the calculation of the delay for each packet. The error of the last packet can be evaluated in order to modify the result of the delay function for the following packet. For example, the calculated delay can be reduced by a percentage degree of the error.

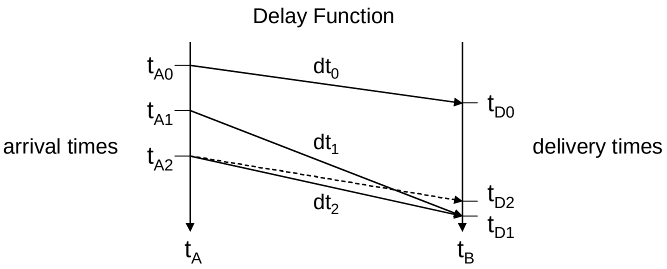


Figure 4.11 - Increasing Delay due to the retaining Packet Order

To sum up, it can be said that the working range of the IP impairment is good enough to test the behavior of most multimedia applications in an IP network. If very small modifications to the transmissions behavior have to be applied, other tools such as the ATM impairment have to be used. Typically, there is no need for these kinds of tests in an IP network, because routers and switches already introduce a much higher degree of variation and delay.

4.3 Measurement Results

Two similar measurements are described in the following subsections. First, we have examined the behavior of a typical video transmission using the popular MBone tool vic. Secondly, the quality requirements of a voice over IP channel were evaluated. The acquired results apply to almost every audio and video transmission over IP.

4.3.1 Video Conferences

The analysis of the video quality requirements of a video transmission, using the video tool vic, was performed in a special lab environment, shown in figure 4.12 [63].

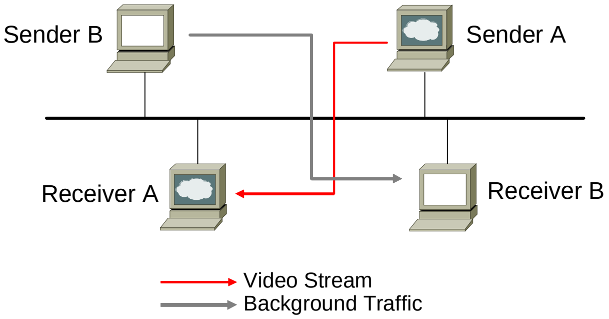


Figure 4.12 - Test Environment to analyze the QoS Requirements of a Video Transmission

A host (sender A) is transmitting a H.261 encoded video to another one (receiver A) in the same shared ethernet (10 Mbps capacity). The configured data rate of the video is 500 kbps. The tool vic provides capabilities to present the available quality by displaying the current packet loss ratio and the current jitter. Two other hosts in the same network (sender B and receiver B) are used to generate an increasing UDP background traffic in order to interfere with the video conference.

At a background traffic rate of about 8 Mbps the jitter of the video transmission increased dramatically. Due to this high delay variation, the packet loss ratio increased as well. Therefore, the video transmission becomes inoperable at a utilization of the shared ethernet of about 80%. An example of the resulting quality is shown on the left side of figure 4.13. Also shown in the same figure (on the right side) is a graph displaying the number of received packets. It can be seen that there are at least two points where the ratio of received packets has been decreased substancially.

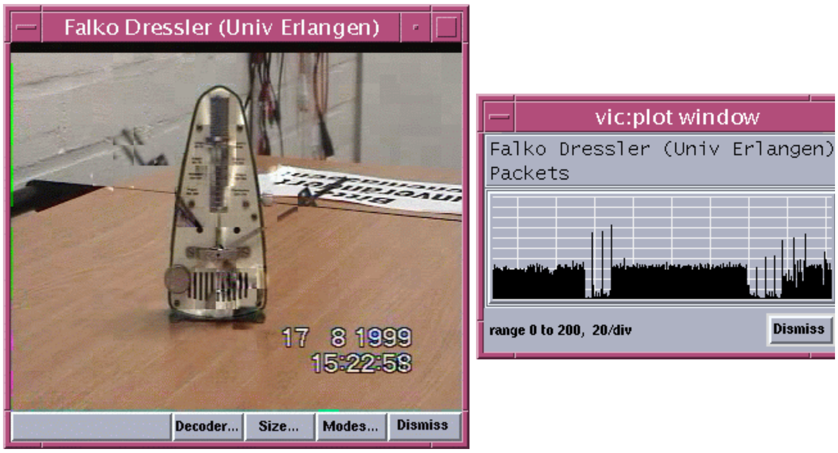


Figure 4.13 - Screenshot of the vic tool while perceiving a high Loss Ratio

It should be admitted that the adaptive algorithms of vic try to increase the quality by reducing the used bandwidth for the video. Nevertheless, this adaptation can only partially solve the problem. Capabilities of the network are required in order to provide a higher end-to-end quality to the application.

4.3.2 Voice over IP

In order to analyze the quality requirements of a typical voice over IP transmission [49], the test setup as shown in figure 4.14 was built [65], [69]. We used the ATM based impairment tool GN-Nettest interWATCH 95000 to modify the network's behavior. The RADCOM VoIP analyzer was used to objectively measure the quality of the voice transmission. Additionally, we evaluated the resulting quality of the audio channel by transmitting classical music as well as using the channel for a speech transmission.

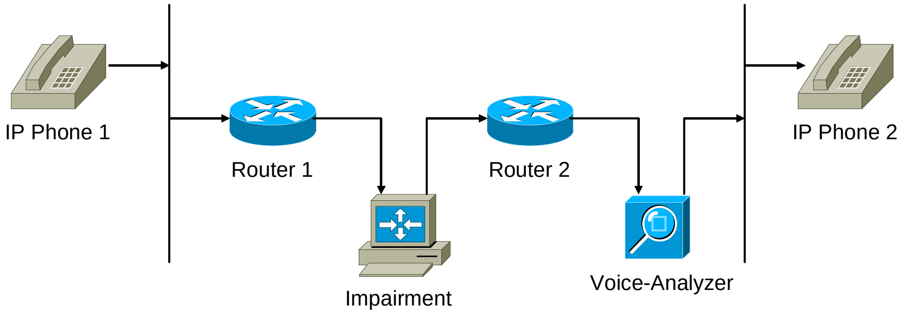


Figure 4.14 - Test setup for the VoIP Tests

In a first reference measurement, the achieved quality of the voice transmission was very high. When introducing a constant delay, we found no reduction in the achieved quality, but at about 200 ms, the interactivity of a conversation was lost.

In the next test, we configured the impairment tool to create a variable delay in order to increase the jitter of the transmission. It has been shown that a low jitter is intercepted by a play-out buffer at the end system, whereas high jitter values have the same effect as lost packets. They are dropped at the end system if they arrive too late.

ATM Cell Loss Ratio	Packet Loss Ratio [%]	Subjective Impressions
0	0	good
10-4	0.069	good
10-3	0.785	degraded quality
10-2	8	unintelligible music, degraded quality of speech
10-1	72.3	unusable

Table 4.4 - Measurement Results of a increasing Packet Loss Ratio

As shown in table 4.4, the quality of the voice transmission decreases rapidly if the packet loss ratio increases. Even at a loss ratio of only about 0.8%, the receiver quality is worse. At 8%, the connection is no longer usable for audio transmissions.

The measurements prove that some steps are still required before initiating important multimedia transmissions:

- the required QoS of the application has to be tested,
- the currently provided quality in the network has to be analyzed, and
- new mechanisms to guarantee a minimum transmission quality have to be used.

5 Modeling IP Multicast Networks and Services

The motivation for quality of service measurements has been discussed in the previous sections. Where an overview of currently available measurement tools for reachability measurements as well as for examination of the available quality of service in IP multicast networks was provided. Most of these applications use RTP encoded packets streams. There are several reasons for this. First, these streams can be initiated by the measurement tools themselves. Secondly, the application can join active multicast sessions and examine the received packets in order to calculate information about the current quality from the originator of the packets towards to the measurement tool.

One of the most important problems to be discussed covers the increased network congestion: The more measurement instances or "probes" are implemented in the network, the better the estimation of the functionality of the multicast network. On the other hand the congestion of the network increases due to the measurements themselves. Therefore, the network will be overloaded with measurement traffic, which reduces the quality of other multicast applications.

Finally, the behavior of the network and the available quality is not predictable if insufficient probes are installed, or, if the measurement tools are inadequately distributed.

5.1 Motivation for the Model

In order to design a tool to measure the reachability, reliability and quality of service of IP multicast networks, the following topics must be addressed:

- What are the optimum places within the network to implement the probes? It is necessary to consider the active multicast services which can be directly used for measurements of the connectivity and the quality of service. Additionally, information about the physical, and possibly, the logical network have to be gathered. In order to predict the usability and the expected quality, a comprehensive knowledge about the intended multicast applications is required, including information about their resource and quality requirements.
- Which information can be gathered? A description of the parts of the network which are actually used for the most important multicast applications is the basis for the deployment of any measurement equipment. Generally, it is not possible to measure or analyze the internet as a whole.
- Which measurements are really required? For example, if high bandwidth video transmissions are scheduled, simple reachability tests are practically useless. Additionally, a reasonable coherence should exist between the measurements and the intended applications (possibly by simulating the transmission of high bandwidth data streams which may have a high impact on the network).

- What is the best time to start particular measurements? Temporal factors should not be neglected. Many very significant test series of the quality of a multicast network may be extracted by a passive eavesdropping of active multicast sessions. Typically, such sessions are subject to time limits. For example, conferences typically last for a few minutes up to several hours. Video transmissions like TV broadcasts show a different behavior. Such applications tend to transmit almost permanently into the network.

In the following section a model is introduced, which allows to deploy measurement tools in light of the topics mentioned above. The first version of the model was presented by Dressler [66]. Based on the collected information, it is possible to predict the expected quality of service for particular services and applications. The information can be gathered from the model or by explicit measurements. An implementation shows the functionality of the model based on concrete or fictitious networks.

5.2 Object Model

A fundamental objective of the model is to associate information about the multicast network and about the most important applications and services. The presented object model bears the working title MRT (multicast routing tool). The following goals provide the fundamental construct of the model:

- consistent object oriented specification
- abstract presentation of multicast networks and services
- minimal complexity
- complete description of all relevant systems
- inclusion of all important operating characteristics and states
- independency of particular measurements tools

After an initial gathering of the network structure and the operational characteristics, it should be possible to forecast an optimum path through the network and to predict the available quality of service. First statements should be feasible in order to allow the design of a network even if concrete measurement results about the behavior of the network are still missing. Of course, the more measurements accomplished and the more operational characteristics gathered, the better calculations can be conducted.

The independency of the model to measurement equipment and to any data mining tools allows the usage for simulations as well.

Basically, the model consists of two classes of objects:

- multicast networks (using a layered structure, based on the TCP/IP model)
- multicast services (based on the typical requirements of the services)

Routing algorithms are required to calculate optimal paths for a particular service through the network. Depending on the capabilities of the employed algorithm, the current state of the network as well as the utilization and the available quality of service may be included into the computation. Due to the basic goals of the model, various algorithms may be implemented and used.

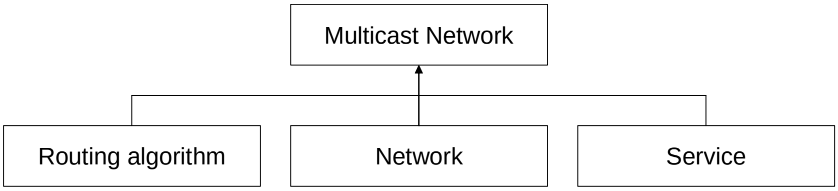


Figure 5.1 - Basic Framework of the Object Model

A third class of objects, the routing algorithms, allows this integration. The basic framework of the object model is shown in figure 5.1.

5.2.1 Modeling Multicast Networks

In order to reduce the complexity of networking protocols, layered models are used. The protocols are divided into single, preferably independent layers, which are responsible for different operations. The basic concept of the horizontal structure is that every higher layer qualitatively improves the services of the lower layers [88]. Single parts of the protocols may be easily changed. Only the interfaces towards to the higher and to the lower layer have to be implemented as specified.

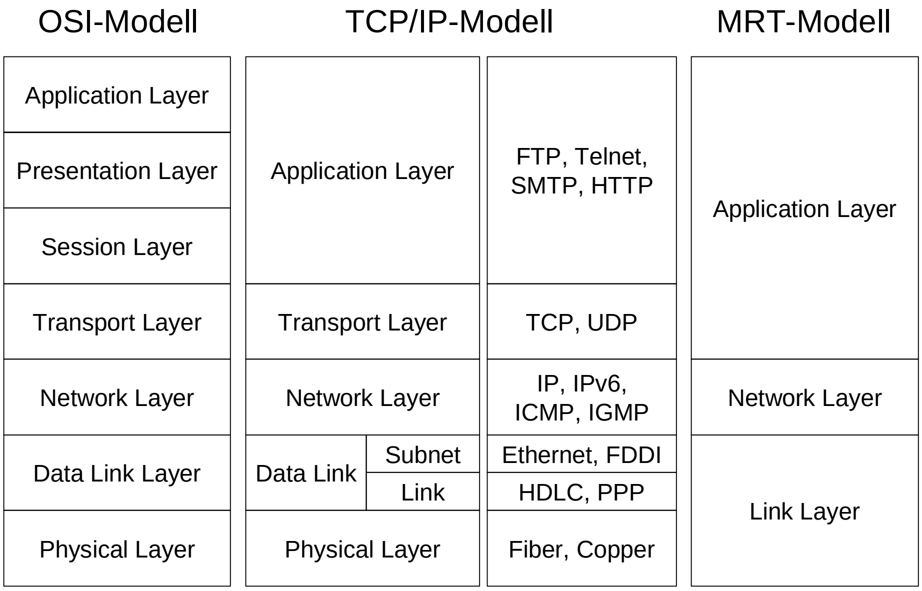


Figure 5.2 - OSI model, TCP/IP suite, and MRT mode

The most well-known example of a layered model is the OSI (open systems interconnect) model specified by the ISO (international standard organization). Despite the fact that the OSI protocols are no longer employed, they still serve as the primary example in lectures and text books for the description of the principles and advantages of layered networking protocols.

In the internet, the TCP/IP suite is used [193]. Like the ISO/OSI model, the TCP/IP suite is divided in single layers. A comparison of both models is shown in figure 5.2.

The lowest three layers of the OSI model and the TCP/IP suite are identical. The physical layer is responsible to access the physical medium and to use it for transmitting single bits.

The data link layer, which is subdivided in the TCP/IP model, provides the transmission of single data frames. Additionally, the access to the physical layer, an error detection and correction, a flow control mechanism and a synchronization of the individual blocks are all defined at this layer.

The network layer is responsible for transmitting message blocks, called packets, from one end system to another one. It also provides addressing schemes, routing algorithms and mechanisms to partition large packets into smaller frames.

The transport layer ensures an end-to-end connection. Transport layer protocols operating on the connection-oriented working principle, such as TCP, exist as well as others which provide a connection-less data transmission, such as UDP.

The TCP/IP suite integrates the session and presentation layer into the application layer. At this top layer, application specific services are defined. Also shown in figure 5.2 is the MRT model. The specification of multicast networks requires only some properties of the TCP/IP suite. Therefore, the MRT model foregoes separate definitions for the physical layer and the data link layer and combines both to a single link layer. An emerging question is whether the transport layer is really required for the MRT model. It has been shown that only very few parameters are required in order to calculate an optimum path through an IP multicast network. Therefore, the MRT model is organized into only three separate layers:

(1) link layer

(2) network layer

(3) application layer

UML (Uniform Modeling Language, [30]) has been used to specify the single objects. The advantage of this specification is the simplified implementation in a object oriented programming language such as JAVA.

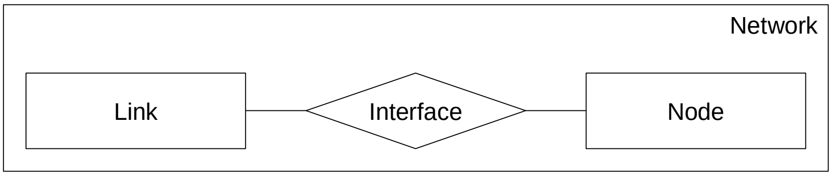


Figure 5.3 - Primary Objects to describe a Network

Figure 5.3 shows the primary objects describing a network: The network consists of components divided into three elementary classes. Nodes are composed of end systems and routers. The end systems run the applications which form the basis for the used multicast services. Routers connect the single networks and transport the multicast packets between all involved end systems. Typically, links are LANs (local area networks) such as ethernet or FDDI, which are used to transmit and exchange information. The nodes and the links are interconnected by interfaces.

Several layers of the MRT define different representations and properties of all the objects which are discussed in the following subsections.

5.2.1.1 Link Layer

The lowest layer of the MRT model includes information about:

- The physical medium, and
- the protocol used at the data link layer.

Typical media are copper wires, fiber optics, and wireless transmissions using radio signals. In order to calculate the quality of service, parameters such as the transmission delay of the employed medium have to be available.

Each of various protocols of the data link layer displays very different characteristics. For example, techniques such as ethernet, POS (packet over sonet) and ATM (asynchronous transfer mode) show a different behavior concerning the transmission quality or the natively support of multicast. ATM provides capabilities to set up connections with a guaranteed quality of service. Other protocols rely on the functionality of higher layers to support some higher degree of quality.

The capabilities to directly support multicast protocols differ as well. Typical IP over ATM protocols such as classical IP or LANE (LAN emulation) have either no knowledge about multicast or use their broadcast mechanism for the transmission of multicast traffic as well. Current ethernet switches implement a technique called IGMP snooping [42]. Special ASICs analyze each received multicast packet in order to use the acquired information to build more precise allocation tables, which associate ethernet multicast addresses and the appropriate ports. Without such a mechanism, each multicast packet has to be flooded through the whole local network in the same way as broadcast packets are treated.

Another capability of the link layer is the mapping of IP multicast addresses to MAC addresses and vice versa. For example, ethernet directly supports ethernet multicast addresses but only offers 23 bit to distinguish between them. IP multicast addresses include 28 significant bit. Therefore, $2^5 = 32$ IP addresses are mapped to a single ethernet address. Even if the probability of observing this effect is very low, because a total number 2^{28} different IP multicast addresses exist, bottlenecks may appear reducing the overall performance at the end systems which are participating at the multicast sessions.

The following subsections introduce the properties of the basic objects at the link layer of the MRT model.

5.2.1.1.1 Object: Link

Link type: In general, the LAN type and the used protocol are identical. In individual cases, the usage of the protocol may vary. This can have a high impact on the calculation of the quality of service. Therefore, the object link should include both the type of the link as well as the used protocol. Typical examples are a shared ethernet, a switched ethernet, FDDI, point-to-point connections, and ATM networks. The object structure is shown in figure 5.4.

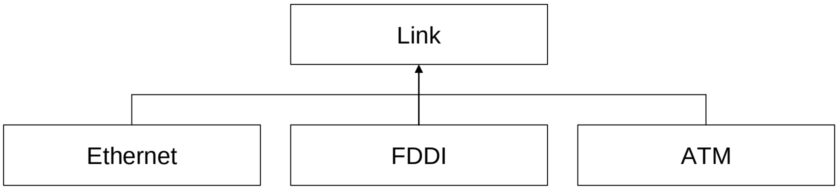


Figure 5.4 - Object Structure of a Link

Accepted interfaces: This parameter defines the interface types which can be connected to this link as well as the maximum number of allowed interfaces. For example, exactly two systems may be connected by a point-to-point link.

Medium type: Typical media are copper wires, fiber optics or radio waves. Based on the properties of these physical media, it is possible to calculate the latency depending on the velocity of the propagation, for example: light in a fiber optics environment. Additionally, default values for the bit error rate can be implemented based on the experience using a particular medium.

Medium length: Combined with the medium type, the length is required in order to compute the bit error ratio or the propagation delay.

Capacity: The type only specifies the protocol or the protocol option. Most links provide a wide range of interface capacities. For example, ethernet is defined for 10 Mbps, 100 Mbps and 1 Gbps. The deployment of the next generation (10 Gbps) is currently underway. Other protocols such as SDH and ATM provide connections with a capacity of 2 Mbps (E1) up to 10 Gbps (OC-192). The capacity of the link does not strongly affect the calculations. The more interesting parameter is the capacity of the interface which is described below.

Transmission delay: The transmission delay describes the time which is required to transport a complete frame from one system to another. Not considered are questions concerning the routing, which are handled in the network layer. Basically, the transmission delay is composed of the medium access time, the transmission time and the propagation delay [105]. Nevertheless, especially in present implementations, an additional delay due to queuing mechanisms and due to complex data link layer structures has to be added to the transmission delay.

MTU (maximum transmission unit): Most protocols do not define a generally accepted packet size. This is of critical importance for real-time transmissions because queue lengths and buffer sizes cannot be globally calculated and reserved in order to achieve a predictable behavior of the data transmission. It is possible to use the defined maximum size of a frame (MTU) for simulations and analytical analysis in order to calculate the worst case for the available quality of service.

Bit error ratio (BER): The bit error ratio is used to compute one of the most important quality of service parameters, the packet loss ratio. The bit error ratios are known for most media and vary between 10^{-12} and 10^{-4} for fiber optics and satellite links respectively. The packet loss ratio (PLR) can be calculated as follows: $PLR = 1 - (1 - BER)^N$. The packet loss ratio describes the probability that a packet of size N bit is correctly transmitted [105].

5.2.1.1.2 Object: Interface

MAC address: The layer-2 address of an interface. This parameter is optional, because the routing algorithm only uses the information at the network layer, i.e., the IP address.

Capacity: The capacity of an interface defines the available bandwidth towards to an end system. Because it is possible to define different capacities for the object link and the object interface, only the latter one is used for calculations because it describes the capabilities of the particular end system.

Queuing delay: Most systems, especially end systems, use the simple FIFO (First-In, First-out) queuing mechanism. Based on an average load of an interface, ρ, and an assumed exponential distribution of the arrival rate, λ, the average queue length, L, can be calculated using the M/M/1-FCFS model as follows (see also [108]): $L = \frac{\rho}{1-\rho}$. Using Little’s law, the average waiting time, W, is the result of $W = \frac{\rho}{\lambda(1-\rho)}$. The operating systems of typical internet routers allow the use of different queuing strategies. This is particularly important for the transmission of real-time traffic. Unfortunately, only a few carriers have implemented such mechanisms because of the high administrative complexity.

In contrast to the common assumption that network traffic is exponentially distributed, current research activities have proven that the actual network traffic shows a self-similar behavior, i.e., it is subject to high variations over longer times [206], [77], [78]. This is important for the calculations of the multicast routing tool which calculates the queuing delay using the average bandwidth and the utilization of the interface. Of course, this is important for off-line simulations. Using online measurements, it is always possible to use actual measurement results for the calculation.

MTU (maximum transmission unit): Every end system can freely choose the MTU size within the specifications of the protocol. Typically, the maximum defined value is used.

Packet loss ratio: This parameter shows the number of dropped packets at the node. Typical reasons are congested queues or overloaded processors. Routers also implement congestion control mechanisms such as RED (random early discard) which randomly drop packets to prevent a overload situation. The packet loss ratio is estimated by measurements. Additionally, the nodes can implement mechanisms in order to to query the packet loss ratio from measurement probes.

5.2.1.1.3 Object: Node

Node type: Basically, there are three types of systems which are considered in the model: hosts, routers, and bridges. Figure 5.5 provides an overview of these objects.

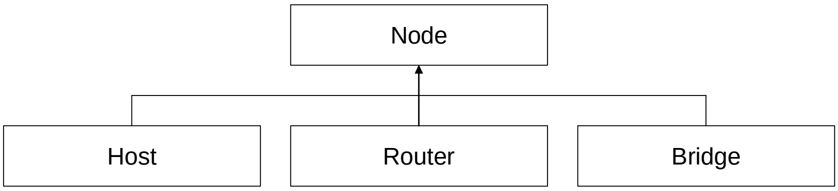


Figure 5.5 - Object Structure of a Node

Hosts are end systems which allow the acquisition and delivery of multicast packets. Routers interconnect different links. At the link layer, the MRT model does not distinguish between hosts and routers. A third type of node is the bridge. Bridges also connect different links but show a different behavior than routers. Routers operate at the network layer in order to forward packets through the network. Bridges work at the data link layer by translating the different link layer protocols. Bridges introduce an additional delay whilst transmitting frames. The frames have to be stored, the headers have to be changed and the new frames have to be transmitted to the destination network. Therefore, the delay is composed of the time it takes to store all bits in the memory, the processing time and the time it takes to transmit the frame to the destination network.

Accepted interfaces: Each system can hold a maximum number of interfaces which can be configured using the accepted interfaces parameter.

Load: The processing time for each packet strongly depends on the current load of a system. The greater the available processor capacity, the more effectively the queues can be processed.

5.2.1.2 Network Layer

The most important layer for the calculation of the optimum path through the network is the network layer. Optimum paths can be found using information about available routes and one or more metrics of these paths describing the quality of service of each one. The network layer of the MRT model is responsible for combining all required information for such calculations.

5.2.1.2.1 Object: Link

The link object accomplishes only few functions at the network layer. Therefore, no new parameters are introduced at this time.

Nevertheless, a new type of link has to be introduced: a network cloud. This type is especially important for the global IP multicast routing. A cloud covers unknown parts of a network. A typical example is a tunnel. A tunnel is a logical connection between two systems. The transmitted data are encapsulated into the tunnel protocol at one side and de-capsulated at the other side. The first global multicast network, the MBone, used tunnel structures to interconnect the single multicast enabled networks. A similar construct emerges today by using VPNs (virtual private networks). The reason for defining this type of link at the network layer is to make it independend of the physical network structure. Therefore, the underlying network structure can be separately modeled and used for calculations of the quality of service.

5.2.1.2.2 Object: Interface

IP address: The calculation of paths through the network depends on an addressing scheme. In an IP network, IP addresses are used for this task. Implementations are available which allow more than one IP address per interface. The MRT model considers such subinterfaces as single, separately modeled interfaces.

5.2.1.2.3 Object: Node

Node type: The distinction between hosts and routers is made at the network layer. The difference between the two objects is that routers are capable of forwarding packets between different links.

Quality of service (QoS): A number of approaches and implementations, described in section 3.6, try to enhance or even to guarantee the quality of service of particular transmissions. The differentiated services architecture uses the TOS byte in the IP header to prioritize single packet flows. Associated scheduling algorithms analyze the TOS field and apply an appropriate forwarding behavior. Other concepts are based on a resource reservation from end-to-end in order to guarantee some amount of quality. An example is the resource reservation protocol. The QoS parameter of the MRT model describes the reserved or the requested quality of service for a particular connection.

5.2.1.3 Application Layer

Most of the functionality of the application layer is specified in a separate class of objects, the services. In order to create a reference between the network objects and the used services, the membership of end systems in the various multicast groups is added to the application layer parameters.

Typically, different types of relationships between an end system and a multicast group are defined:

- end systems can send data to a multicast address,
- they can receive data from this group, and
- the nodes can both send and receive.

Applications working according to the one-to-many principle include exactly one sender and any number of receivers (possibly none). The routing decisions are taken based on standard algorithms because no superposition of the data transfers within the service can emerge. Many-to-many transmissions in multicast networks occure more frequently. For the same service, a number of senders and receivers exist. Each receiver gets data from all senders. Therefore, parts of the network, at least the connection of the receiver, are shared for all active transmissions of this service. This behavior has to be incorporated for the calculation of optimum paths through the network. The same applies for considering simultaneously running services, if they are spatiotemporally overlapping.

5.2.2 Modeling Multicast Services

Another class of objects to introduce are the multicast services. Basically, these objects are completely independent of the underlying networks. Typical multicast applications such as teleconferences consist of a collection of tools which harmonize altogether. Nevertheless, these tools work independently from each other and can be assembled by the user based on his requirements. For example, a video conference typically consists of a video and an audio channel. Often, other media are required in use together with the audio/video channels. Typical examples are a whiteboard allowing the participants of a conference to commonly draw figures, or a shared text editor for creating a session protocol.

5.2.2.1 Object: Service

Figure 5.6 shows the object structure of the MRT model for multicast services. A service contains a number of service parts. These parts consist of definitions of membership relations to single multicast groups. The link to the network model is provided by these multicast groups.

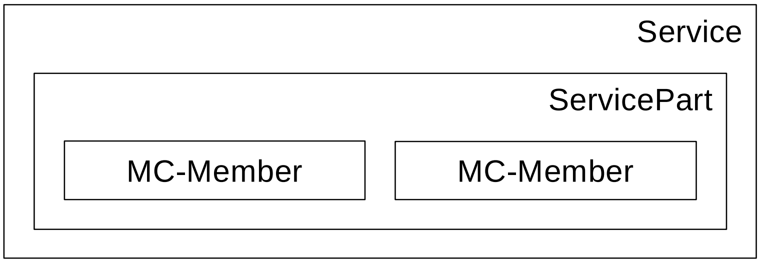


Figure 5.6 - Basic Objects of Multicast Services

5.2.2.2 Object: Service Part

The object service part is described in this section including all its properties which are required in the context of the calculation of optimum paths through a multicast network for a particular service.

<u>Application type:</u> The various applications strongly differ in their resource requirements, their quality requirements, and their behavior in the case of overload. Therefore, it is required to model these properties in order to calculate optimum paths through the network for a particular service.

Even the presentation of the application (audio, video, whiteboard) is insufficient. For example, a video transmission of a conference requires only about 64 kbps throughput but a maximum delay of 200 ms restricts the usage. Other high quality video broadcasts may require a guaranteed throughput of 10 Mbps but have no hard limits for the delay. Additionally, other QoS parameters of a service part have to be modeled, such as the maximum jitter or the maximum packet loss ratio.

Multicast address: Each application transmits the data to a single multicast group. The network is responsible for ensuring the proper delivery to all active receivers.

Port: The port has the same function as for unicast transmissions. The port number identifies the process on the receiving host which is waiting for the transmitted data.

TTL (time to live): The TTL defines the maximum lifetime of an IP packet. At each hop (router) through the network, the TTL is decremented by one. If the value of the TTL becomes zero, the packet is discarded. The reason for this behavior is to prevent endless circulating packets due to routing loops. In IP multicast networks, the TTL is also used to administratively restrict the propagation of multicast flows. For example, the border router of an enterprise should be configured to ensure that no multicast packets with a TTL < 32 leave the network of the organization.

5.2.2.3 Examples of Multicast Services

The following figures show two examples of the object structure. A video conference including three service parts (video, audio and whiteboard) is shown in figure 5.7.

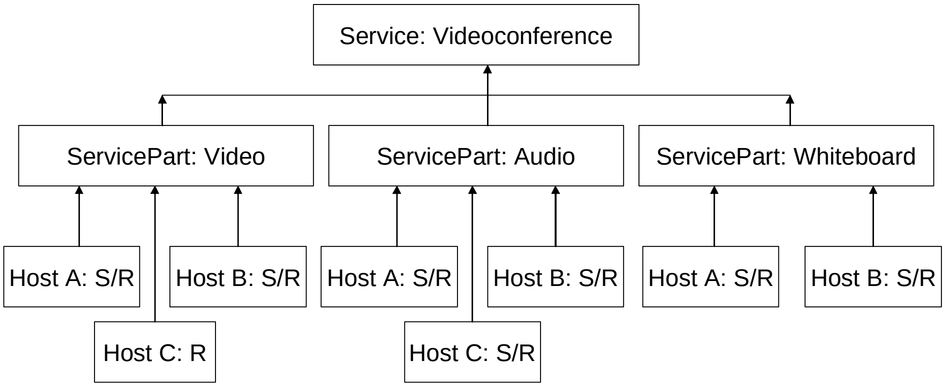


Figure 5.7 - Example of the Object Structure of a Video Conference

The letters S and R identify the sender and the receivers respectively. In a typical conference scenario all participating hosts send and receive data to and from the multicast network. In this example, Host A and B are participating at all service parts. Both hosts send and receive data on all available channels. Host C is only receiving the video and has no whiteboard application installed. Therefore, host C is participating bidirectionally at the audio service part only.

The typical situation of a one-to-many transmission is shown in figure 5.8. A single sender (host A) is transmitting data on a number of parallel channels (audio and video) towards several receiving hosts. Host B and host C, which are consuming the data are not sending information back to the channels.

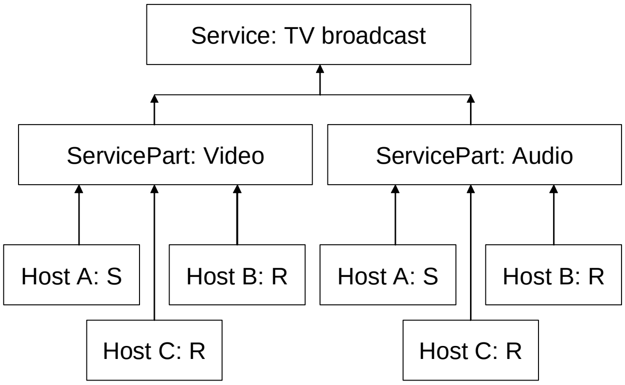


Figure 5.8 - Example of the Object Structure of a TV-Broadcast

Typically, at least status information about the reception quality are sent back to the source of the multimedia data. Therefore, the last example is an unusual situation. Nevertheless, the back channel commonly requires less or no guaranteed quality of service for its data transfer than the multimedia transmission.

5.2.3 Routing Algorithms

The primary goal of the development of the MRT model was to provide a framework, which allows one to gather data about a multicast network including the services which are using this network. This model is intended to make decisions based on the collected information, if a particular service can be used and what quality can be expected.

The final decision can only be made if the employed paths through the network are known exactly. Two concepts exist in order to achieve this knowledge:

The routing tables of all gateways have to first be brought together and then analyzed. This results in knowledge about the real behavior of the network.

Secondly, routing algorithms can use the modeled information in order to provide proposals about a optimal usage of the network. Additionally, simulations can use such algorithms to detect potential bottlenecks.

Unfortunately, the routing of IP multicast cannot be reduced to the problem of finding a best path through the network from a source host A to a destination host B. Even the solution to this problem is often not simple. New approaches and concepts, which are not subject of this work, have to be developed in order to find routing protocols and strategies for IP multicast networks [154], [155], [158] and the completeness, the computability and the quality of such new algorithms have to be verified. A complete taxonomy of multicast routing algorithms can be found in [199]. Additionally, an analytical comparison is provided in [175].

The following subsections introduce some well-known concepts as well as some work in progress. The Dijkstra algorithm [60] is explained, which has been chosen for the prototypical implementation of the MRT model.

5.2.3.1 Optimum Multicast Trees

The search for an optimum multicast tree is also known as the Steiner tree problem [207], [123]. The Steiner tree problem in graphs (SPG) can be described as follows: Let $G = (V, E)$ be an undirected graph with a finite set of vertices, *V*, an edge set, *E*, and a cost function assigning a positive, real cost to each edge in *E*. Given a subset $S \subseteq V$, a subgraph $G' = (V', E')$ of *G* such that *V'* contains all the vertices in *S*, *G'* is connected and the sum of the cost of all edges in *E'* is minimal [200], [201].

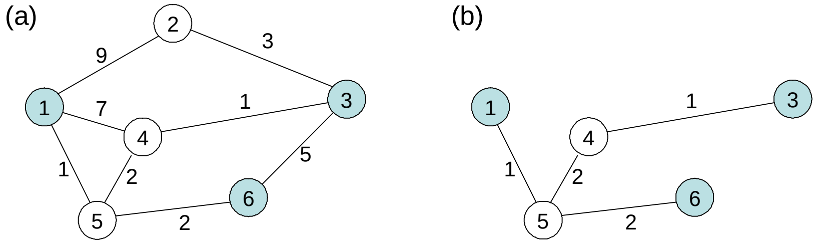


Figure 5.9 - Steiner Tree Problem in Graphs [201]

Figure 5.9 shows an example graph (a) and a Steiner tree solution (b). The selected, or special, vertices are color-labeled. The optimal solution has a cost of six. In this example, the solution includes some non special vertices. These vertices are referred to as Steiner vertices.

In the past, it has been shown that the Steiner tree problem in graphs is *NP*-complete, i.e., there is no known algorithm which can solve the problem in polynomial time [44], [86]. Therefore, heuristics and approximations are used which allow multicast routing decisions to be made in a short time period [169]. The most well-known class of algorithms which can find a suitable solution for the problem are the shortest path tree algorithms. One commonly used instance, which is also found in MRT, is the Dijkstra algorithm, which is discussed in the next subsection. Another class are the minimum spanning tree algorithms. An example is the Prim algorithm, which is described in detail in [45].

5.2.3.2 MRT: Dijkstra-Algorithm

The Dijkstra algorithm has been chosen for the first, prototypical implementation of the MRT. This algorithm allows one to solve the shortest path problem for a single source, i.e., a spanning tree is calculated for the transmission of data from a particular source to any destination in the network.

The algorithm uses a graph $G = (V, E)$ consisting of a number of vertices, *V*, and a number of directed and, by a single metric weighted edges, *E*. Let $w(u, v)$ be a cost function, whereas $(u, v) \in E$, assigning a metric to the vertex from *u* to *v* and $s \in V$ be a starting vertex. The algorithm processes a set *S* of vertices, for which the shortest path has already been calculated, i.e., for which the distance towards to the starting vertex *s* is known. The algorithm allows the inclusion of quality of service parameters of single edges (metrics of a connection). The complexity of $O(V^2)$ is maintainable. If a particular service includes more than one source, *n* passes of the algorithm with a complexity of $O(V^2)$ each are required for each of the *n* sourced.

The Dijkstra algorithm works as shown in figure 5.10.

```
DIJKSTRA ($G$, $w$, $s$)
INITIALIZE-SINGLE-SOURCE ($G$, $s$)
$S \leftarrow 0$
$Q \leftarrow V[G]$
while $Q \neq 0$
   do $u \leftarrow$ EXTRACT-MIN($Q$)
      $S \leftarrow S \cup \{u\}$
      for each vertex $v \in$ Adj[$u$]
         do Relax ($u$, $v$, $w$)
```

Figure 5.10 - Working Principle of the Dijkstra Algorithm [45]

A more comprehensible description of the algorithm is shown in figure 5.11. The spanning tree for a source node *s* has to be calculated. The example network consists of five nodes (edges) including the source node. Various paths (vertices) connect the nodes. All vertices are directed and labeled with a cost.

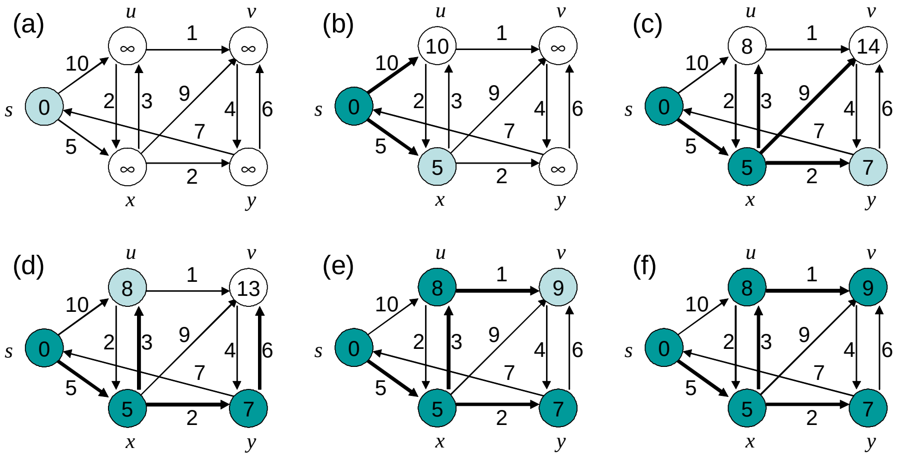


Figure 5.11 - Execution of the Dijkstra Algorithm [45]

(a) This is the situation at the beginning of the while-loop. The source edge is labeled with 0 and all others with ∝. The current node is marked with a pale color.

(b) In the first iteration, all neighbors of *s*, which are all vertices $v \in V$ for which there is an $e \in E$ with $e = s \rightarrow v$, are labeled with the cost of the edge from *s* to *v* plus the label of the current vertex. Additionally, the edges are marked bold from *s* to each neighbor *v*. Finally, a new current node is selected using the following criteria: the node has not yet been a current node, and the label of this node is minimal compared with the other candidates. Therefore *x* is selected as the next current node. The last current node is marked with a dark color in order to distinguish the nodes which have already been a current node.

(c) The same steps as in (b) are accomplished. The neighboring nodes of *x* are newly labeled if the calculated cost is smaller than the label. If a new label is applied to a vertex, the edge towards this vertex which is currently marked bold is unmarked and the new connection is marked. In the shown example, the edge from *s* to *u* is unmarked and the edge from *x* to *u* is marked. Finally, a new vertex, *y*, is chosen based on the minimum label.

(d), (e) The iterations of the algorithm are continued until no vertex is left which has not yet become a working node.

(f) This is the situation after the successful execution of the Dijkstra algorithm. All vertices are marked with a dark color and all are labeled with a cost value. Much more important is the marking of the edges. The resulting spanning tree correlates with the marked edges.

Unfortunately, the algorithm accepts only a single metric describing the quality of service. Typically, routing protocols using the Dijkstra algorithm such as OSPF (Open Shortest Path First, [149]) use only a single parameter such as the bandwidth or an administratively configured metric. A new concept is discussed in the next subsection.

5.2.3.3 Routing based on multiple QoS Parameters

In [117], an interesting approach is described which takes routing decisions based on a number of simultaneously operating metrics. Examples are the available bandwidth, the end-to-end delay and the packet loss ratio. The QoS metrics are classified into three categories additive, multiplicative, and concave [204]. Let $m(l)$ be defined as the performance metric for link *l*, for any path $P(u, v) = (u, i, j, \ldots, k, v)$ from node *u* to *v*.

- A metric is additive, if $m(u, v) = m(u, i) + m(i, j) + \ldots + m(k, v)$. For example, the end-to-end delay $d(u, v)$ is additive and is equal to the sum of the individual link metrics along the path.
- A metric is multiplicative, if $m(u, v) = m(u, i) \times m(i, j) \times \ldots \times m(k, v)$. For example, the probability, $1 - P(u, v)$, for a packet to reach node *v* from node *u* along *P(u,v)* is multiplicative and is equal to the product of the individual link metrics along the path.
- A metric is concave, if $m(u, v) = min[m(u, i), m(i, j), \ldots, m(k, v)]$. For example, the bandwidth $b(u, v)$, available along a path, is concave and is equal to the minimum bandwidth along the links on $P(u, v)$.

The idea from Kanbara et al. [117] is to execute a single run of the Dijkstra algorithm for each quality of service parameter which is important for the final path selection. For each run, the best *n* solutions are recorded ordered by the best metric. A final comparison based on the minimum requirements of the application tries to find an optimum path based on the required quality of service.

The complexity of this algorithm only increases linearly with the number of QoS parameters to $O(RV^2)$. In a multicast environment, the calculation has to be performed separately for each sender. The impact between the concurrent multicast data flows is not observed.

5.2.4 Object Hierarchy

The complete object hierarchy of the MRT model is shown in figure 5.12. The top-level object network includes the previously described classes: the multicast networks, the multicast services, and the routing algorithms, which are used to calculate an optimum path through the network based on the application requirements. The network consists of devices, which can be divided into nodes, links, and interfaces. Examples of nodes are routers and hosts. Examples of links are ethernet, FDDI, and ATM. The services consist of service parts. The latter belong to multicast groups in order to achieve a relationship between the services and the nodes.

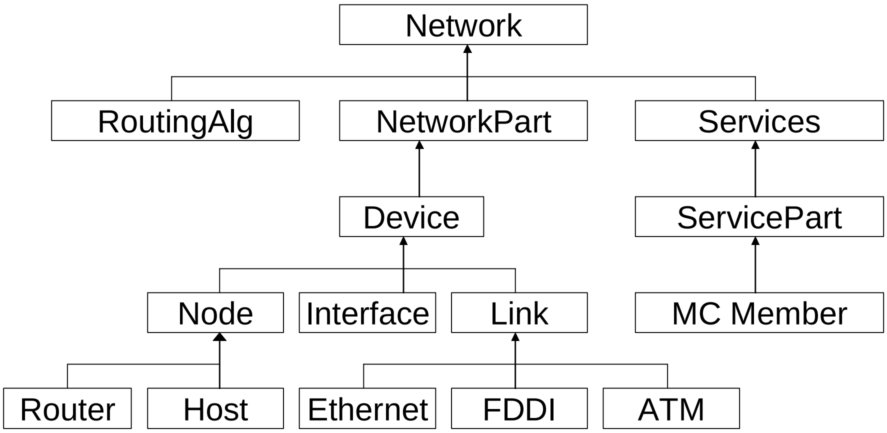


Figure 5.12 - Object Hierarchy of the MRT Model

The presented object structure is the basis for a prototypical implementation of the multicast routing tool, which is described in section 5.4.

5.3 Usage of the Model

Following the specification of the MRT model, typical application scenarios of the multicast routing tool are discussed in this section. All described scenarios can be covered by the existing implementation of the MRT. The following utilization methods are outlined:

- Analysis of a multicast network

  The analysis of a multicast network should draw conclusions about the structure of the network. Partitioned networks should be detected as well as the network parts, which are really required for a particular multicast service.

- Offline simulation

  In the preliminary stages of the development of new network structures or in order to install new applications with high resource and quality requirements, a simulation should provide initial information about the expected behavior of the network.

- Online measurement

  The MRT model does not define any measurement environment nor does it specify any test equipment to use. Nevertheless, a primary task of the model is to assist a network administrator with implementing the quality of service measurements by recommending a proper configuration for the measurements. This means that important parameters such as the positioning of the measurement probes within the network as well as an optimum measurement method for the particular services and the best operating times can be proposed by the MRT.

Additional applications are conceivable but these are typically based on the described principles. Employment of the multicast routing tool for the proposed approach to measure the quality of service in a multicast environment is obligatory (compare section 7 "Multicast Quality Monitor (MQM)").

5.3.1 Analysis of a Multicast Network

It is possible to represent multicast networks using the MRT. A view of the physical structure can be provided as well as a view of the logical characteristics. Additionally, the employed multicast services can be modeled. In summary, all this information allows the analysis of multicast networks by using available routing algorithms. Figure 5.13 shows a simple network consisting of a number of routers and four end systems (hosts). Without the knowledge about the required applications, a prediction of the functionality of the IP multicast services is not possible.

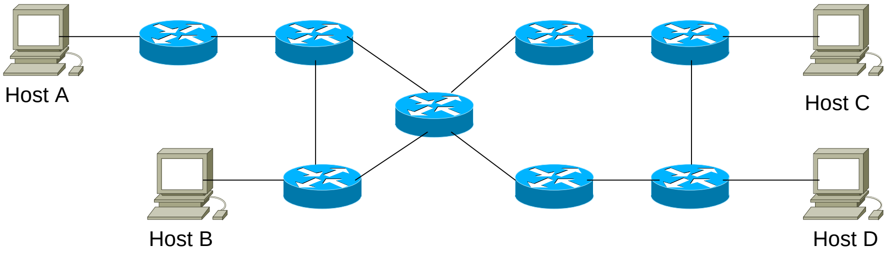


Figure 5.13 - Physical Structure of the Network

It certainly is possible to use the information about the structure of the IP network and to use directed queries to the concerned routers in order to calculate routes for unicast transmissions. A multicast network has a more complex structure. Additionally, there is a much higher number of involved parameters which have to be considered. Therefore, a multilevel approach is advisable. Transmissions in the multicast network typically work according to the many-to-many principle. Therefore, an analysis of multiple paths though the network is required. A single multicast tree exists for each sender towards to all receivers.

A historical view of the most common failures in the multicast routing in the campus network of the University of Erlangen-Nuremberg and those in the Bavarian university network shows that it is always necessary to analyze the employed services. The primary concept is to build a directed graph. The vertices of this graph are represented by the end systems of the service. The edges define the communications relationship between the nodes. All the edges can be weighted in order to define a required end-to-end quality of service.

Figure 5.14 shows an example of such a graph. A multicast service between three end systems is shown. Host A transmits data using IP multicast towards to two receivers within the network, host B and host C.

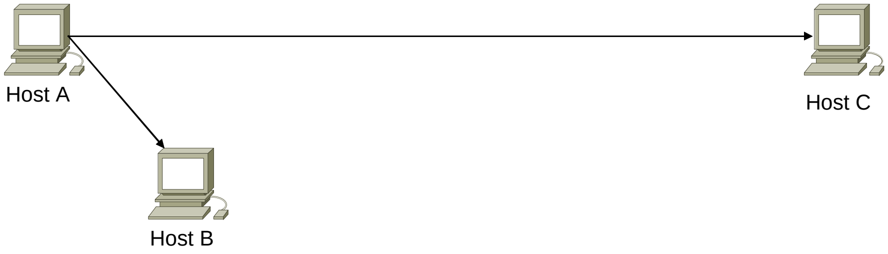


Figure 5.14 - Logical Data Flow of a Multicast Service

Until now, the model includes information about the physical structure of the network including all devices such as end systems and routers as well as the interfaces which connect the nodes and links. Additionally, information about the multicast service, which has to be considered, is included in the model. The inclusion of different service parts has been avoided in this example without the restriction of generality. The affiliation of end systems to a multicast service including the characterization of a node to be a sender or a receiver is shown.

In order to acquire more precise information from the network showing the relevance of the examined multicast service, the graph showing the communications relationship has to be overlaid with information about the physical IP network. The attached routing algorithm is used for this task.

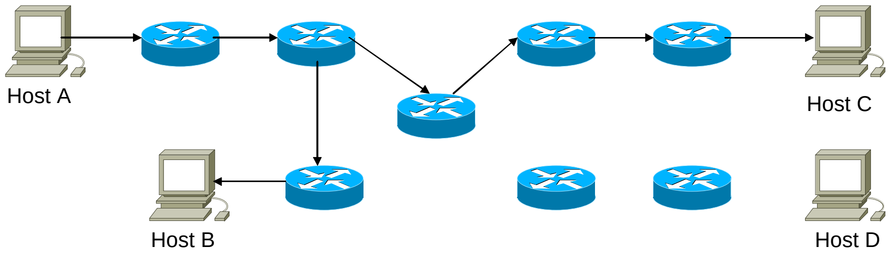


Figure 5.15 - Real Data Flow in the IP Network

The resulting graph is shown in figure 5.15. The connections of the physical network are reduced to those, which are really required for the multicast transmission of the selected service. Additionally, the basic reachability between all participating end systems can be verified without considering the required quality of service.

The multi-layered and service oriented model to describe multicast networks and the employed applications and services allows the network operator to easily acquire information about the functionality of the network. Additionally, it provides the capability to operate on very large multicast networks including a number of parallel working services. The quality of service for a particular application can also be predicted.

More meaningful results are possible if offline simulations are used or if measurement tools are implemented which gather more information out of the network.

5.3.2 Offline Simulation

One step further is possible if the model is used for offline simulations of multicast networks and the behavior of multicast services. The basis for all operations is the detailed description of the physical network and the employed applications within the MRT model.

The static model of the network is insufficient for the simulation. The behavior of the network and, therefore, the maximum available quality of service for the transmissions of the multicast services strongly depends on the utilization of the network, on the implemented mechanisms to increase or to guarantee any kind of quality, and the behavior of the single systems (implemented queuing mechanism and the like).

The information in the model has to be enhanced by simulated measurement data. There are a number of tools available for simulations of unicast networks. One example is the network simulator 2 (ns-2, [75]). Using this or similar tools, it is possible to estimate the average utilization of single connections in the network. These parameters fulfill the first condition for simulating multicast networks.

5.3.3 Online Measurement

The basic principle of the MRT model is the abstraction of the underlying network layers. Using the model, it should be possible to accomplish a meaningful analysis of the gathered measurement information. Additionally, optimum places to install measurement probes and suitable measurement methods may be determined. The chosen methods allow a simple test of the functionality as well as the measurement of the current quality of a network connection.

The routing algorithm of the MRT uses the modeled physical network structure in addition to the information about the multicast services in order to calculate an optimum path through the network based on the applications' requirements. This knowledge about the currently employed paths in the physical network allows the identification of ideal places for the measurement equipment.

Figure 5.16 shows the example network, which has been presented in section 5.3.1. Also shown are some distributed measurement probes. This is only one possible measurement setup.

Typically, these measurement instances are placed along the used paths. Unfortunately, a compromise often has to be made for the following reason: Optimally, measurement probes are deployed along the used paths. Additionally, the tools should be installed close to the end systems which are participating in the multicast services. Finally, in case of a network problem, which typically results in a re-routing, i.e., the employment of completely new paths, the measurement probes should be distributed all over the network. The responsibility for and the accessibility of the different routers in the network have to be considered as well.

A compromise between all these conditions cannot be achieved in every case.

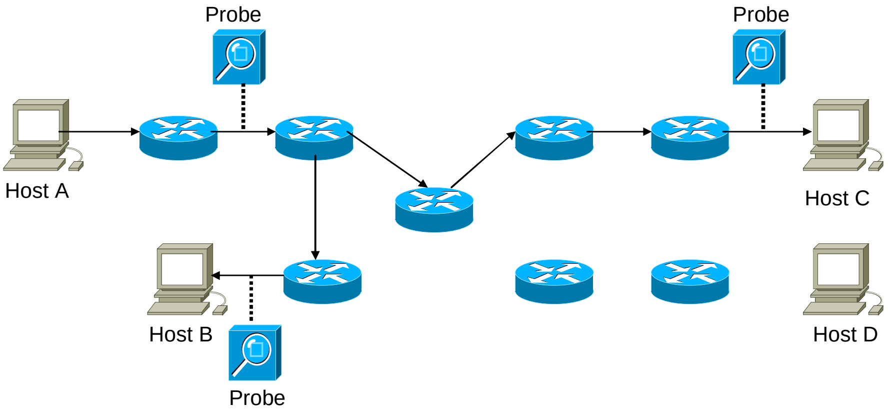


Figure 5.16 - Distribution of the Measurement Probes

By integrating dynamically gathered information the graph-oriented, static model of multicast networks can be enhanced to a operational model. The model is characterized by a graph, which describes the network structure, a number of states, including the current state of the multicast network and the services, and a number of state transitions [119].

Each state is described by a certain amount of information such as the functionality of the reachability:

- The functionality of networks (UP / DOWN) specifies the reachability of network partitions. In general, the functionality of sub networks can be identified by querying the responsible network devices such as routers.
- The functionality of a network interface (UP / DOWN) provides information about the state of end systems or single interfaces of routers. If an interface is functioning, data can be received and transmitted. As an example, the detection of this value can be achieved by using the unicast based reachability measurement tool ping, which uses ICMP in order to query a destination.

Even more complex state can be integrated into the model using a suitable technological description. Two examples are given in order to explain the principle:

- Unicast routing tables show the path to a destination IP address, if one exists, including the outgoing interface.
- The state of network connections, i.e., the operational bandwidth, the delay, and the jitter, includes information about the currently available quality of service of this connection.

The MRT model helps to control the queries of real systems in order to prevent the retrieval of unnecessary information. For example, if the employed multicast services do not have any quality of service requirements, it is not necessary to implement measurements of the quality of service parameters. The quality of TV broadcasts does not depend on the one-way delay, but the video transmission is sensitive to the jitter. Therefore, no measurement of the one-way delay, which is expensive due to the synchronization requirements, is necessary. The jitter is much easier to estimate.

By gathering only selected values, the number of queries, which might dramatically affect the network, can be reduced. Additionally, the measurement data can be more efficiently computed. Necessary transmissions of measurement information between the probes and the central computer, which is analyzing the data, can also be reduced.

If partitions of the network exist for which the internal physical network structure is not known, it is difficult to predict the behavior of the complete network just by "looking" at it.

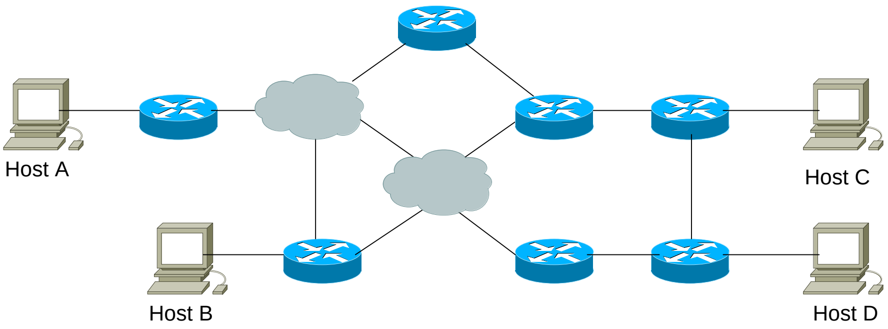


Figure 5.17 - Network Structure containing unknown Parts (Clouds)

The calculation of optimum paths through the network regarding a required quality of service of a particular service is no longer possible based on the analytical information about the physical structure of the network, or based on simulations of the theoretical utilization of the single network connections and the node. An example of such a network structure is shown in figure 5.17.

Information about the optimum path between host A and C can be only obtained by using suitable measurements. An example of a useful measurement setup has been already shown in figure 5.16. Ideally, additional measurement probes are implemented near the unknown parts of the network. This allows the collection of data defining the behavior of the network within the cloud as well as the prediction of the global network behavior. This is a common approach due to the insufficient access to the components in external networks. A potential solution is the incorporation of parts of the measurement equipment into the standard operating system of the gateways in the network. Nevertheless, the new features have to be enabled by the network administrators.

5.4 Implementation in JAVA

A first prototypical implementation of the MRT has been carried out at the University of Erlangen-Nuremberg. In the context of a master thesis, a student helped to develop a tool, named after the model: MRT - multicast routing tool.

The main focus of the implementation was to create an object oriented description of the multicast services and the underlying IP multicast infrastructure. Based on statically modeled properties as well as dynamically incorporated measurement results, an attached routing algorithm can be used to calculate optimal paths through the network for a particular multicast service.

The specification of the MRT model was achieved using UML. This language provides an easy adaptation in an object oriented programming language. JAVA was chosen for the implementation. This decision was based on the object oriented structure of JAVA and because this programming language allows a rapid prototyping of the model including a graphical front end. This graphical representation is required in order to simplify the modeling of the IP multicast infrastructure and the multicast services. Additionally, it is used to show the calculated best paths and potential problems in a comprehensible way.

The layers, which are discussed in section 5.2.1, have been realized. Specifically, the link layer, the network layer, and the application layer have been implemented. During this task, various similarities between objects at the different layers have been discovered. Admittedly, significant distinctions have been shown between the representations of the objects.

Each implemented layer defines a instance of each object. The properties of the single objects and those of the layers are parameters of the instances. Examples are the maximal available bandwidth and the utilization of a router.

Another class of objects defines the routing algorithms. The implemented alternative is the Dijkstra algorithm. It is used to calculate an optimum path from a sender to a receiver. In order to start the Dijkstra algorithm, a graph is first created, which is represented by a number of vertices and weighted edges. Using this graph, a spanning tree can be calculated. As not all hosts are active receivers of the multicast service, unrequired branches are pruned in a final step.

The single objects of the MRT are encapsulated using data abstraction methodologies [88]. Furthermore, they are improved with more unambiguous specifications. Starting with a global project, a network consisting of network parts, services, and routing algorithms are each specified and implemented. The complete realized object structure is shown in figure 5.18. The configuration of the multicast networks and of the multicast services has been already described in section 5.2.

The object routing algorithm includes sub objects simplifying the calculations by graph oriented algorithms. A new object, graph, has been created consisting of vertices and edges. Both, the vertices and the edges, can be weighted in order to give various metrics, which are used to calculate an optimum path. An adjacency matrix describes the neighboring relationships. This matrix is used to prune the unnecessary branches out of the calculated spanning tree in order to retain only the required vertices and edges in the tree.

New objects such as new link types can easily be included into the system and are immediately available to the user.

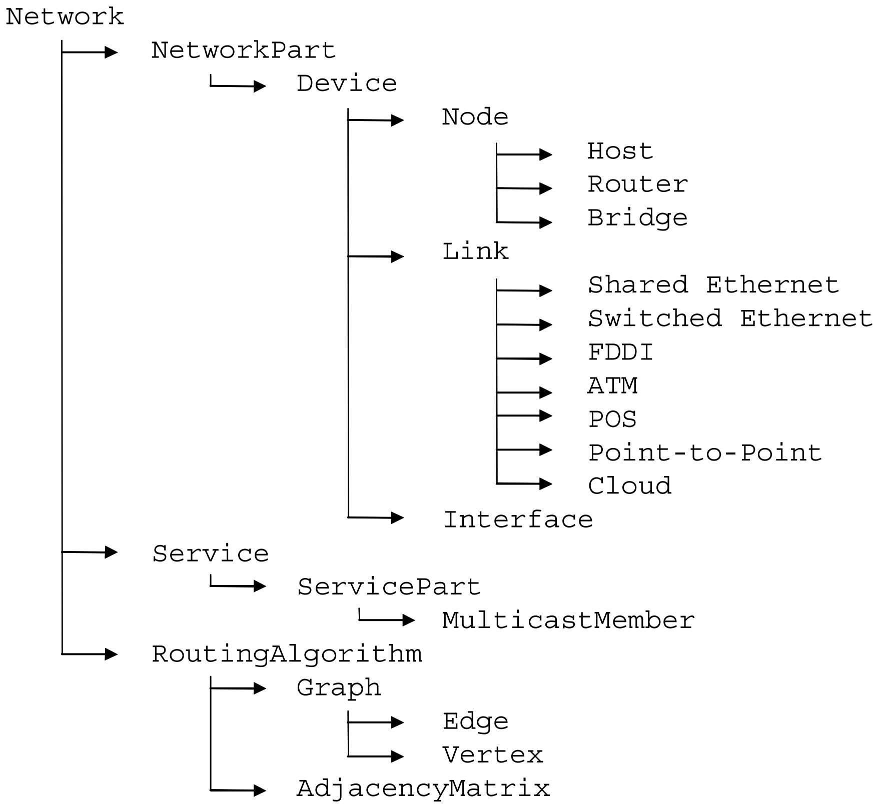


Figure 5.18 - Implemented Object Structure

Figure 5.19 shows a screenshot of the typical appearance of the MRT. A more detailed description of the implementation of the MRT is provided in appendix B. The main window always shows the current selected network. Single parts of the network can be hidden in order to provide an overview of the whole network.

Several menus allow the creation of new objects, which can be freely placed by the user on the desktop. The connections between the objects as well as the properties of each object can be configured using context menus. The example shows objects of the following classes: nodes (routers and end systems), links (ethernet switches and point-to-point links), and interfaces (the connections between nodes and links). All the objects are represented by intuitively understandable icons.

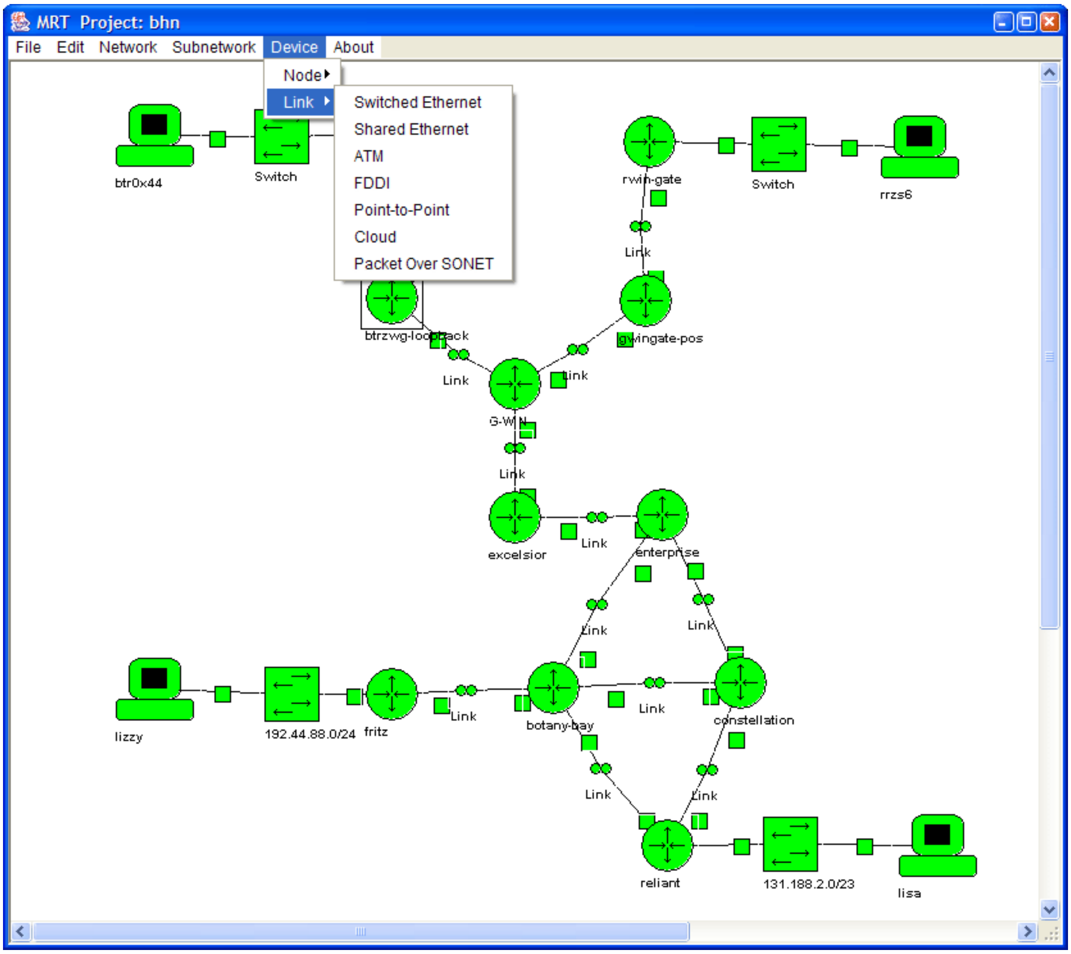


Figure 5.19 - Screenshot of the Multicast Routing Tools

As previously described, a first routing algorithm has been implemented, the Dijkstra algorithm. Following the definition of the multicast services including the service parts, the multicast group addresses and port numbers, and the participants, the distance calculation can be started. A single computation is started for each sender in of the service. The result is a number of optimum paths from each sender to all the receivers.

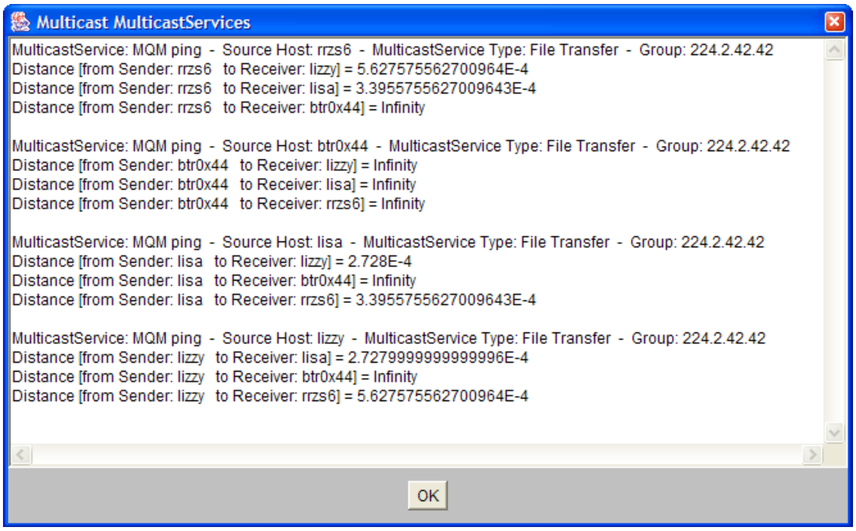
The example shown in figure 5.19 consists of four end systems displayed at the edges of the network. In particular, the network represents parts of the networks of the University of Erlangen-Nuremberg, the University of Bayreuth and the University of Regensburg, which are all interconnected by the German research network (G-WiN), represented by the single router in the middle of the network.

All four hosts are participating at a multicast session. For a more comprehensible illustration of the mechanisms of the MRT, we consider a video conference. Therefore, all hosts are sending and receiving data between each other.

An exemple calculation is shown in figure 5.20. The computation of the distance to each receiver is separately executed for all sending hosts. The distance numerically describes the quality of service of the end-to-end connection from the sender to the receiver. While no QoS measurement results are available, all the calculations are performed analytically based on the modeled properties of the networks, such as the maximum throughput of an interface.

In this example it is assumed that the border router of the University of Bayreuth has failed, or, at least, it is incapable of forwarding multicast traffic. The calculated distance from all senders to the host in Bayreuth is infinity. The same applies to the connections from Bayreuth to the other receivers.

As an example, the connections of the host rrzs6 in Regensburg are described. The distance to Bayreuth is infinity, i.e., there is no path between both hosts. The distance to the host lisa in Erlangen is much smaller (3.396 x 10^{-4}) than the distance to the host lizzy (5.628 x 10^{-4}), which is also located in Erlangen). The reason for this difference is the theoretically available bandwidth. The host lisa is connected by a gigabit ethernet interface to the campus network, whereas the host lizzy is only connected by a 100 Mbps interface.



```
MulticastService: MQM ping  -  Source Host: rrzs6  -  MulticastService Type: File Transfer  -  Group: 224.2.42.42
Distance [from Sender: rrzs6   to Receiver: lizzy] = 5.627575562700964E-4
Distance [from Sender: rrzs6   to Receiver: lisa] = 3.3955755627009643E-4
Distance [from Sender: rrzs6   to Receiver: btr0x44] = Infinity

MulticastService: MQM ping  -  Source Host: btr0x44  -  MulticastService Type: File Transfer  -  Group: 224.2.42.42
Distance [from Sender: btr0x44   to Receiver: lizzy] = Infinity
Distance [from Sender: btr0x44   to Receiver: lisa] = Infinity
Distance [from Sender: btr0x44   to Receiver: rrzs6] = Infinity

MulticastService: MQM ping  -  Source Host: lisa  -  MulticastService Type: File Transfer  -  Group: 224.2.42.42
Distance [from Sender: lisa   to Receiver: lizzy] = 2.728E-4
Distance [from Sender: lisa   to Receiver: btr0x44] = Infinity
Distance [from Sender: lisa   to Receiver: rrzs6] = 3.3955755627009643E-4

MulticastService: MQM ping  -  Source Host: lizzy  -  MulticastService Type: File Transfer  -  Group: 224.2.42.42
Distance [from Sender: lizzy   to Receiver: lisa] = 2.7279999999999996E-4
Distance [from Sender: lizzy   to Receiver: btr0x44] = Infinity
Distance [from Sender: lizzy   to Receiver: rrzs6] = 5.627575562700964E-4
```

Figure 5.20 - Calculated Distances from each Sender towards to all Receivers

The numerical calculation shows that a path between the several participants of the multicast conference exists or not. Additionally, the theoretically available quality of service is provided by a weighted distance. Admittedly, it is not shown which path is actually used for the transmission. Also not visible is the information concerning the location of network failures.

In a second step of the analysis, the multicast routing tool displays the calculated paths using different colors. The previous example is continued and the results are shown in figure 5.21. The paths, which are used for the conference, are marked red.

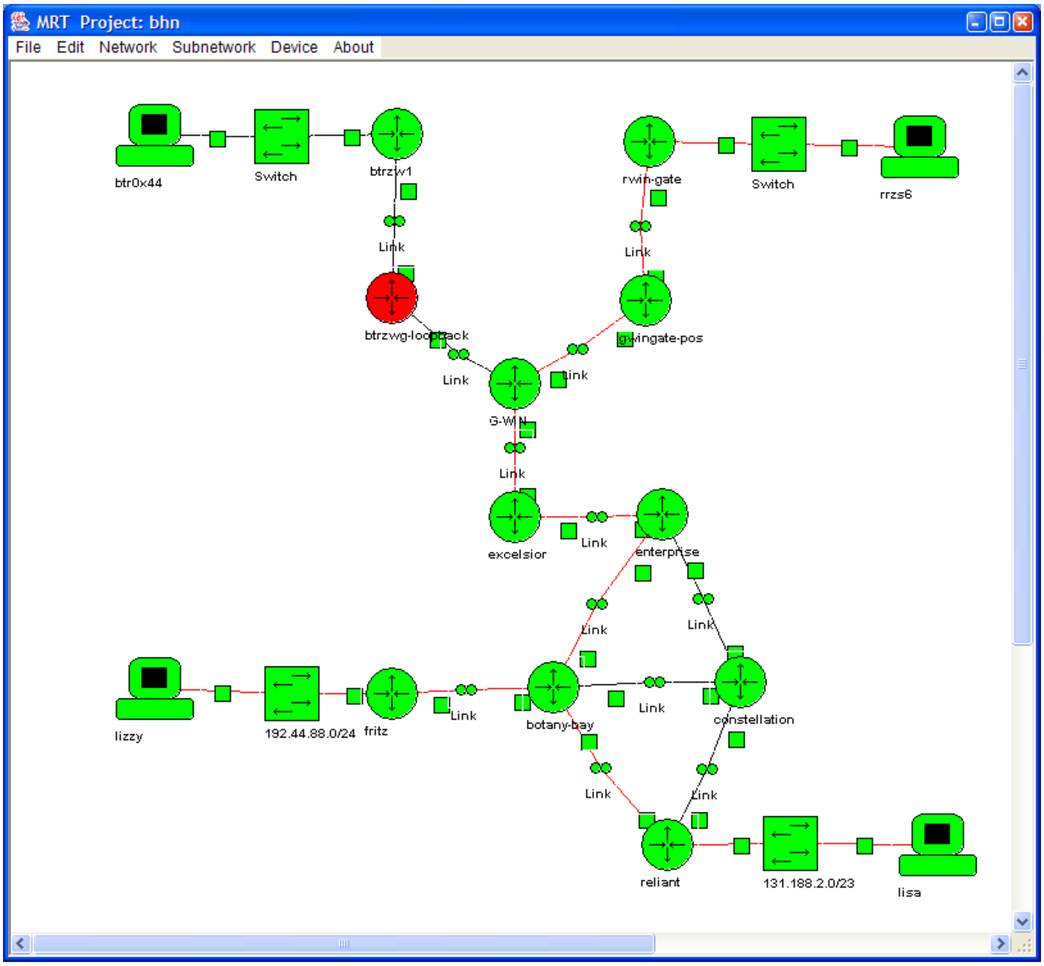


Figure 5.21 - Graphical Representation of the Reachability and the used Paths

Additionally, the MRT marks the locations of expected network failures. In figure 5.21, the border gateway of the University of Bayreuth is displayed in red. In this example, the failure has been configured manually. Nevertheless, the network administrator receives information about occurring failures or missing resources. He can then progress using suitable methodologies and tools in order to examine the failure more closely.

An interface to import dynamic information has been implemented as well. This interface is intended to incorporate measured data from quality of service measurement tools. The routing algorithm uses this data, if available, to calculate routes based on the current state of the network. This import function uses a readable text format.

During the implementation, attention has been paid to define the format in an expandable way. New types of measurement results can be added. Additionally, the function to import the dynamic data can be changed, for example from the reading of text files to a socket interface allowing the measurement probes to directly deliver the measured data to the multicast routing tool.

5.5 Summary

In the previous subsections a definition of the multicast routing tool (MRT) has been provided. The representation of the complex structures of a multicast network including the used multicast services is based on an object oriented approach. Using the MRT it is possible to realize the following goals:

- The integration of quality of service parameters allows one to model static properties such as the maximum throughput of an interface as well as dynamic states. The latter can be either simulated or forecasted and gathered from measurement probes. The measurement results may provide an overview of the functionality of a multicast network and also serve to to provide information about the current available quality of service of a network connection.
- The MRT model affords the calculation of optimum paths through the network based on the known available quality of service within the multicast network and the corresponding quality requirements of the employed multicast services. Using the implemented routing algorithm, it is not possible to determine the best path. Nevertheless, a best path permitting an optimal functionality of the application can be found.
- The graphical representation of the multicast networks provides an impression of the logical and the physical structure of these networks to the end user and to the network administrator respectively. Due to the integration of the visual representation of potential failures, the fault analysis and diagnostics is simplified. The precondition for maximized accuracy and efficiency is the well-organized deployment of measurement equipment and monitoring probes as well as the in-time integration of the measurement results into the MRT model.
- Additionally, the model provides support for implementing measurement probes within the network. Based on the requirements and the characteristics of the most important multicast services, the measurement equipment can be deployed and calibrated in order to achieve optimum results.
- Besides the optimum deployment, a primary task of the model is to help optimize the measurement methods. The MRT allows the installation of measurements based on the requirements of the current running services and on the expected multicast services respectively. Current active applications as well as forthcoming services can be located. An analysis of these services, for example, allows the direct employment of the active applications to measure the quality of service of the multicast network (if there are measurement tools available, which allow such a procedure). The quality of forthcoming multicast sessions can be predicted by defining measurement scenarios based on the requirements of these services and initiating the tests accordingly.

It is necessary to implement a monitoring environment for IP multicast networks in order to achieve the global goal to predict the expected quality of service for a particular application. Also required is a detailed analysis of these networks. Therefore, the next steps are required: First, a metric has to be developed, which can be applied to the quality of service within the network and to the quality requirements of a particular service. This metric is to be used in order to obtain a numerical description of the availability of the necessary resources. Secondly, appropriate measurement methods and tools have to be developed. These methods should obtain the test results using mechanisms which minimize the impact on the network and, therefore, also minimize the influence on other meaningful applications. It is important to be able to adjust the measurement methodology using the proposed model to obtain the minimum required information.

6 Definition of a Metric for Multicast Services

The basic reason for specifying metrics is to evaluate costs. Typically, service level agreements (SLAs) are used between the service providers and their customers. The mechanism of SLA is shortly explained in this section. The primary purpose of this section is to give an overview to multicast and quality of service metrics and to provide some basic calculation methods to quantify the quality of an end-to-end connection. This methodology is expanded to also cover the principles of multicast services. Additionally, the efficiency of multicast is evaluated. A few examples are provided explaining the significance of the discussed metric.

6.1 Parameters of the Metric

The first step towards a framework for defining internet performance metrics has been provided by Paxson [159], [160]. He distinguishes between analytically and empirically specified metrics. Another approach by Awerbuch et al. [19] describes a cost-sensitive analysis of communication protocols.

In the following, a number of basic parameters, i.e., single metrics are discussed.

6.1.1 Classic Parameters

First, a number of classic parameters are introduced. These values have been used in the past to describe the "quality" of networks. Typically, a network was called a high quality network, if the availability of this network was very high. A best path was chosen based on the maximum number of hops. The change of this attitude towards a quality of service enabled network is described by metric parameters shown in the next subsection.

- Bandwidth of a network link

  This parameter is defined by the maximum capacity of a network link to carry data, measured in bits per second, where data does not include those bits needed solely for link-layer headers. For links with variable sized transmission units, this metric is ill-defined unless a transmission size is also specified [159].

- Maximum throughput of a network path

  This value is specified by the maximum throughput on a given path in the internet between two hosts. Typically, it is defined by bottleneck bandwidth of the slowest path.

- Available bandwidth of a network path

  This parameter indicates the amount of bandwidth which can be utilized at a particular moment. The available bandwidth is a dynamic parameter which has to be measured for application in a metric.

- Hop count of a route

  The hop count is defined by the number of routers a packet visits on a particular path between two hosts.

- Reachability of a particular host

  The reachability of a particular host is provided at a given time if packets can be successfully transmitted from a given source to the particular host.

- Reliability of a network path

  The reliability is a measure for the distribution of the reachability over a period of time. Thus, the reliability of a network connection specifies the percentage successful reachability over the time. The measurement period must be included into the definition of this metric.

6.1.2 QoS related Parameters

Today, additional parameters to describe the quality of a network connection are required. Initiated by the implementation of multimedia services, the quality of service of a network connection became an important resource. The following metric parameters allow the weighting of a network path in terms of QoS.

- Delay along a network path

  This value defines the time a packet needs for traveling from end to end along a given network path.

- Maximum delay along a network path

  This value specifies the maximum measured delay on a particular path through the network. It can be used for the configuration of applications as the worse case value.

- Jitter along a network path

  The jitter is defined as the variance of the delay. This metric defines the jitter of a specific network path.

- Maximum jitter along a network path

  This value specifies the maximum measured jitter on a particular path through the network. The buffer size at the receiver of a data stream depends on this measure.

- Packet loss ratio along a network path

  The packet loss ratio at a given time is defined as the probability of a packet not to arrive successfully at its destination along a given network path.

6.1.3 Miscellaneous Parameters

Other metrics which cannot be classified to one of the already shown groups are discussed in this subsection. Typically, these parameters are not easy to "measure" or to "calculate". In general, empirical studies are used to gain the values of these parameters. Nevertheless, these metrics have to be included to the acquired overall quality which is visible to the end user.

- Mean NOC turn-around time

  This value defines the time period which elapses between the submission of a trouble ticket to the NOC (network operation center) and the final solution of the problem.

6.2 Service Level Agreements

Following the definitions of single metrics, patterns of utilization are discussed. Typically, so called service level agreements are used to specify a minimum service quality between ISPs and their customers and between different ISPs respectively.

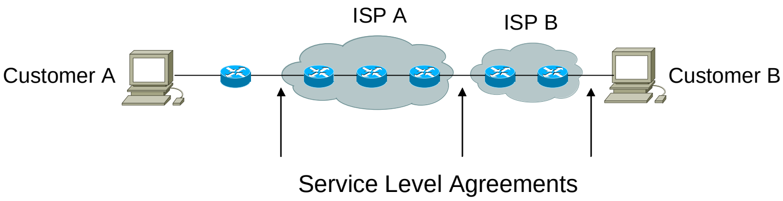


Figure 6.1 - Service Level Agreements on different Network Connections

The typical scenario is shown in figure 6.1. SLAs are set up at the borders of different networks. Single metrics, or even a number of them, can be used to specify a minimum assured service quality. Today, most SLAs are limited to the bandwidth of the link between the contracting parties and the reliability of common parts of the networks.

The specification of service level agreements always relies on monetary savings. The ISPs try to minimize the level of guarantee and the customers try to minimize their expenses.

Currently, no ISP is able to guarantee a minimum end-to-end quality. The internet is not able to carry high priority traffic. Admittedly, first providers started to employ QoS mechanisms in their networks which allow them to offer new service levels. Typically, such (expensive) services are called "premium" or "assured".

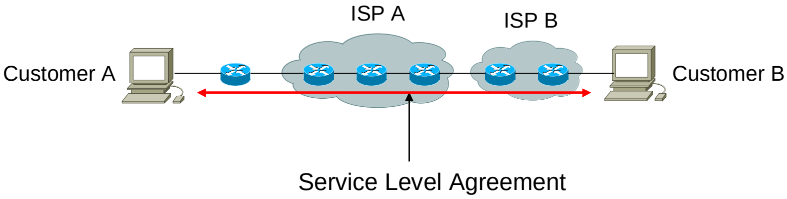


Figure 6.2 - Service Level Agreement on an End-to-End Basis

In general, there is still a discussion on the optimum definition of SLAs. In practice, a SLA is employed to define the behavior of a particular network connection. This type of SLA is much easier to specify and to control. In most research areas, the definition of the behavior of a network connection is demanded on an end-to-end basis [1], [160] as shown in figure 6.2. Therefore, a new type of SLAs is required describing the assured end-to-end service quality.

6.3 Calculation Methodology for Multicast Services

The main focus here is on specifying a calculation methodology providing a numerical quantification of the available quality of service for a particular multicast service. Three steps are required to calculate the desired result:

- weighting the metric values
- specification of a calculation method for the end-to-end quality
- definition of a methodology to apply the results to multicast services

6.3.1 Weighting of single Metrics

Calculation methods for quality of service specific metrics are discussed by Paxson [159] and the members of the IPPM WG of the IETF. Weighting methods are necessary to evaluate single measured metrics.

In general, the measured value *v* of a metric has to be first scaled using some scaling factor *k* so that $v' = k \times v$ is in the interval $0 \le v' \le 1$. The scaling factor *k* depends on the type of the metric as well as on the requirement of a particular service.

Furthermore, given the demands of a particular service which may define a maximum for the metric v_{max} and a favored value v_{fav}, the available quality *m(v)* for a particular service and a single metric *v'* can be defined as:

$$m(v) = \begin{cases} v' & (v < v_{\mathrm{fav}}) \\ s \times v' & (v_{\mathrm{fav}} \le v \le v_{\mathrm{max}}) \\ 0 & \text{else} \end{cases} \qquad \text{Equation 6.1}$$

Thus, the quality of the connection is defined as zero if no reasonable transmission is possible, and as *v'* if the demanded quality is available. If some metric value between the favored and the maximum value is acquired, the quality *m* is specified as a downscaled version of *v'*. The suggested value of the scaling factor *s* which is in the interval $0 \le s \le 1$ is 0.5.

The definition of the calculation method strongly depends on the metric type. For example, the calculation of the connectivity *m'* can be achieved using the formula:

$$m' = \begin{cases} 1 & \text{connectivity is provided} \\ 0 & \text{else} \end{cases} \qquad \text{Equation 6.2}$$

Nevertheless, this is only a simplification of the general form which has previously shown in equation 6.1.

6.3.2 Calculation of the End-to-End Quality

Typically, the end-to-end quality of a connection which is used by a particular service depends on more than a single metric. Thus, the quality of a particular connection *c* can be written as a vector of single weighted metrics m_i:

$$c = \begin{bmatrix} m_0 \\ m_1 \\ \dots \\ m_n \end{bmatrix} \qquad \text{Equation 6.3}$$

A numerical representation of the overall connection quality *c'* is the product of the values of *c*:

$$c' = m_0 \times m_1 \times \dots \times m_n \qquad \text{Equation 6.4}$$

The higher the value of *c'* which is in the interval $0 \le c' \le 1$, the higher the quality of the end-to-end connection is. If *c'* equals to zero, the connection cannot be used for the particular service.

6.3.3 Methodology for Evaluating the Quality of a Multicast Service

In a multicast environment, the connection quality from a single source to all receivers has to be separately estimated. Here, the quality of the multicast distribution tree is proposed as the average of all single end-to-end connection qualities, thus, the quality *q* can be calculated as:

$$q = l \times \frac{c_0 + c_1 + \dots + c_n}{n} \qquad \text{Equation 6.5}$$

In this equation, l is a weighting factor in the range $0 \le l \le 1$. For services which strongly depend on the reachability of all destinations, l is suggested to be 1 if $\forall c_i\colon c_i > 0$. If a single connection cannot be used, l should be set to 0. If the demands of the service are not so high, l may adopt other values in the shown interval as well.

If more than a single source is active in the particular service, the calculation shown in equation 6.5 has to be repeated for all these sources.

6.4 Efficiency of Multicast

To further advance the deployment of IP multicast, many consider it necessary to provide a quantitative measure of the potential benefit of employing multicast rather than unicast [22], [40]. This benefit is provided by the efficiency of IP multicast [144].

Implementing multicast has many implicit and explicit costs that must be considered, such as router resources and added network complexity [157].

Chalmers and Almeroth [40] specified a simple metric defining the advantage of multicast in comparison with unicast as:

$$\delta = 1 - \frac{\text{multicast hops}}{\text{unicast hops}} \qquad \text{Equation 6.6}$$

This multicast metric will be a fraction in the range of $0 \le \delta \le 1$. It specifies the savings of transmissions over single network links. If the value equals to zero, no benefit is gained in using multicast. The higher the value becomes, the higher is this benefit.

One key factor that has to be taken in consideration with the cost for multicast traffic is the group size. Chuang and Sirbu have proposed a cost function, also known as the Chuang Sirbu Law, which defines the relationship between hop counts and group size [43]:

$$\frac{L_m}{L_u} = N^k \qquad \text{Equation 6.7}$$

L_m is the total length of the multicast distribution tree, L_u is the average unicast routing path, N is the multicast group size, and k is a scaling factor in the range $0 \le k \le 1$ describing the achievable efficiency. The interesting point made by the authors was that for the majority of topologies investigated k was nearly 0.8.

Chalmers and Almeroth [40] used this approach to provide a formula estimating the advantage of multicast depending on the group size N:

$$\frac{\text{multicast hops}}{\text{unicast hops}} = \frac{L_m}{(L_u)(N)} \approx \frac{N^{0.8}}{N} = N^{-0.2} \qquad \text{Equation 6.8}$$

And thus:

$$\delta \approx 1 - N^{-0.2} \qquad \text{Equation 6.9}$$

The discussed approaches provide a numerical estimation for benefit of using multicast. The result can be applied to some SLA between a customer and an ISP. In general, it is quite impossible to determine the size of a particular group [126]. Admittedly, multimedia transmissions cover most of the multicast connections. The mechanism of the commonly used transport protocol RTP provides some aids to achieve information about the current receivers [174]. In the context of quality of service based SLAs, this multicast metric has to be combined with various QoS metrics. At least, it is recommended to accommodate metrics for a maximum end-to-end delay and a maximum jitter to the agreement.

It has to be considered that different interests in the efficiency of multicast over unicast exist. Three parties are involved in a multicast transmission:

- End users

  Typically, it is meaningless for the end users whether the delivered service has been transported by multicast or by unicast. They just demand a high quality of the transmission. If an end user participates at a video conference, he can be esteemed as a content provider and the following rules for the content providers can be applied.

- Content providers

  If multicast is employed, the total number of simultaneous connections is dramatically reduced resulting in a much smaller load of the servers and the network connection of the content providers.

- Network providers

  The network providers must consider the demands of their customers. Additionally, they must ensure an optimal utilization of their resources such as network links, routers, etc. The usage of multicast reduces the required bandwidth for a particular service. Therefore, the network providers have to apply a new cost models (the current models are based on the utilized bandwidth) which also consider the actual cost of implementing and managing a multicast network.

6.5 Examples for SLAs

Basically, all the SLAs and the used metrics should be based on the demands of actually implemented services. Two examples are provided which introduce the mechanisms of performance and quality metrics as well as the concepts of service level agreements.

6.5.1 Video Conference

A first example is shown in figure 6.3. Three customers are connected to an ISP. They all completed the same SLA with the ISP with the following content:

- minimum throughput: 1 Mbps
- maximum delay: 200 ms
- maximum jitter: 80 ms
- maximum packet loss ratio: 1%
- minimum reliability: 99.9999%

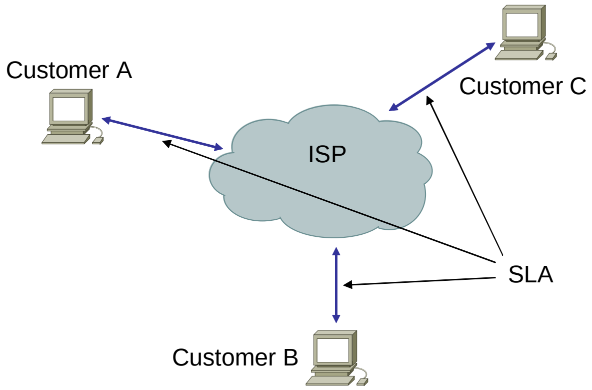


Figure 6.3 - SLAs for a Video Conference

The values defined by the SLA must be compared to the achieved (measured) values in order to have an instrument for verification of the promised service quality. A possible solution to measure the available quality of service is the proposed multicast quality monitor (compare section 7).

6.5.2 TV Broadcast

A second example is a TV broadcast. The scenario is shown in figure 6.4. In this case, two different SLAs have been specified. The service provider A agreed with the ISP upon the following values:

- minimum throughput: 5 Mbps
- maximum delay: 1 sec
- maximum jitter: 500 ms
- maximum packet loss ratio: 0.1%
- minimum reliability: 99.999%

It is obvious that the specified values differ strongly from the first example. The reason is the behavior of the TV broadcast. On the one hand the delay and the jitter can be relatively high due to the non-real-time character of the service which typically includes a play-out buffer. On the other hand the minimum throughput and the maximum packet loss ratio are tougher defined depending on the high quality video content.

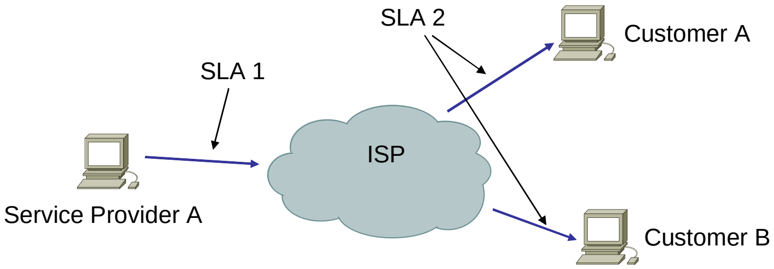


Figure 6.4 - SLAs for a TV Broadcast

In this example, the customers also have completed a SLA with the ISP. For instance, they might have specified some minimum throughput or a minimum reliability of the network connection.

7 Multicast Quality Monitor (MQM)

The multicast quality monitor (MQM) is a new approach to measure the reliability and the quality of service of an IP multicast network. The MQM is designed to work in an intra-domain as well as in an inter-domain environment. The basic concepts have already been presented at various conferences [70], [71].

The goal of the MQM is to enable a network administrator as well as an end user to locate and to isolate failures and bottlenecks in an IP multicast network. In order to achieve this task, the multicast quality monitor includes functions to measure the multicast connectivity between several nodes. Additionally, the quality of service of these connections can be estimated.

The MQM defines its own measurement protocol allowing to test the reachability. The gathered information can be used to calculate QoS parameters such as the delay. In addition, RTP has been chosen for most of the quality measurements. Based on self-made RTP streams an active estimation of the connection quality can be started. The presented approach also allows to use RTP streams of active multicast services such as a video conference in order to compute the test results.

This sections is organized as follows: First, the principles and the basic structure are discussed. Secondly, the different measurement methodologies are described detail. A third part provides information about the involved processes, the communication between them, and the used protocols. In the following the collection and the analysis of test results are discussed. Results of some sample measurements and a short outlook to future enhancements summarize this section.

7.1 Working Principles

The multicast quality monitor is intended to be used in all kinds of multicast networks. A basic principle is to avoid the distinction between measurements in an intra-domain environment, such as campus or enterprise networks. The same is intended for an inter-domain environment, for example between several universities interconnected by any number of ISPs.

7.1.1 Basic Structure

The MQM consists of a number of measurement probes, which are distributed over the network in order to achieve information about the quality of service within the network. There are two prospects to build such an environment:

(1) Centralized

A central management station is operating the global management. The distributed measurement probes wait for commands from the controlling machine in order to execute these commands. An example of the centralized approach is shown in figure 7.1. In general, the control traffic is independent of the measurement traffic. The measurement results are transferred to the management station where the evaluation of the results is started. Examples of this operation principle are the multicast reachability monitor and the multicast beacon. Both rely on commands initiated by a single controlling process.

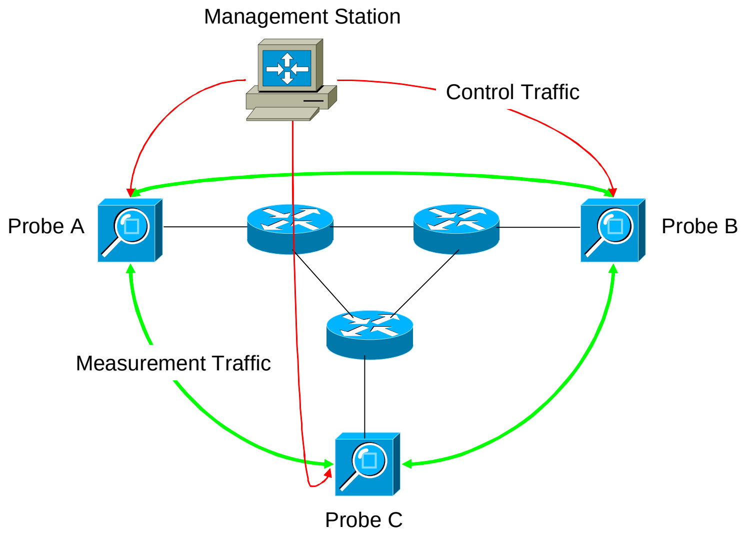


Figure 7.1 - Centralized Control of Measurement Probes

Unfortunately, this approach has a few drawbacks. First, the scalability has to be questioned. The core has to send commands to all measurement probes as well as to collect the results. Even if it may be possible to control a large number of probes simultaneously, the data collection will collapse. This is the case if a small number of transmissions towards to the management station occur at the same time due to the limited number of resources to this machine. Secondly, a single point of failure exists. The measurement is stopped, if the management machine is failing or if the network connection towards to a probe is broken down.

(2) Distributed

Completely autonomously working probes are distributed over the network. This scenario is shown in figure 7.2. The same connections are used for the measurement and the control traffic. Using this approach, it is much easier to allow a proper scaling as well as some independency from network failures.

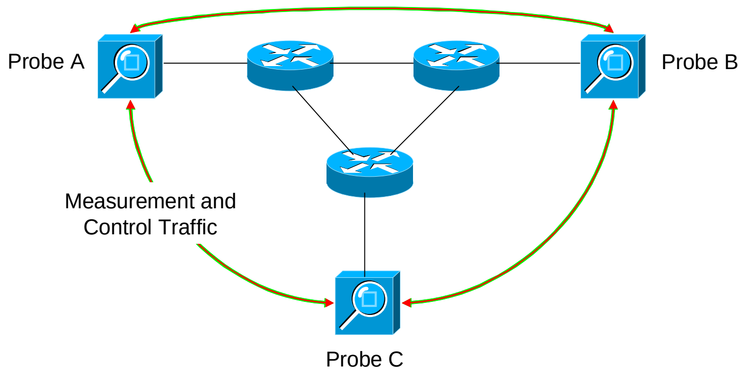


Figure 7.2 - Distributed Control of Measurement Probes

> Nevertheless, it can become difficult to coordinate the measurement between the autonomous probes, if dynamically initiated measurements should take place. Synchronization mechanisms have to be introduced, which for themselves may show scalability problems.

The multicast quality monitor is a hybrid form of both concepts. The tests are executed using intelligent probes which work completely independent from each other as well as from a central management process. This independency of the management probes from a controlling host provides a high degree of flexibility and robustness.

Nevertheless, such a management process exists allowing the initiation of dynamic tests. In order to estimate the quality of service within the multicast network, several measurements have to be performed. Some of these require to insert high bandwidth data streams for testing purposes. To decrease the impact on the network, or, at least, to minimize it, such measurements should only take place for a minimum time. A static configuration of all the probes would not allow such actions. Therefore, a central control mechanism is being used.

7.1.2 Differentiation of Measurements

The multicast quality monitor distinguishes between two basic measurement goals:

(1) Reachability and Reliability

Basically, the reachability is tested by sending ping packets from a source towards a destination (unicast) respectively towards to multiple destinations (multicast). If the request is answered by receiving a response message, the network is able to transfer packets in both directions between the source and the destination(s). Therefore, the destination host is reachable.

If the test is periodically performed, the reliability of the network connection can be estimated. For example, if 20 of 25 ping packets have been answered, the reliability of this particular connection is 80%. The more single tests have been executed, the more meaningful results of the reliability measurement can be calculated.

A reachability test always requires minimal resources such as CPU time, memory, and available bandwidth in the network. Therefore, the measurement is intended to run on end systems as well as on routers in order to achieve a complete view of the behavior of the global network.

(2) Quality of Service

The measurement of the quality of service of a network connection has a completely different behavior. Depending on the expected information, diverse tests have to be performed. The measurements should be coordinated with the behavior of the typical applications.

For example, the delay can be measured using a single test or a number of tests utilizing a low bandwidth. Admittedly, the results cannot be adapted to a high bandwidth video transmission. Due to growing queue lengths, the delay might dramatically increase if the network gets congested.

The same principles apply to all other QoS measurements as well. The behavior of the network changes if the utilization increases.

Such quality measurements typically require much more resources of the systems running the test. Therefore, this part of the MQM is planed only for running on end system. The routers could achieve this task too but they are assigned to forward packets as fast as possible not to introduce and to an analyze high bandwidth data streams.

7.1.3 Beacon Mechanism

Typically, all the measurements are initiated by a network administrator. The reliability measurement is intended to endlessly run if it has been started once. Different from reachability tests, which consume only a few resources from the network, the measurements of the quality of service should only take place for a short time in order to minimize the impact on the other multicast services.

Thus, the MQM provides a mechanism to dynamically start and stop the QoS measurements. This procedure is named the beacon mechanism. So called beacon messages are sent to the distributed probes. The beacons contain commands initiating actions at the measurement systems. Examples are the start of a high bandwidth data stream in order to analyze the behavior of the network in a congested situation.

The beacon mechanism is explained in more detail in the following parts of this section.

7.2 Reliability Measurements

The test of the connectivity between two hosts is achieved by a simple ping-response procedure. In order to differentiate between common network failures and a malfunctioning multicast forwarding, the reachability has to be tested for the unicast as well as for the multicast connection. For the unicast case, the ping mechanism described in section 3.5.1 is being used.

7.2.1 Multicast Ping

Originally, there is no mechanism available for the test of the reachability between nodes in a multicast network. A new approach is described in the following: the MQM ping mechanism.

Measurement probes are used to periodically transmit MQM ping packets to a configured multicast group. Additionally, they are listening for incoming MQM ping messages. If such a message has been received, it is examined in order to distinguish between requests and responses. If it is a request, a response is created and sent back towards to the source, i.e., to the same multicast group.

The format of a MQM ping message can be very simple. In order to analyze the reachability measurement, the receiver of a MQM ping response message requires three parameters: the IP address of the originator of the MQM ping request, the IP address of the source of the received response, and its own IP address. The latter two can be determined easily. Only the first one is missing. Therefore, the IP address of the originator is put into the MQM message.

Due to the principles of IP multicast, it is required to ping everyone from everywhere since it is not possible to use the information of A reaches B and C, and, on the other hand, both, B and C, reach A via IP multicast to provide any information about the connection between B and C. This is true for IP unicast as well, but in IP multicast everyone gets each response but cannot detect the complete state of the network using these messages.

7.2.2 Methodology

The MQM ping mechanism is shown in figure 7.3. For a single test, the probe A sends a MQM ping request packet (1) to all the others (probe B and probe C) using a well known multicast address. The other probes receive this request (2) and respond (3) back to the originator by sending a MQM ping response which is received (4) by the requesting probe A.

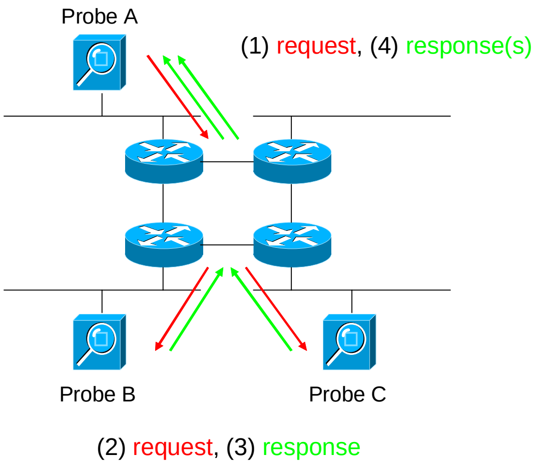


Figure 7.3 - MQM Ping Mechanism

Analyzing the received messages, the following propositions can be formulated:

- Host A knows that the bidirectional reachability between A and B is given.
- The same applies to the connection between A and C.
- Host B received a packet from A, therefore the unidirectional path from A to B is OK.
- Host C knows about the unidirectional reachability from A to C.

The typical lifetime for a forwarding state entry in a multicast router is about three minutes. If no multicast packet was seen during this time, the entry is timed-out and removed from the internal forwarding table of the router. Together with the knowledge that IP multicast is not a reliable protocol which ensures the proper delivery of each packet, one minute has proven to be a good choice for the period of sending MQM ping messages. A higher ping rate would result in an unnecessary congestion of the network.

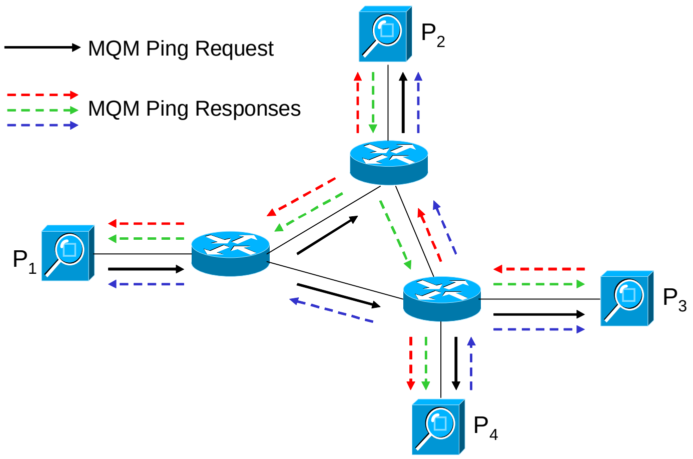


Figure 7.4 - Saving MQM Ping Messages

Focusing on the scalability of this approach, it seems to be unworthy to suspect any problem. A single packet per minute seems to be a very low data rate. Nevertheless, first, the number of probes might reach a critical value. Secondly, due to the principles of multicast, all the ping messages are multiplied by the number of receivers which are sending a response message. For example, if only ten probes have been distributed over the network, the ten MQM ping requests result in 90 response messages.

Based on these results, new paradigms have to be considered. Figure 7.4 shows the basic behavior of a single MQM ping request.

Utilizing the multicast mechanisms, single messages can be saved. In the shown example, probe P_1 is sending a MQM ping request to all other participating systems (black). P_2, P_3, and P_4 are answering by sending MQM ping response messages to the same multicast group (green, red, blue). Using the results of this single test (only P_1 has sent a MQM ping request), the following propositions can be formulated:

- Host P_1

  P_1 sent a request to P_2 (black) and received a response (green)

  P_1 sent a request to P_3 (black) and received a response (red)

  P_1 sent a request to P_4 (black) and received a response (blue)

  Conclusion: all other participating probes are reachable

- Host P_2

  P_2 received a request from P_1 (black)

  P_2 received a response from P_3 (red)

  P_2 received a response from P_4 (blue)

  Conclusion: all other probes may unidirectional reach P_2

- Host P_3

  P_3 received a request from P_1 (black)

  P_3 received a response from P_2 (green)

  P_3 received a response from P_4 (blue)

  Conclusion: all other probes may unidirectional reach P_3

- Host P_4

  P_4 received a request from P_1 (black)

  P_4 received a response from P_2 (green)

  P_4 received a response from P_3 (red)

  Conclusion: all other probes may unidirectional reach P_4

Collecting these information at a central point allows to complete the analysis in order to get a global reachability graph.

If mechanisms can be found to synchronize the MQM ping requests, it would be possible to dramatically reduce the number of measurement packets. In our example, four probes are involved. If all of them send requests, the total number of measurement packets is 16 (4 requests + 3 x 4 responses). In the optimum case, as shown above, only 4 messages are sent (1 request + 3 x 1 responses).

The here proposed approach is preventing unnecessary MQM ping requests. If a probe has received at least 2 requests within the last period, it suppresses its own request message. Because the probes are typically started at random times, this concept shows a good behavior. In the shown example, the total number of messages would be reduced to 8 (2 requests + 3 x 2 responses).

7.3 Quality of Service Measurements

The estimation of the quality of service using the MQM is divided into two measurement methods:

(1) One-way delay, round-trip time

The delay measurements are based on the MQM ping mechanism. In order to allow this kind of measurement, the protocol for the MQM packets has been extended to include a number of timestamps required for the calculation of the delay values.

(2) Packet loss ratio, ratio of reordered and duplicated packets, jitter

The measurement of these quality parameters is using the RTP protocol. Each header of a RTP packet already includes a sequence number and a timestamp. Both values can be used in order to estimate the ratio of lost, reordered, and duplicated packets as well as of the variation of the delay, the jitter. Additionally, RTP has chosen because most multimedia transmissions already use RTP for their data transfer. Therefore, the received packets of these applications can be used for the calculations as well as the self-initiated packet streams.

7.3.1 One-Way Delay, Round-Trip Time

Basically, it has to be distinguished between the measurement of the one-way delay (OWD) and the estimation of the round-trip time (RTT). The latter is much easier to discover and no synchronization between the clocks of the involved systems is required. Nevertheless, the OWD is the more meaningful value. Especially, this is true for transmissions of multimedia content. Typically, such transfers are unidirectional. For example, a single source is sending a video towards to a number of receivers. The backchannel is only used for control information and requires much less quality than the video stream itself.

7.3.1.1 Methodology

In addition to the IP address of the originator of a MQM ping request message, timestamps are embedded in the MQM message. In order to measure the different delay values, the following information has to be provided in the MQM packet:

- IP address of the originator of the request
- sending timestamp taken at the originator of the request
- sending timestamp taken at the sender of the response

The measurement of the delay values requires an accuracy of microseconds. In order to retrieve correct values for the one-way delay, highly synchronized clocks are necessary. GPS clocks can be used to achieve this task.

The evaluation of the gathered information is following a few basic rules. Let $T_O(X)$ be the sending timestamp from the originator of the MQM request, $T_R(X)$ be the timestamp from the originator of the response message, and $T_L(X)$ the timestamp taken at the reception of this response packet. Given a packet from A to B to A, the one-way delays can be calculated as follows:

$$OWD_F(A, B) = T_R(B) - T_O(A) \qquad \text{Equation 7.1}$$

$$OWD_S(B, A) = T_L(A) - T_R(B) \qquad \text{Equation 7.2}$$

$OWD_F(A,B)$ stands for calculations based on the MQM ping request messages and $OWD_S(A,B)$ uses the information in a MQM ping response.

The round-trip time from A to B can be determined using this equation:

$$RTT(A, B) = OWD_F(A, B) + OWD_S(B, A) \qquad \text{Equation 7.3}$$

$$RTT(A, B) = (T_R(B) - T_O(A)) + (T_L(A) - T_R(B)) = T_L(A) - T_O(A) \qquad \text{Equation 7.4}$$

It can be seen that the round-trip time depends only on timestamps taken at a single host. Therefore, the accuracy of the calculated delay always is much higher than the estimated one-way delay times.

The one-way delay values calculated of response messages can be used to additionally rate the other round-trip times using the following equation:

$$RTT'(A, B) = OWD_S(A, B) + OWD_S(B, A) \qquad \text{Equation 7.5}$$

$$RTT'(A, B) = (T_L(A) - T_R(A)) + (T_L(B) - T_R(B)) \qquad \text{Equation 7.6}$$

The term $T_L(A)$ - $T_R(A)$ only depends on timestamps taken at host A. The same applies to the term $T_L(B)$ - $T_R(B)$, which only uses timestamps taken at host B. Therefore, the synchronization of the clocks of both hosts is not required for this kind of calculation of the round-trip time. Admittedly, it has to be considered that the single one-way delay measurements occurred at different times. The behavior of the network might have changed during this time. Due to the assumed period of MQM ping messages of about one minute, the results can be considered as correct.

7.3.1.2 Example

In order to explain the mechanisms in more detail, the example of the last section is used again. The single transmitted packets are presented in figure 7.5.

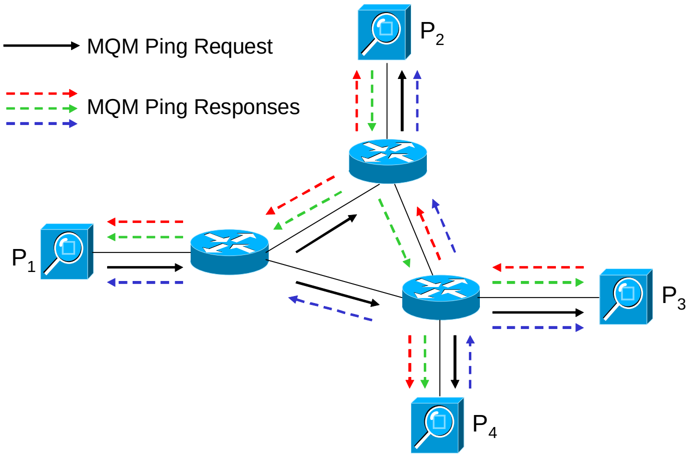


Figure 7.5 - Calculating Delay Values using a single MQM Ping Request

To recapitulate: P_1 has sent a MQM request message to all participating probes. P_2, P_3, and P_4 answered to the request by sending a MQM ping response packet.

After the collection of all the data, the following calculations can be executed using the presented equations:

- $OWD_F(P_1,P_2)$; $OWD_S(P_2,P_1)$; $RTT(P_1,P_2)$
- $OWD_F(P_1,P_3)$; $OWD_S(P_3,P_1)$; $RTT(P_1,P_3)$
- $OWD_F(P_1,P_4)$; $OWD_S(P_4,P_1)$; $RTT(P_1,P_4)$
- $OWD_S(P_2,P_3)$; $OWD_S(P_3,P_2)$
- $OWD_S(P_2,P_4)$; $OWD_S(P_4,P_2)$
- $OWD_S(P_3,P_4)$; $OWD_S(P_4,P_3)$

Therefore, in addition to the already shown results, the following delay values can be calculated:

- $RTT'(P_2,P_3) = OWD_S(P_2,P_3) + OWD_S(P_3,P_2)$
- $RTT'(P_2,P_4) = OWD_S(P_2,P_4) + OWD_S(P_4,P_2)$
- $RTT'(P_3,P_4) = OWD_S(P_3,P_4) + OWD_S(P_4,P_3)$

7.3.1.3 Error Analysis

Concerning the quality of the calculated results, potential sources of errors have to be analyzed. During the calculation of the delay, the largest error which can be introduced is based on a low synchronization quality of the clocks of the different hosts. The quality of the estimated OWD can be increased using GPS synchronized clocks.

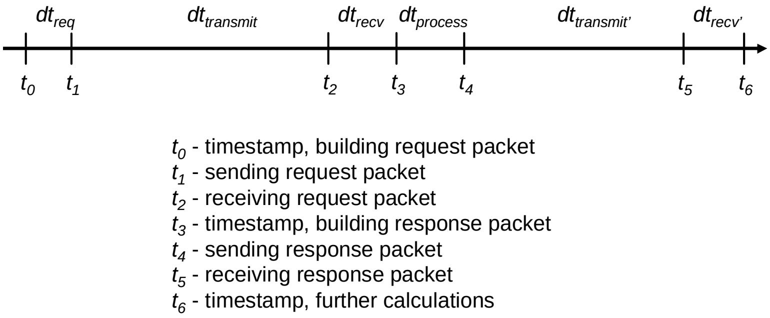

dt_{req}
$dt_{transmit}$
dt_{recv}
$dt_{process}$
$dt_{transmit'}$
$dt_{recv'}$
t_0
t_1
t_2
t_3
t_4
t_5
t_6
t_0 - timestamp, building request packet
t_1 - sending request packet
t_2 - receiving request packet
t_3 - timestamp, building response packet
t_4 - sending response packet
t_5 - receiving response packet
t_6 - timestamp, further calculations

Figure 7.6 - Time Bar showing the Events during the Delay Measurement

Another possible source of errors is the time it takes from creating a timestamp until sending the packet as well as from receiving a packet until the new timestamp is taken. Figure 7.6 shows a time bar explaining the effect. Admittedly, it can be considered that:

$$(dt_{req}\ , dt_{process}\ , dt_{recv} \sim 1\ \mu s) << (dt_{transmit} \sim 100\ \mu s) \qquad \text{Equation 7.7}$$

On a typical host, the processing time is much smaller than 1 µs but the worst case scenarios has been used. Therefore, the difference between the processing and the transmission times is so large that a failure introduced by a little varying processing time can be ignored.

7.3.2 Packet Loss Ratio, Ratio of Reordered and Duplicated Packets

The measurement of the packet loss ratio as well as of the ratio of reordered and duplicated packets is based on a continuous stream of RTP packets. The header of each packet includes a sequence number which has to be analyzed.

7.3.2.1 Methodology

The methodology can be separated into two parts: The operation mode specifies the working principles of the quality of service measurement, how to retrieve RTP packets. The analysis defines methods to compute QoS parameters out of the collected information.

The multicast quality monitor distinguishes between two operation modes:

(1) Passive measurement

The concept of the passive measurement is to consume information about the quality of a particular data transfer without actively sending packets into the network. Using the basic principles of IP multicast, a measurement probe can join the same group on which the transmission occurs.

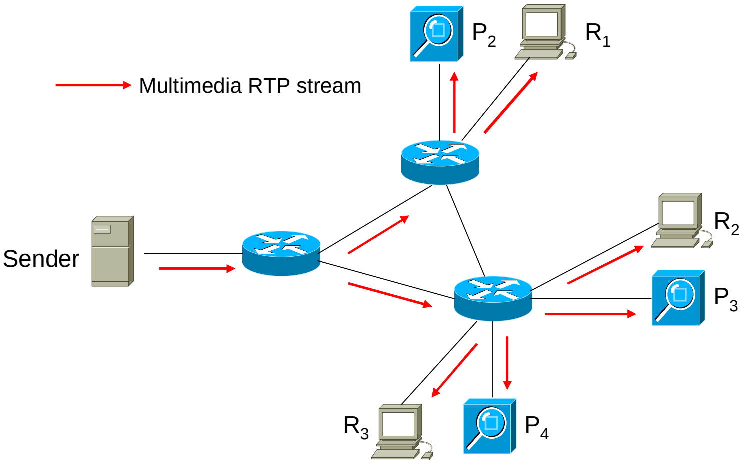


Figure 7.7 - Passive QoS Measurement using foreign Data Streams

The principles of this method are shown in figure 7.7. A sender is transmitting some multimedia content towards to a number or receivers (R_1, R_2, R_3). The same transmission is intercepted by three measurement probes (P_2, P_3, P_4).

Typically, the multicast applications use RTP to transmit their data in conjunction with RTCP. The QoS measurement is designed to analyze both protocols in order to achieve as much information as possible.

Even if this measurement method is called passive, there might be still some impact on the network. If a probe joins a multicast group with active senders at some place in the network where no other receiver is located, it requests the network to transmit the multicast data towards to its location. And, therefore, it unnecessarily stresses the network components on the data path.

(2) Active measurement

The active measurement works very similar to the passive one. The difference is that no active sender exists, or that for any reason this traffic is not intended to be used for measurements. Therefore, the measurement environment has to worry about creating usefully RTP streams itself.

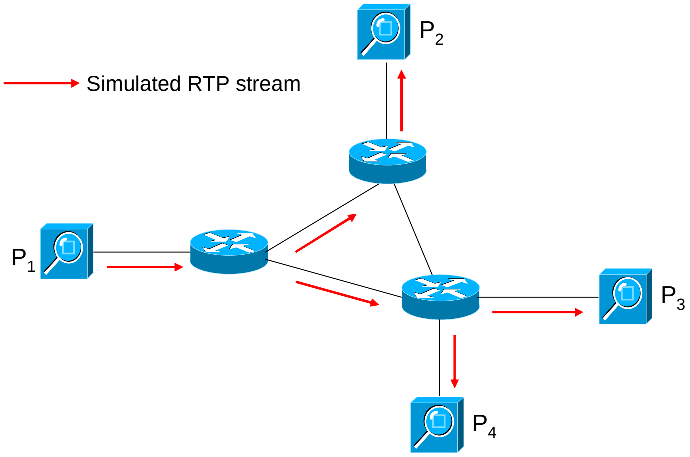


Figure 7.8 - Active QoS Measurement using simulated Data Streams

An example is shown in figure 7.8. The multimedia stream of the previous example is replaced by a simulated data stream transmitted by probe P_1. The other probes (P_2, P_3, P_4) are still receiving and analyzing the RTP packets.

7.3.2.2 Analysis

All the parameters, the packet loss ratio, the ratio of reordered packets, and the ratio of duplicated packets can be gathered using the sequence number in each packet of a RTP stream. According to the working principles of most multimedia applications, the following bases of calculation are used, whereas S_n is the sequence number of the n-th received packet:

Packet loss ratio: A number of packets is supposed to be lost, if the sequence number of the received packet differs from the sequence number in the last seen packet by more than one. Or more formal, if $S_{n+1} > S_n + 1$. The packet loss ratio is defined as the percentage of lost packets to the number of expected packets over a period of time. If packets are reordered, they are expected to be lost.

Ratio of reordered packets: Packets are supposed to be reordered, if the sequence number of the received packet is smaller than the expected one, or if $S_{n+1} < S_n$.

Ratio of duplicated packets: Some consecutive packets are supposed to be duplicated, it the sequence number of the received packet is equal to the sequence number in the last seen packet, or if $S_{n+1} = S_n$.

An exception is the overflow of the sequence number. This event has to be separately handled.

The passive measurement allows to use the RTCP control traffic from each participant of the multimedia transmission to get information about the packet loss ratio in the network. The receiver of RTP traffic is required to periodically send feedback information towards to the sender. This feedback is called a RTCP receiver report (RTCP RR). Using IP multicast, the receiver reports are sent to the same multicast group address as the multimedia content.

The RTCP RR includes the current packet loss ratio from the source towards to the particular receiver. Therefore, analyzing the RTCP traffic allows to gather information about the connection quality from the sender towards to each participant.

7.3.2.3 Error Analysis

The measurement of the packet loss ratio, the ratio of reordered or duplicated packets only depends on the sequence number in the packets of a RTP stream. Therefore, no additional failure can be introduced by the measurement probes.

7.3.3 Jitter

The same concepts used for the estimation of the packet loss ratio are applied to the measurement of the jitter.

7.3.3.1 Methodology

A constant packet flow is necessary for the jitter measurements. Additionally, each packet must contain a sending timestamp allowing to calculate the interval which elapsed between the transmission of two consecutive packets.

The header of a RTP packet also includes a timestamp, thus RTP streams are eavesdropped and analyzed. The source of these RTP packets can be an active application as well as a measurement probe which is simulating the behavior of a multimedia session. Therefore, the same methodology is used as described in the last section.

7.3.3.2 Analysis

Two definitions how to calculate the jitter can be found in the literature (section 3.2.3). The first one is based on the interarrival times of consecutive packets. The distance between successive packets is calculated in order to estimate the variance of this interarrival time. The advantage of this method is the independency of external clocks. All timestamps required by the calculation are locally taken. Admittedly, a continuously packet flow is necessary with the following properties:

- the packets are created and sent at with an equal interval
- no pauses are allowed in the transmission
- a packet loss ratio of zero must be assumed

A packet flow with a constant bit rate (CBR) is required. Unfortunately, most multimedia applications use a variable bit rate (VBR) in order to save resources. For example, a still picture requires less transmission capacity than a moving image. A primary goal of the QoS measurements using the MQM is to consume as less resources of the network as possible. Therefore, this kind of jitter calculation cannot be used.

The second definition of the jitter, which is used by the MQM, is specified as the variance of the delay of successively received packets. Single delay measurements are executed and the jitter is calculated as the average offset of the mean delay.

The average delay D_A is calculated using the single delay values D_i for each received packet i and a total number n of collected packets:

$$D_A = \frac{1}{n}\sum_{i=1}^{n} D_i \qquad \text{Equation 7.8}$$

The jitter J_i of each single packet i is defined as:

$$J_i = |D_A - D_i| \qquad \text{Equation 7.9}$$

Using this formula, the average jitter J_A and the maximum J_{max} can be calculated as follows:

$$J_A = \frac{1}{n}\sum_{i=1}^{n} J_i \qquad \text{Equation 7.10}$$

$$J_{\max} = max(J_i) \qquad \text{Equation 7.11}$$

The MQM follows this definition of the calculation of the jitter, which is recommended by RFC 1889 [182] as well. According to RFC 1889 a real-time approximation of the jitter can be calculated as follows: Let S_i be the RTP timestamp of packet i and R_i the arrival time for this packet, then for two packets i and j, the difference D may be expressed as:

$$D(i,j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i) \qquad \text{Equation 7.12}$$

The jitter J is calculated continuously as each data packet i is received using the difference D for this and the previous packet i-1 according to the formula:

$$J = J + \frac{(|D(i-1, i)| - J)}{16} \qquad \text{Equation 7.13}$$

This algorithm is the optimal first-order estimator and the gain parameter 1/16 gives a good noise reduction while maintaining a reasonable rate of convergence [37].

7.3.3.3 Error Analysis

The calculation of the jitter depends on the accuracy of timestamps taken at the sending host as well as on such taken at the measurement probe. Differences of timestamps are the basis for the computation and times based on the same clock source are always subtracted. Therefore, the synchronization of the clocks of the involved hosts is of less relevance.

The variance of the duration from receiving a packet and taking the timestamp can introduce a computation failure. As shown in section 7.3.1.3, this error is small enough to be ignored.

7.4 Processes and Inter-Process Communication

The multicast quality monitor consists of a number of different processes. The jobs of each one are explained in the following. Additionally, the different tasks of the inter-process communication (IPC) are illustrated.

7.4.1 MQM Processes

The following processes are specified in the context of the MQM:

- MQM sender
- MQM receiver
- RTP sender
- RTP receiver
- RTCP receiver
- monitor
- manager

Each of those can run nearly independently on the same as well as on different hosts. The number of coexisting processes on a single probe is restricted. These constraints as well as the basic working principles of each process are separately explained in the next few subsections.

7.4.1.1 MQM Sender

The MQM sender is being used to send periodically MQM ping messages to a specific multicast group address. Various properties of the behavior of the sent packet stream can be configured:

- IP multicast address and port number
- TTL
- packet size
- time interval

The process is logging all sent packets in order to allow a more complete analysis of the measurement results. These entries are logged:

- sequence number
- local time
- packet size

Only a single MQM sender is allowed on each measurement system.

7.4.1.2 MQM Receiver

The MQM receiver is intended to watch for any kind of MQM messages on a configurable multicast group. Each received packet is analyzed and the appropriate action is started based on the message type. The following events may occur:

- Reception of a MQM ping request

  If a MQM ping request is received, first, a timestamp is taken. Secondly, the request is being answered by sending a MQM ping response message including the new timestamp and the IP address of the source of the request packet. Finally, a log entry is created containing the following information:

  - IP address of the originator
  - timestamp of the request
  - local time
  - packet size

- Reception of a MQM ping response

  A MQM ping response message finalizes a complete reachability test. Therefore, no new message is created. Nevertheless, a reception timestamp is taken and the event is logged. The following information are saved:

  - IP address of the originator
  - timestamp of the request
  - IP address of the responder
  - timestamp of the response
  - local time
  - packet size

- Reception of a beacon message

  A beacon message is used to dynamically start and stop quality of service measurements based on the analysis of RTP streams. If a beacon is received, it is scanned for commands targeting the receiver of the message. If matching commands are found, according actions are performed. An overview of the available commands is provided in table 7.1. Each command includes the parameters for the multicast group (G) and port number (P). The parameter hold time specifies the running time of the started process in seconds.

Command	Precondition	Action
start RTP sender on group G/port P	no active RTP sender on G/P	a RTP sender on G/P is started, the hold time is initialized.
	a RTP sender on G/P is already running	the hold time is updated
stop RTP sender on G/P	no active RTP sender for G/P	-
	a RTP sender for G/P is running	this RTP sender is stopped
start RTP receiver on G/P	no active RTP receiver on G/P	a RTP receiver on G/P is started, the hold time is initialized
	a RTP receiver on G/P is already running	the hold time is updated
stop RTP receiver on G/P	no active RTP receiver for G/P	-
	a RTP receiver for G/P is running	this RTP receiver is stopped
start RTCP receiver on G/P	no active RTCP receiver on G/P	a RTCP receiver on G/P is started, the hold time is initialized
	a RTCP receiver on G/P is already running	the hold time is updated
stop RTCP receiver on G/P	no active RTCP receiver for G/P	-
	a RTCP receiver for G/P is running	this RTCP receiver is stopped

Table 7.1 - Available Commands in a Beacon Message

The configurable properties of the MQM receiver are the IP multicast address and the port number. Only a single MQM receiver is allowed to run on a host.

7.4.1.3 RTP sender

The RTP sender is intended to initiate a continuous flow of packets. The sender can be manually started as well as by a beacon command. Each packet contains a RTP header and therefore, a sequence number and a sending timestamp. The behavior of the RTP sender can be configured in various ways:

- IP multicast address and port
- TTL
- packet size
- time interval

In addition to the RTP packets, the RTP sender is periodically transmitting RTCP sender reports (SR), which is requested by RFC 1889. The SR contains information about the number of already transmitted packets. Additionally, a timestamp is included allowing the receiver to calculate the delay between both hosts.

Similar to the MQM sender, this process is logging the following information about all sent packets in order to allow a complete analysis of the measurement results:

- multicast group address
- RTP header information (version, flags, payload type)
- sequence number
- local time
- packet size

Additionally, each sent RTCP SR is logged as well. The number of RTP sender processes per host is not limited.

7.4.1.4 RTP receiver

After receiving a packet, the RTP receiver, whose first action was to get a current timestamp, is decoding the information in the RTP header. The following information about the RTP packets are included in each log entry:

- multicast group address
- IP address of the sender
- RTP header information
- sequence number
- timestamp of the sender
- local time
- packet size

The number of simultaneously running RTP receivers is not limited.

7.4.1.5 RTCP receiver

The RTCP receiver is intended to catch all RTCP packets allowing a later analysis. The RTCP receiver does not distinguish between sender reports and receiver reports. All received RTCP packets are logged. There is no limit for the number of coexistent RTCP receiver processes.

7.4.1.6 Monitor

The primary job of the monitor is to supervise the locally running processes and to listen for requests of a manager process which is collecting the measured data. If such a request is received, the monitor advises all running measurement processes to rotate the log files. Afterwards, it transfers every file to the manager which has not yet been transmitted. All successfully passed files are moved to a backup directory.

7.4.1.7 Manager

The manager process fulfills two tasks. First, it is responsible for collecting the measurement results. Secondly, the beacon messages are created and sent by the manager process.

7.4.1.7.1 Collection of Measurement Results

In order to gather the measurement results from all the probes, the manager is required to separately setup management connections to the monitor processes at the measurement stations. TCP has been chosen for the transport protocol, because it offers a reliable data transfer. Typically, only one probe is queried at a time to minimize the utilization of the network towards to the host running the manager. The manager saves the collected data in an appropriate format for further investigation.

7.4.1.7.2 Beacon Messages

Beacon messages are sent by the manager process in order to initiate single RTP based quality of service measurements. Each beacon message may contain an arbitrary number of single commands destined for individual probes.

The goal of this mechanism is to allow the manager to address individual probes using an universally valid message. All the beacons are transmitted to the same multicast group as the MQM ping messages are.

An example of the beacon method is shown in figure 7.9. The following beacon message is sent to all active probes:

```
mesg for probe A: start a RTP sender on group G
mesg for probe B: start a RTP receiver on group G
mesg for probe B: start a RTCP receiver on group G
```

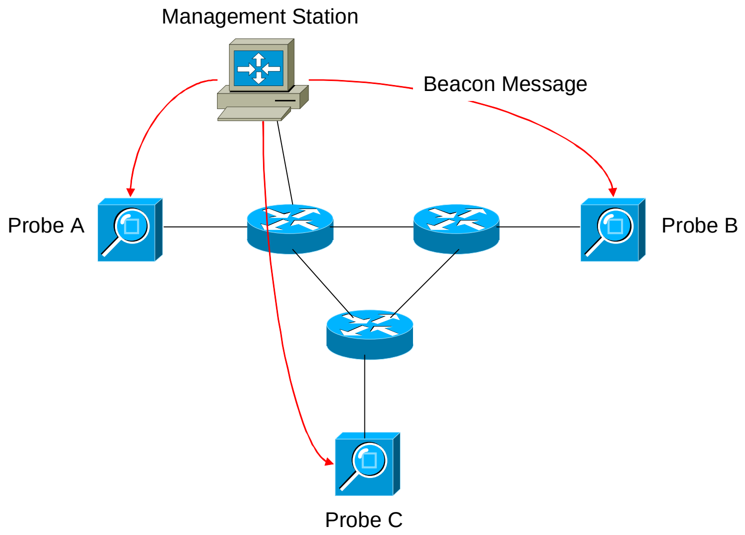


Figure 7.9 - The Beacon Message is sent to all Probes

Due to the principles of IP multicast, a packet can be lost. Therefore, the beacon messages have to be periodically repeated until the hold time is expired.

Figure 7.10 shows the scenario after the first successfully transmitted beacon message. Probe A started a measurement data flow and probe B is receiving and analyzing these data. Additionally, the beacon message is repeated until the hold time has been expired.

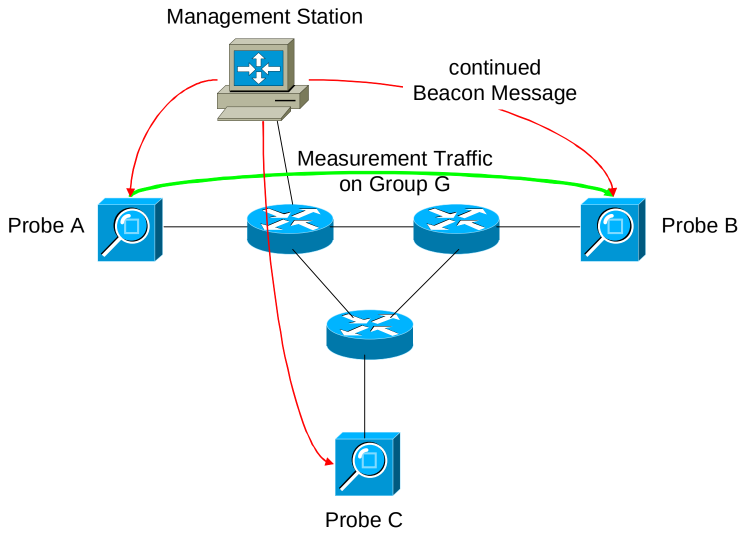


Figure 7.10 - The initiated Measurement and the repeated Beacon

In addition, a possibility to immediately stop measurements is required. This can be achieved by creating a beacon message containing the appropriate stop commands.

7.4.2 Inter-Process Communication

There are a number of communication paths between the different processes. Basically, the IPC can be organized in four categories:

- Measurement data transmitted between the probes

  MQM ping messages as well as RTP streams are originated, received and analyzed by the measurement probes.

- Transfer of the measurement results to the manager

  The test results have to be transferred to a common place in order to perform a complete analysis of the behavior of the multicast network, especially of the quality of the different network connections.

- Control information between the monitor and other local processes

  The monitor is responsible for transferring the measurement results to the manager. Before this action can be executed, the single processes have to be informed in order to rotate their log files.

- Beacon messages to initiate new measurements

  A dynamic behavior of the measurements can be achieved using these beacon messages. Measurements can be started and stopped on behalf of the manager process.

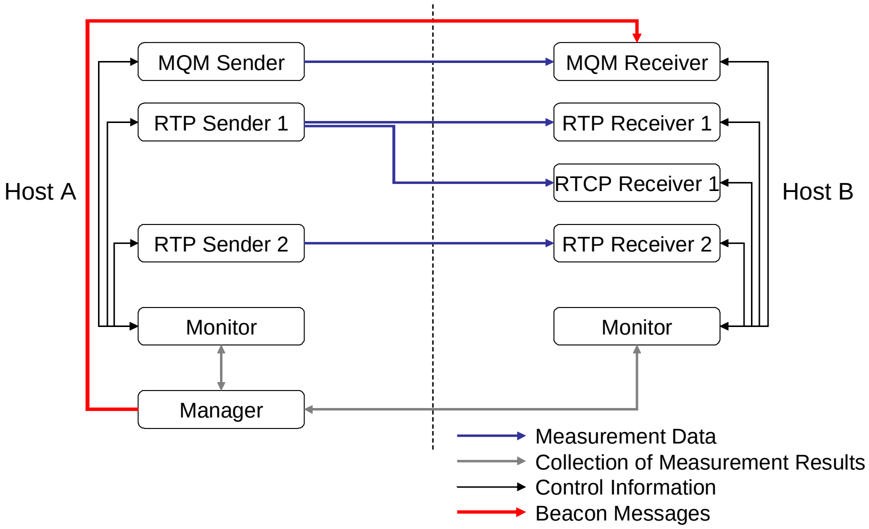


Figure 7.11 - Communication Paths between the different Processes

A summary of all the communication paths between the different processes is shown in figure 7.11. The collection of the measurement results is done using TCP as the transport protocol. The measurements themselves as well as the transmission of the beacon messages are required to use UDP. Depending on the kind of measurement, RTP encoded packets are sent or the MQM protocol defined in the next subsection is used. The specification of the MQM does not define the details how to implement the communication between the monitor and the other local processes.

7.5 Communication Protocols

The network communication between the processes of the multicast quality monitor depends on three mechanisms. Each relies on a different protocol. The measurement of the jitter and the packet loss ratio is based on RTP. A description of RTP and RTCP is provided in appendix A. The measurement results are collected out of band using a simple file transfer mechanism.

All other communications are based on the MQM protocol which is explained in this section. Basically, the concept of the MQM protocol is based on the multicast reachability monitor [11].

7.5.1 MQM Header

Each MQM message contains a MQM header which is shown in table 7.2.

Position [byte]	Length [byte]	Description
0	1	version
1	1	padding
2	2	type

Table 7.2 - MQM Header Format

Version:

> The field describes the version of the specification of the MQM protocol. The first prototypical implementation defined this value to 1. Currently, a student is working on a second, more complete implementation, which is using the version number 2.

Padding:

> The padding field is a place holder for further specifications. The value must be set to zero.

Type:

> The type specifies what kind of MQM message is encapsulated by the MQM header. The following types are defined in version 1 of the MQM:
>
> - MQM ping request (type 0)
> - MQM ping response (type 1)
> - MQM beacon (type 2)

7.5.2 MQM Ping Message

The MQM ping request as well as the MQM ping response uses the same packet format which is shown in table 7.3.

Position [byte]	Length [byte]	Description
0	4	MQM header
4	4	IP address of the originator
8	8	timestamp of request
16	8	timestamp of response

Table 7.3 - MQM Ping Message Format

MQM header:

The MQM header contains type MQM ping request or MQM ping response.

IP address of the originator:

This is the IP address of the originator of the MQM ping mechanism. This address is inserted by the receiver of the ping request message.

Timestamp of request:

The timestamp inserted by the originator. The first four bytes of the timestamp is the number of seconds elapsed since January, 1st 1970. The time zone must be UTC (universal time). The second four bytes define the microseconds of the current second.

Timestamp of response:

The timestamp is taken at the receiver of the request message.

7.5.3 MQM Beacon Message

The MQM beacon message is sent in order to instruct remote measurement probes to start or to stop measuring the quality of service using RTP streams. A beacon message may contain several commands at once. The packet format is shown in table 7.4.

Position [byte]	Length [byte]	Description
0	4	MQM header
4	2	hold time
6	2	message length
8	4	target probe address 1
12	4	multicast group address 1
16	8	command 1
...	...	...
x	4	target probe address n
x+4	4	multicast group address n
x+8	8	command n

Table 7.4 - MQM Beacon Message Format

MQM header:

The MQM header contains type MQM beacon.

Hold time:

The hold time specifies the duration of validity of this beacon. Thus, all processes, which have been started due to the reception of this beacon, have to be terminated if the hold time has been expired.

Target probe address n:

This is the IP address of the target probe which is the intended receiver of the command.

Multicast group address n:

The multicast group used for the particular measurement is being defined in this value.

Command n:

The command defines the action which should be performed by the target probe. The structure of this field is specified as follows:

1 bit:	start (1) / stop (0) a process
7 bit:	TTL to be used for the transmission
8 bit:	process type:

- RTP sender (type 0)
- RTP receiver (type 1)
- RTCP receiver (type 2)

16 bit:	UDP port number
16 bit:	data rate in number of packets per second
16 bit:	packet size

7.6 Sample Measurements

A first implementation of the multicast quality monitor has been created to verify the theoretical concepts. An overview of this prototypical implementation which has been used for these sample measurements is provided in appendix D.

The two basic measurement methods have been separately examined. First, the tests of the connectivity, which are using the MQM ping protocol, have been executed and the achieved delay values are presented. Secondly, RTP based QoS measurements were started. Distributed measurement probes joined an active video broadcast in order to analyze the transmission quality from the source of this multimedia traffic towards to the measurement probes.

7.6.1 MQM ping based Tests

Four hosts have been used for the MQM ping based delay measurements. The network configuration is shown in figure 7.12. In order to achieve comparable and meaningful results, measurements in a local campus network have been accomplished as well as between different sites in the German research network. In the shown example, hosts in Erlangen, Regensburg and Bayreuth were involved. The number of hops between the particular hosts are represented in the figure.

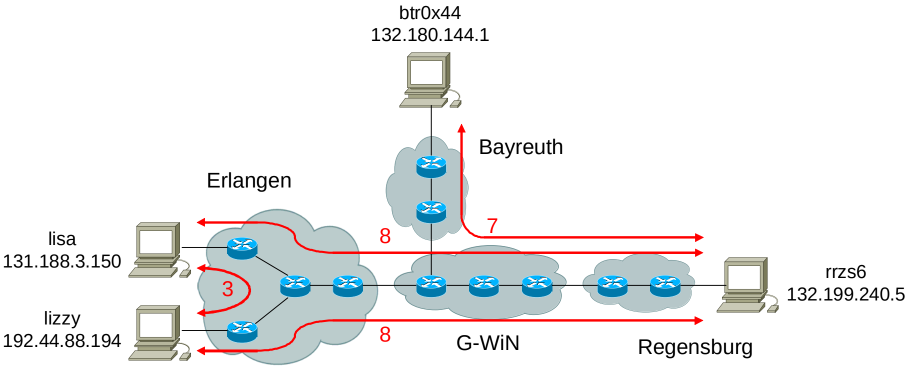


Figure 7.12 - Network Configuration used for the MQM based Tests

Figure 7.13 shows the measurement results between the hosts lisa and lizzy. Both systems are located within the campus network of the University of Erlangen-Nuremberg. The delay closely oscillates around 1.8 ms. A single peak of about 17 ms has been discovered. The reason could be a temporary high load of a involved router.

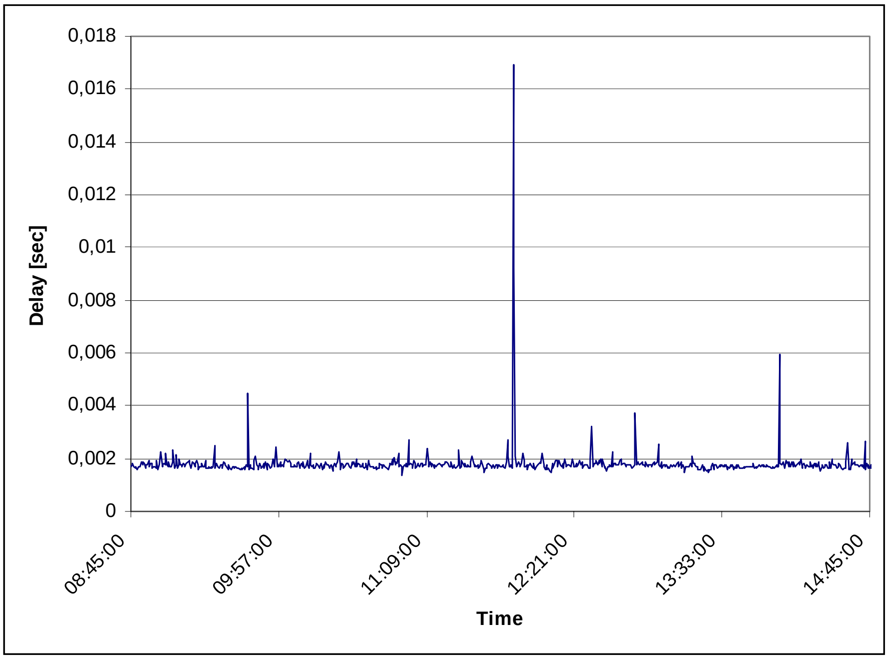


Figure 7.13 - MQM ping within the local Network in Erlangen

The results of a test between Erlangen (lisa) and Regensburg (rrzs6) are shown in figure 7.14. The mean round-trip time was 12 ms. This is a rather good value for connections over the internet. The high peaks (up to 250 ms) are introduced by the high number of involved routers.

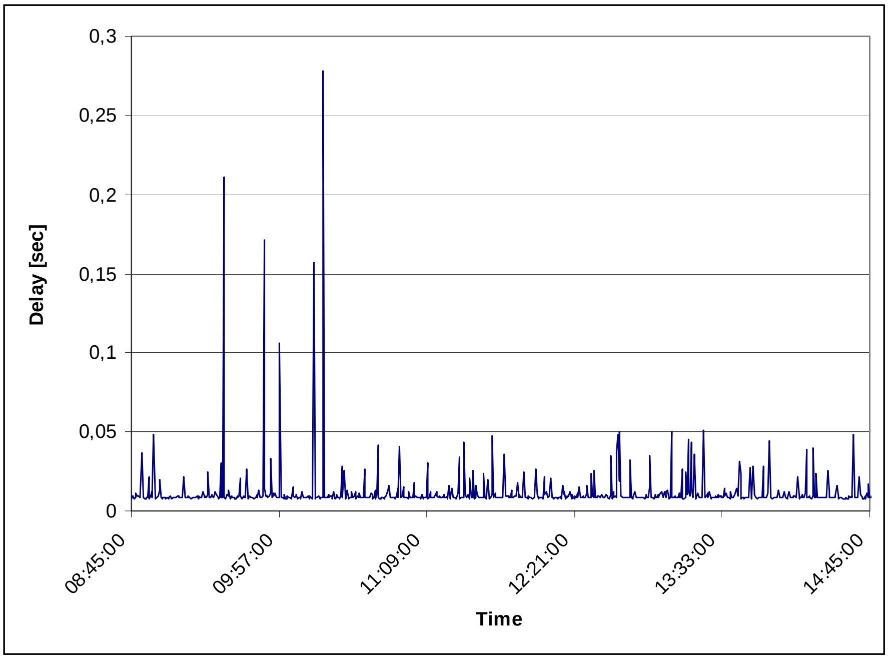


Figure 7.14 - MQM ping between Erlangen and Regensburg

The last measurement has been repeated using a slightly different scenario. This time, the MQM ping request has been initiated in Regensburg (rrzs6) and the transmission delay to a host in Erlangen (lizzy) has been measured. The results, which are shown in figure 7.15 are comparable to the last test. Once again a mean value of 12 ms has been discovered.

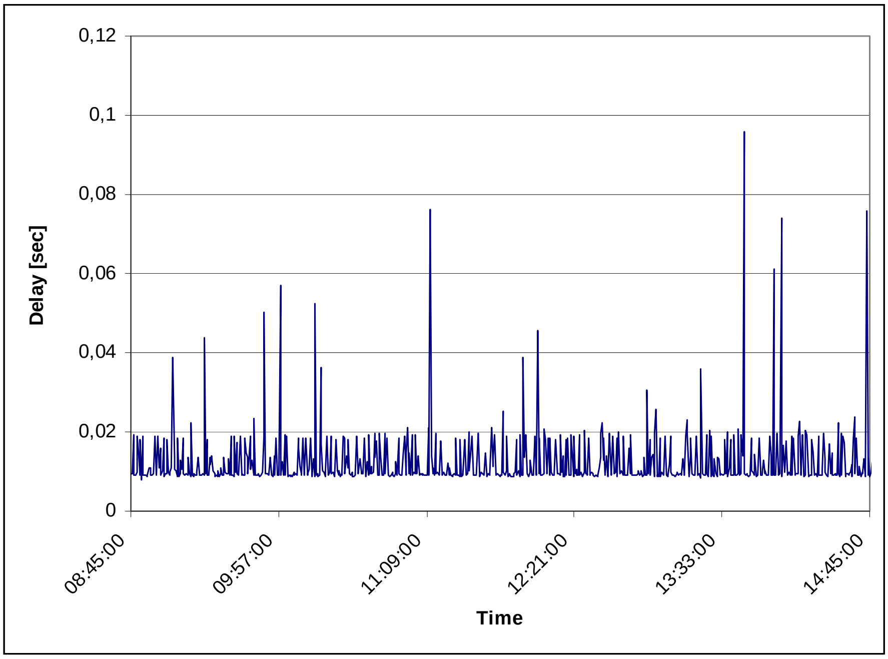


Figure 7.15 - MQM ping between Regensburg and Erlangen

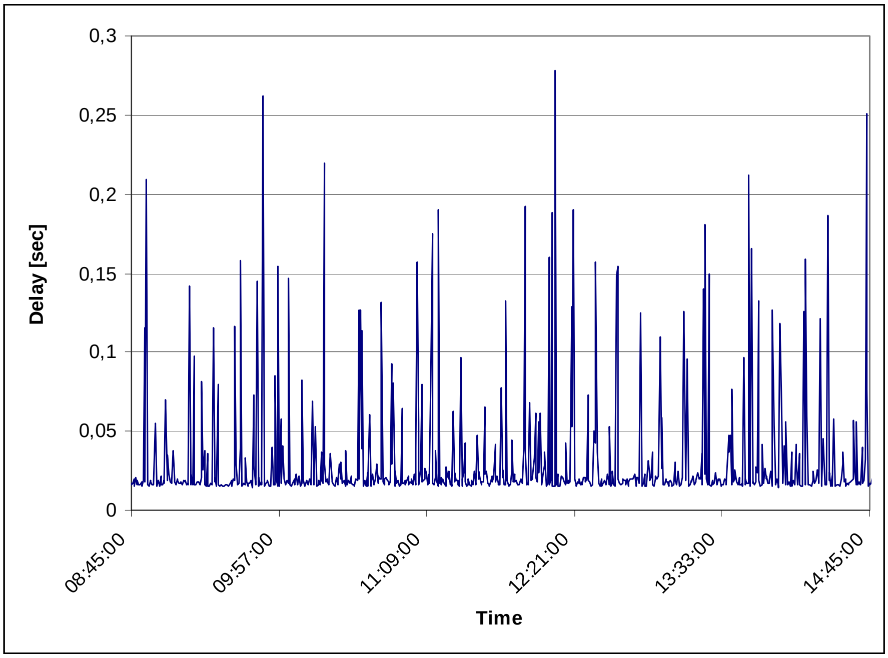


Figure 7.16 - MQM ping between Regensburg and Bayreuth

A final test has been started between Regensburg (rrzs6) and Bayreuth (btr0x44). It has been determined that the connection quality between Regensburg and Bayreuth is worse than the quality between Erlangen and Regensburg. The mean round-trip time between the involved probes was 31 ms. The complete measurement results are provided in figure 7.16. The primary reason is the less efficient connection of the campus network of the University of Bayreuth and the G-WiN.

Always, the multicast connectivity should be compared to the unicast connectivity in order to distinguish between failures of the network itself and problems in forwarding multicast traffic.

To compare the behavior of the unicast tests using ICMP messages and this of the here defined multicast ping mechanism, two test series have been performed.

First, the round-trip time between two hosts in Erlangen has been measured using both the ICMP ping and the MQM ping mechanism. The results are shown in figure 7.17. The mean delay of 1.9 ms using the MQM ping seems to be much higher than the achieved ICMP result of 0.2 ms. The reason for this difference is the low precision of the unicast ping. The command was used on a Sun workstation running Solaris 2.6. The ping tool has a resolution of 1 ms. Therefore, most single measurements resulted in a delay of zero, which significantly reduced the mean value of this test.

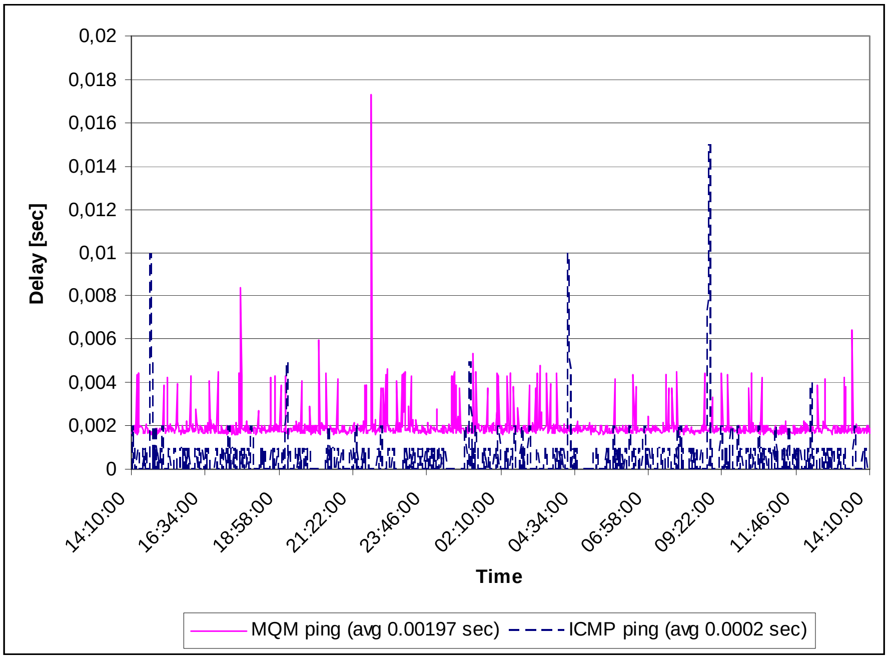


Figure 7.17 - Unicast vs. Multicast Ping (within the Local Campus Network)

A second test has been accomplished between Regensburg and Bayreuth. The results of the measurements are shown in figure 7.18. It can be seen that the curves of both tests show a similar behavior. Additionally, the mean values for the measured round-trip times are equal (32 ms). The high variance was caused by queuing effects at the involved routers.

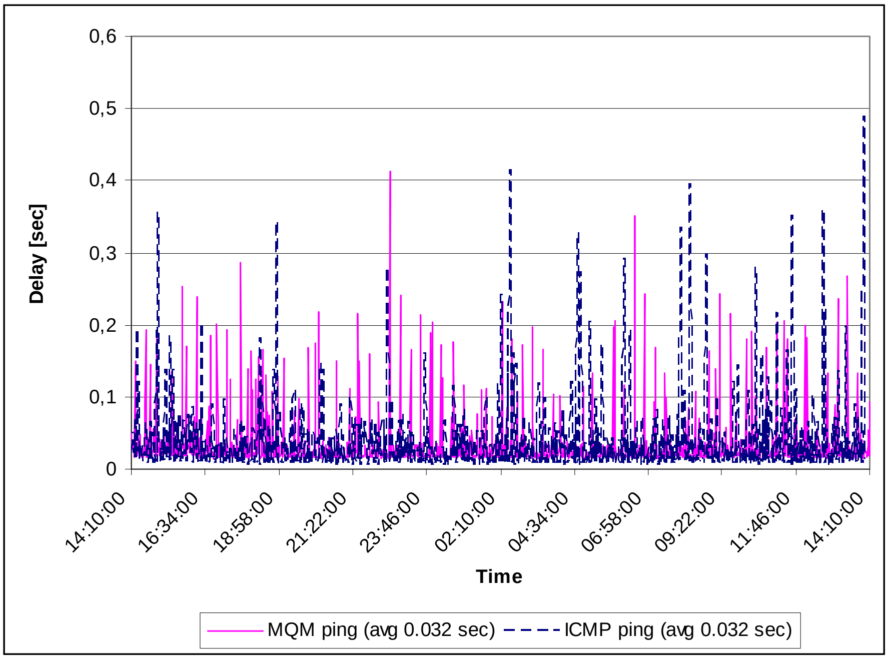


Figure 7.18 - Unicast vs. Multicast Ping (between Regensburg and Bayreuth)

All the executed tests have shown the expected behavior. During the tests, a real network failure was discovered. There was no multicast reachability between Erlangen and Bayreuth. The unicast connectivity was proven by using the ICMP based ping. The scenario is shown in figure 7.19.

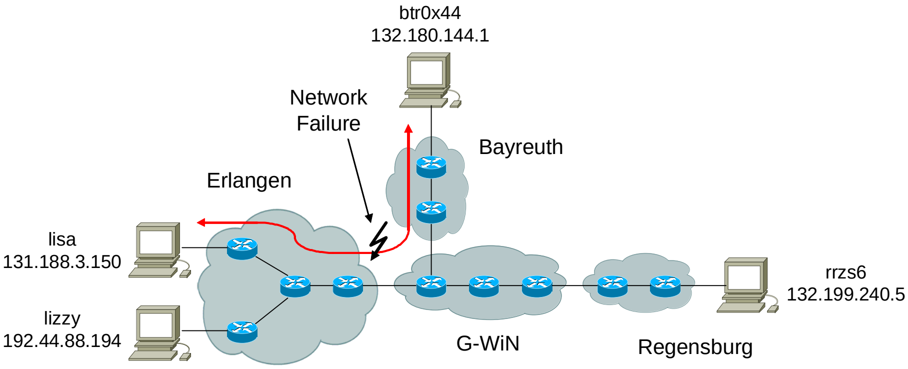


Figure 7.19 - Failure of Multicast Forwarding

We were able to locate the failure using the results of the MQM ping tests as well as some additional tools [194]. After this analysis, the problem was quickly repaired. This coincidence allowed a verification of the function of the reachability tests by using the multicast quality monitor in a real environment.

7.6.2 RTP based Tests

The RTP based tests have been executed during the world soccer championship in Korea. A server at the University of Erlangen-Nuremberg was streaming the TV transmission into the internet using the popular multicast tool vic for the video broadcast. A maximum data rate of 1 Mbps has been configured. Several distributed instances of the multicast quality monitor were used to analyze the transmission quality from the server towards to the particular probe.

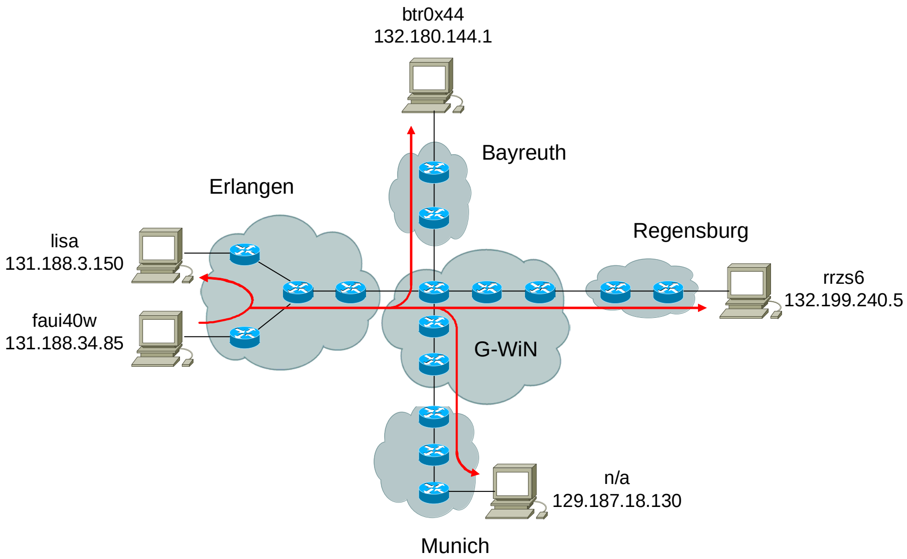


Figure 7.20 - Network Configuration used for the RTP based Tests

A small excerpt of the used network configuration is provided in figure 7.20. The server (faui40w) is shown on the left side of the picture. All other hosts displayed in the figure are running an instance of the MQM.

In total, 102 participants, distributed all over Germany, have been joined the transmission. This number has been discovered by analyzing the received RTCP packets [192]. The prototypical implementation of the multicast quality monitor has been used for this task.

We repeated the test on several days in order to achieve comparable results. Two measurements were selected which are considered to present here.

The results of the first test are shown in figure 7.21 and figure 7.22 respectively. Both graphs show the distribution of the packet loss ratio during the measurement. On the abscissa, the elapse time represented by the number of received RTP packets is plotted. On the logarithmically scaled ordinate, the packet loss ratio is applied.

The first two results are shown in figure 7.21. The RTP stream containing the broadcasted video was analyzed by MQM instances in Erlangen and Munich. It was shown that the achieved quality of service is rather good at both locations. The mean packet loss ratio within the campus network in Erlangen was 10^{-5}. Three routers were involved in the transmission of the video. The obtained loss ratio in Munich was about 10^{-3}. The quality of service still was high enough for an uninterrupted video.

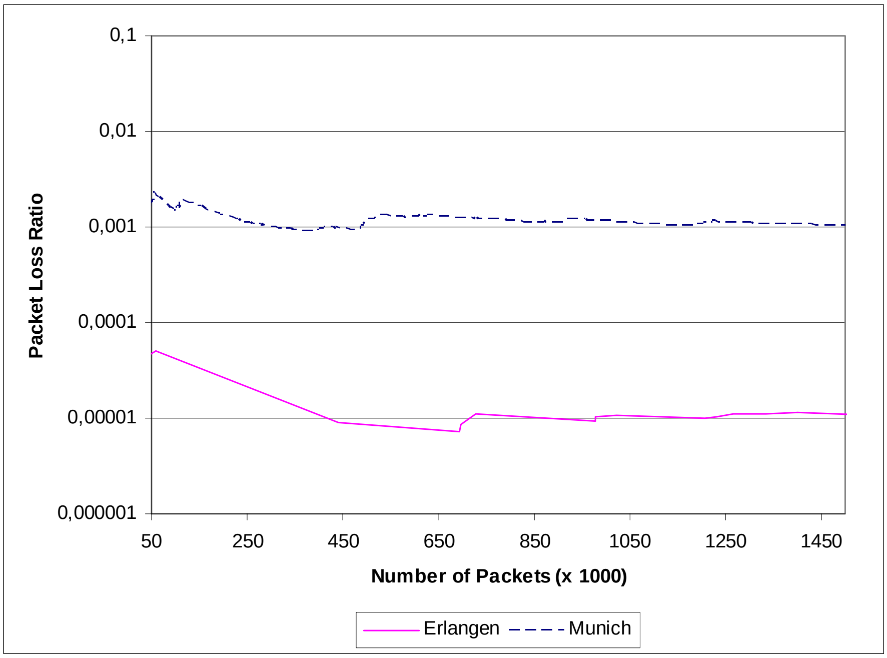


Figure 7.21 - Analysis of the RTP Stream in Erlangen and Munich

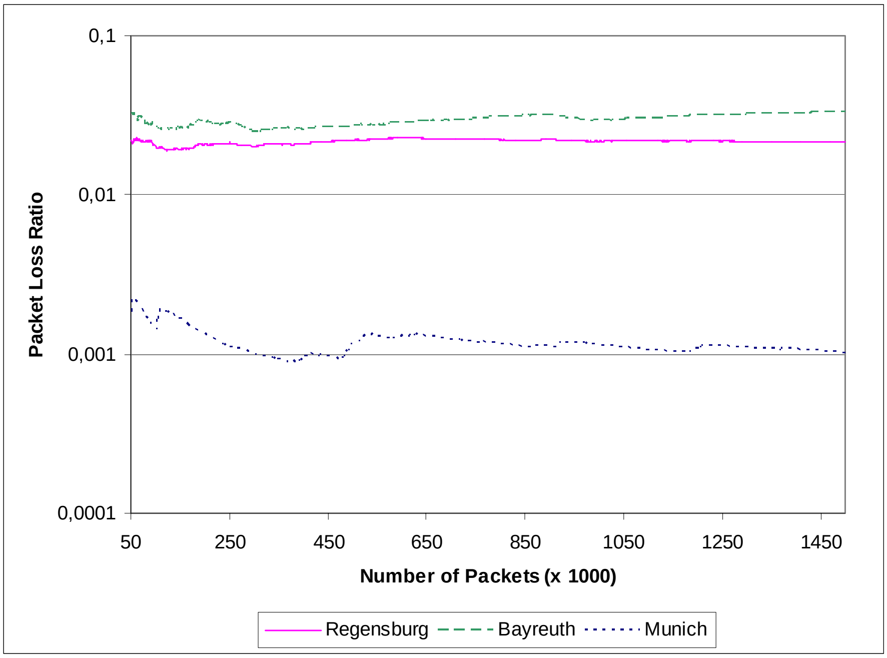


Figure 7.22 - Analysis of the RTP Stream in Regensburg, Bayreuth, and Munich

Figure 7.22 shows the achieved quality from the sender in Erlangen to receivers in Regensburg, Bayreuth, and Munich. The latter has been included in order to allow an easier comparison of the presented results. It was shown that the quality towards to Regensburg and to Bayreuth was not high enough for a proper video transmission. The instances of the MQM measured a packet loss ratio of 2 x 10^{-2} in Regensburg and of 3 x 10^{-2} in Bayreuth. Faulty pictures were the result of such a loss ratio.

The measurement results of a second test are shown in figure 7.23 and figure 7.24. The preferences of the video server have been the same as described before.

The achieved quality of service towards to receivers in Erlangen, Eichstaett, and Munich is presented in figure 7.23. The transmission quality of the video broadcast was high enough in all these cases. The measured packet loss ratio was 5 x 10^{-5} within Erlangen, 3 x 10^{-3} to Eichstaett, and 10^{-3} to Munich.

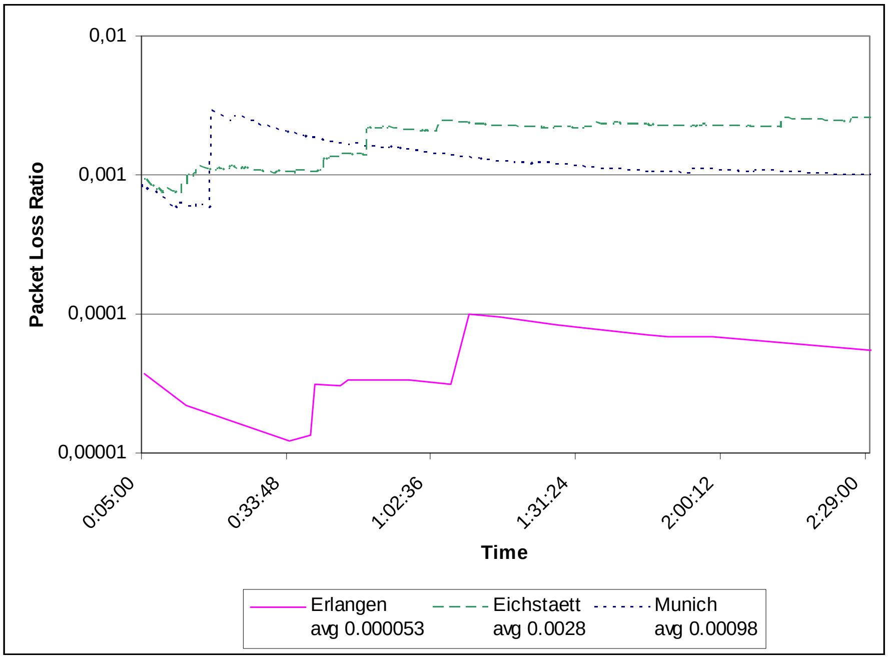


Figure 7.23 - Analysis of the RTP Stream in Erlangen, Eichstaett, and Munich

The remaining results are shown in figure 7.24. Presented are the measurement results for the packet loss ratio towards to Regensburg and to Bayreuth. The resulting quality of the video at both locations was moderate. For comparison reasons, the graph for Munich was included into the picture as well.

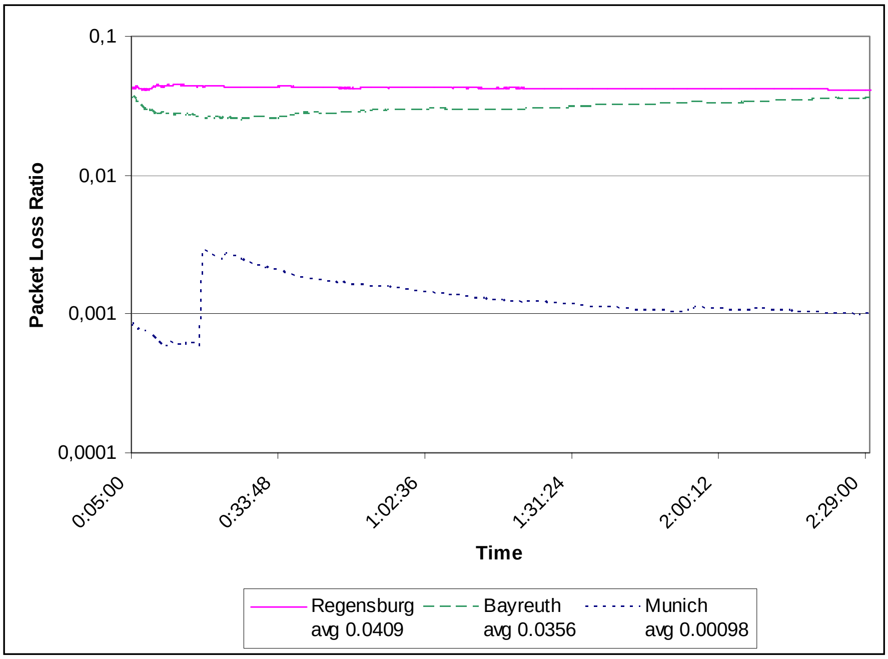


Figure 7.24 - Analysis of the RTP Stream in Regensburg, Bayreuth and Munich

In summary, the analysis of active RTP streams allowed a very good examination of the quality of service of a particular network connection without having an impact on the behavior of the network by introducing special measurement data.

7.7 Summary and Outlook

The current specification of the multicast quality monitor allows the test of the reachability of measurement probes in an IP multicast network as well as to measure the quality of service of the connections between them. The following goals can be realized using the MQM:

- Reachability measurement

  Using the MQM ping mechanism, it is possible to test the one-way as well as the bidirectional connectivity between various systems in an IP multicast network. Ping request messages are sent to a multicast group in order to solicit answers from the receivers of the message.

- Estimation of the reliability

  The reliability of single connections of a multicast network can be estimated by evaluating the results of periodically executed connectivity tests. The reliability gives an overview of the stability of the network as well.

- Delay measurement

Using the MQM ping mechanism, it is possible to calculate the latency of a transmission between particular end systems. In order to accomplish this task, each ping message contains timestamps of the originator and of the responder. The working principle of the ping mechanism allows to calculate the one-way delay as well as the round-trip time. The measurement of the one-way delay requires highly synchronized clocks of the involved systems.

- Measurement of the packet loss ratio

  Using RTP streams, the packet loss ratio can be measured. The header of each RTP packet includes a sequence number, which allows the discovering of missing individual packets. The concept of the multicast quality monitor allows to analyze packet streams of active multimedia transmissions as well as of self-initiated RTP streams by measurement probes.

- Estimation of the ratio of reordered and duplicated packets

  The evaluation of the ratio of reordered and duplicated packets follows the same principles as the measurement of the packet loss ratio: The sequence numbers of RTP packets are used to achieve this task.

- Jitter measurement

  Additionally, the jitter measurement is based on the analysis of received RTP streams. The timestamp information of two consecutive packets are examined in order to estimate the variation of the delay.

The multicast quality monitor has been developed with scalability questions in mind. Two potential problems have been envisioned and appropriate design decisions have been considered:

- Implosion of ping response messages

  Deploying the MQM, the number of measurement probes is increasing with the size of the network which has to be investigated. The MQM ping mechanism is required to be periodically executed to preserve the state information at the involved network components. Due to the working principle of IP multicast, each message is transmitted to every participant of the particular multicast group. Therefore, even a slight increasing of the number of measurement probes, and, thus, the number of sent MQM ping messages, the number of MQM ping responses is imploding. An approach has been provided allowing a dramatically reduction of this number. Only about two response messages are required per request.

- Too many coexisting high bandwidth measurement streams

  It has to be considered that each actively sent measurement packet has an impact to the behavior of the network. Particularly, this applies to high bandwidth RTP streams which are used in order to estimate the network expected quality of service for an upcoming video broadcast. Therefore, mechanisms have to be developed, which reduce the number of simultaneously running measurements. The presented multicast beacon mechanism allows to dynamically starting and stopping such measurements. Due to this central coordination of the single tests, the probability for unnecessary parallel measurements is reduced.

To conclude these considerations, an outlook on two open tasks is appended. First, currently, we only have an implementation for end systems. Especially the reachability measurement is a task which should be run on network components e.g. on routers. Secondly, another very important quality parameter is still unnoticed, the join and the leave latency.

In an IP multicast environment, applications can join any multicast group at any time. The network is responsible to achieve the appropriate tasks such as constructing a new multicast tree in order to transmit the data of each active sender to the new participant. The time taken between joining a group and the arrival of the first packets is called the join latency [3]. The users of TV broadcasts expect to see the movie at the same time they click on the according button.

A similar behavior is the leave latency. A supposed user zapps through multiple channels requesting data for each associated multicast group. If the leave latency is too high, the network transports data for all these groups to the end system. Therefore, the network utilization dramatically increases as well as the load of the network connection of the host.

In summary it can be said that a complete overview of the quality of service of an IP multicast network has to include information about the join and leave latency as well.

8 Summary

In the last few years the patterns of utilization of the internet have dramatically changed. The landscape of modern network applications is formed by multimedia services. Within this scope, the relevance of IP multicast is enormously increasing. Typical applications in a multicast environment are video conferences and broadcast transmissions of multimedia content. Already today, radio stations are well known who try to increase their transmission zone by using the internet. Additionally, TV broadcasts which are globally transmitted are a trend-setting domain in the area of IP multicasting.

All these applications require a reliable transmission service. Additionally, a very high transmission quality is necessary due to the demands of the multimedia content. It is the job of the internet service providers, the hardware manufacturers and the developers of new protocols and mechanisms to build networks which satisfy these requirements.

The analysis of current multicast networks clearly shows that there are a number of bottlenecks due to the high complexity of the protocols and of the configuration of the network components. Another problem is the missing and the faultily provision of the appropriate quality of service to selected applications respectively.

Originally, the internet has been designed for a best effort transmission, i.e., data are expected to be transferred as good as possible through the network. Even today, this mechanism is the basis for all IP networks. New approaches and developments allow to prioritize data in the network and to apply a appropriate handling to these packets. Other concepts try to guarantee a minimum end-to-end quality of service by reserving the required resources along the data path. Unfortunately, the implementation of these ideas is slowly preceding. Often, the promised quality of service parameters are not provided. Tools for an analysis of IP multicast networks are required.

The primary goal of this work was to define a framework which allows to perform an analysis of IP multicast networks. On the one hand, the reliability of multicast networks and the reachability between single systems via IP multicast should be determined respectively. The results have to be compared with tests of the general network, i.e., with the unicast connectivity in order to distinguish between global network failures and problems of the multicast forwarding. On the other hand, measurements of the transmission quality are mandatory required. The results of these tests allow the estimation of the currently achievable quality of service in the network as well as the prediction of the behavior of expected transmissions.

First, typical multicast networks were investigated in order to analyze potential bottlenecks. Further on, existing methods and tools for analyzing multicast networks were explored. It should be determined if these tools are able to solve the mentioned goals. It was emphasized that none of these methods allows a comprehensive overview of the reliability and the quality of IP multicast networks. Additionally, all these approaches show scalability problems. Really large networks cannot be examined using these instruments.

In the context of this work a new approach was presented which is designed for such measurements. The system named multicast quality monitor (MQM) specifies a measurement environment which allows the execution of the following examinations of IP multicast networks.

In order to test the reachability and the reliability a new multicast ping mechanism was introduced. It allows to analyze the connectivity of single measurement probes within an IP multicast network. The basic concept is to periodically send messages to a multicast group. All receivers are required to send an appropriate response. Based on the received packets, a reachability graph can be calculated. The periodically execution of this action is mandatory in order to refresh the entries in the multicast routing tables in all network components. The reliability can be calculated using the results of single reachability tests over a period of time.

The analysis of the quality of service in a multicast network relies on two different mechanisms. The delay measurement is based on the same methods which are used to estimate the reachability. In order to achieve the desired information, the measurement packets, which are encoded using the proposed MQM format, must include additional timestamp information. Such timestamps are taken at the sending of MQM ping request messages, at the reception of these packets, and at the reception of the associated MQM ping response message. Based on this information, it is possible to estimate the one-way delay as well as the round-trip time. Obviously, the calculation of the one-way delay depends on the time information of different hosts. More exact values can be determined if both clocks are highly synchronized. Typically, GPS receivers are used for this task.

The second part of the quality of service measurements is based on the examination of continuous data streams. RTP encoded packets are used which include important information for the measurement such as sequence numbers and timestamps. Based on these data, the variance of the delay, the jitter, the packet loss ratio, and the ratio of reordered and duplicated packets can be calculated. Sources for the RTP streams which have to be analyzed might be active multimedia transmissions. The advantage of this measurement method is to minimize the impact on the multicast network, because no additional measurement traffic has to be introduced. If currently no active transmission is available or if the parameters of an active one do not match the desired requirements such as the throughput, the MQM allows to simulate such transmissions. The reception and the analysis of the RTP stream remain unmodified.

One of the most important problems during the development of the MQM was the scalability. All the single mechanisms have been investigated by figuring out their behavior during the analysis of really large networks.

The MQM ping mechanism was designed in a manner that two ping request messages sent by the distributed measurement probes are enough for a complete analysis of the whole network. The response messages to both requests can be used to achieve information which allow the calculation of a complete reachability graph including the associated delay values.

It is recommended to use active multicast applications for the analysis of the transmission quality through the multicast network. Additionally, the specified beacon mechanism allows to dynamically starting and stopping tests, i.e., all RTP based measurements can be centrally controlled. This working principle ensures that no unnecessary measurement data streams have to be enabled. The impact on the network and, thus, on the active services can be minimized by using the beacon mechanism.

Basically, it is not possible to examine all conceivable network connections in the internet in order to analyze their reliability and the quality of service. A solution was proposed by specifying a model for multicast networks and multicast services. This model allows to detailed describe the network infrastructure of IP multicast networks. Based on the modeled information as well as on measurement data and on simulation results, optimum paths between several end systems can be calculated including the achievable quality of service. Additionally, the model allows to incorporate information about the used multicast services. Combined with this information, the analysis of the network behavior can be exacting predicted. Suggestions for optimal locations for the measurement probes as well as for suitable measurement methods are possible. The number of the single tests can be dramatically reduced while the information content is increasing.

In summary is can be stated that a framework was successfully specified which allows an extensive analysis of the transmission behavior of IP multicast networks. All typical quality of service parameters of recent multimedia applications were integrated. The scalability of the proposed system was ensured in a high degree. Additionally, the preconditions for enhancements were created in order to include future, currently unknown requirements.

A perspective to potential refinements should conclude this work. A broad application of the system in real networks is important for the deployment of the specified approach. Suitable facilities to access the results for the beneficiaries, the end user and the network administrators, are required as well. To achieve this task, the prototypical implementation has to be enhanced and new methods to query and to present the results have to be applied. A standardization of the proposed techniques and methods could help the here specified system to a wider usage.

A RTP Definitions

A few basic definitions according to RFC 1889 [182] have to be summarized in order to understand the mechanisms of RTP. Finally, the header formats of RTP, RTCP SR, and RTCP RR are described.

A.1 Basic Definitions

RTP payload: The data transported by RTP in a packet, for example audio samples or compressed video data.

RTP packet: A data packet consisting of the fixed RTP header, a possibly empty list of contributing sources (see below), and the payload data.

RTCP packet: A control packet consisting of a fixed header part similar to that of RTP data packets, followed by structured elements that vary depending upon the RTCP packet type. Typically, multiple RTCP packets are sent together as a compound RTCP packet in a single packet of the underlying protocol.

Port: The abstraction that transport protocols use to distinguish among multiple destinations within a given host computer. TCP/IP protocols identify ports using small positive integers. RTP depends upon the lower-layer protocol to provide some mechanism such as ports to multiplex the RTP and RTCP packets of a session.

Transport address: The combination of a network address and port that identifies a transport-level endpoint, for example an IP address and a UDP port. Packets are transmitted from a source transport address to a destination transport address.

RTP session: The association among a set of participants communicating with RTP. For each participant, the session is defined by a particular pair of destination transport addresses (one network address plus a port pair for RTP and RTCP). The destination transport address pair may be common for all participants, as in the case of IP multicast, or may be different for each, as in the case of individual unicast network addresses plus a common port pair. In a multimedia session, each medium is carried in a separate RTP session with its own RTCP packets. The multiple RTP sessions are distinguished by different port number pairs and/or different multicast addresses.

Synchronization source (SSRC): The source of a stream of RTP packets, identified by a 32-bit numeric SSRC identifier carried in the RTP header so as not to be dependent upon the network address. All packets from a synchronization source form part of the same timing and sequence number space, so a receiver groups packets by synchronization source for playback. Examples of synchronization sources include the sender of a stream of packets derived from a signal source such as a microphone or a camera, or an RTP mixer. A synchronization source may change its data format, e.g., audio encoding, over time. The SSRC identifier is a randomly chosen value meant to be globally unique within a particular RTP session. A participant need not use the same SSRC identifier for all the RTP sessions in a multimedia session; the binding of the SSRC identifiers is provided through RTCP. If a participant generates multiple streams in one RTP session, for example from separate video cameras, each must be identified as a different SSRC.

End system: An application that generates the content to be sent in RTP packets and/or consumes the content of received RTP packets. An end system can act as one or more synchronization sources in a particular RTP session, but typically only one.

Byte Order, Alignment, and Time Format: All integer fields are carried in network byte order that is, most significant byte (octet) first. This byte order is commonly known as big-endian. All header data is aligned to its natural length, i.e., 16-bit fields are aligned on even offsets, 32-bit fields are aligned at offsets divisible by four, etc. Octets designated as padding have the value zero. Wallclock time (absolute time) is represented using the timestamp format of the Network Time Protocol (NTP), which is in seconds relative to 0 h UTC on January 1, 1900 [146]. The full resolution NTP timestamp is a 64-bit unsigned fixed-point number with the integer part in the first 32 bits and the fractional part in the last 32 bits. In some fields where a more compact representation is appropriate, only the middle 32 bits are used; that is, the low 16 bits of the integer part and the high 16 bits of the fractional part. The high 16 bits of the integer part must be determined independently.

A.2 RTP Packet Format

The header of each RTP packet has the format shown in figure A.1.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure A.1 - RTP Header

The first twelve octets are present in every RTP packet, while the list of CSRC identifiers is present only when inserted by a mixer. The fields have the following meaning:

version (V): 2 bits

> This field identifies the version of RTP. The version defined by this specification is two (2). (The value 1 is used by the first draft version of RTP and the value 0 is used by the protocol initially implemented in the "vat" audio tool.)

padding (P): 1 bit

If the padding bit is set, the packet contains one or more additional padding octets at the end which are not part of the payload. The last octet of the padding contains a count of how many padding octets should be ignored. Padding may be needed by some encryption algorithms with fixed block sizes or for carrying several RTP packets in a lower-layer protocol data unit.

extension (X): 1 bit

If the extension bit is set, the fixed header is followed by exactly one header extension.

CSRC count (CC): 4 bits

The CSRC count contains the number of CSRC identifiers that follow the fixed header.

marker (M): 1 bit

The interpretation of the marker is defined by a profile. It is intended to allow significant events such as frame boundaries to be marked in the packet stream. A profile may define additional marker bits or specify that there is no marker bit by changing the number of bits in the payload type field.

payload type (PT): 7 bits

This field identifies the format of the RTP payload and determines its interpretation by the application. A profile specifies a default static mapping of payload type codes to payload formats. Additional payload type codes may be defined dynamically through non-RTP means. An initial set of default mappings for audio and video is specified in the companion profile Internet-Draft draft-ietf-avt-profile [183], [186], and may be extended in future editions of the Assigned Numbers RFC [171]. An RTP sender emits a single RTP payload type at any given time; this field is not intended for multiplexing separate media streams.

sequence number: 16 bits

The sequence number increments by one for each RTP data packet sent, and may be used by the receiver to detect packet loss and to restore packet sequence. The initial value of the sequence number is random (unpredictable) to make known-plaintext attacks on encryption more difficult, even if the source itself does not encrypt, because the packets may flow through a translator that does.

timestamp: 32 bits

The timestamp reflects the sampling instant of the first octet in the RTP data packet. The sampling instant must be derived from a clock that increments monotonically and linearly in time to allow synchronization and jitter calculations. The resolution of the clock must be sufficient for the desired synchronization accuracy and for measuring packet arrival jitter (one tick per video frame is typically not sufficient). The clock frequency is dependent on the format of data carried as payload and is specified statically in the profile or payload format specification that defines the format, or may be specified dynamically for payload formats defined through non-RTP means. If RTP packets are generated periodically, the nominal sampling instant as determined from the sampling clock is to be used, not a reading of the system clock. As an example, for fixed-rate audio the timestamp clock would likely increment by one for each sampling period. If an audio application reads blocks covering 160 sampling periods from the input device, the timestamp would be increased by 160 for each such block, regardless of whether the block is transmitted in a packet or dropped as silent. The initial value of the timestamp is random, as for the sequence number. Several consecutive RTP packets may have equal timestamps if they are (logically) generated at once, e.g., belong to the same video frame. Consecutive RTP packets may contain timestamps that are not monotonic if the data is not transmitted in the order it was sampled, as in the case of MPEG interpolated video frames. (The sequence numbers of the packets as transmitted will still be monotonic.)

SSRC: 32 bits

The SSRC field identifies the synchronization source. This identifier is chosen randomly, with the intent that no two synchronization sources within the same RTP session will have the same SSRC identifier. Although the probability of multiple sources choosing the same identifier is low, all RTP implementations must be prepared to detect and resolve collisions. Section 8 describes the probability of collision along with a mechanism for resolving collisions and detecting RTP-level forwarding loops based on the uniqueness of the SSRC identifier. If a source changes its source transport address, it must also choose a new SSRC identifier to avoid being interpreted as a looped source.

CSRC list: 0 to 15 items, 32 bits each

The CSRC list identifies the contributing sources for the payload contained in this packet. The number of identifiers is given by the CC field. If there are more than 15 contributing sources, only 15 may be identified. CSRC identifiers are inserted by mixers, using the SSRC identifiers of contributing sources. For example, for audio packets the SSRC identifiers of all sources that were mixed together to create a packet are listed, allowing correct talker indication at the receiver.

A.3 RTCP Packet Format

The RFC defines four RTCP packets:

- SR: Sender report, for transmission and reception statistics from participants that are active senders.
- RR: Receiver report, for reception statistics from participants that are not active senders.
- SDES: Source description items, including CNAME.
- BYE: Indicates end of participation.

The first two, SR and RR, can be used for quality measurements and are described in figure A.2 and figure A.3.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|    RC   |   PT=SR=200   |             length            | header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         SSRC of sender                        |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|              NTP timestamp, most significant word             | sender
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ info
|             NTP timestamp, least significant word             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         RTP timestamp                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     sender's packet count                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      sender's octet count                     |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                 SSRC_1 (SSRC of first source)                 | report
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ block
| fraction lost |       cumulative number of packets lost       |   1
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           extended highest sequence number received           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      interarrival jitter                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         last SR (LSR)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   delay since last SR (DLSR)                  |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                 SSRC_2 (SSRC of second source)                | report
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ block
:                               ...                             :   2
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                  profile-specific extensions                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure A.2 - RTCP Sender Report (SR) Header

The sender report packet consists of three sections, possibly followed by a fourth profile-specific extension section if defined. The first section, the header, is 8 octets long. The fields have the following meaning:

version (V): 2 bits

> Identifies the version of RTP, which is the same in RTCP packets as in RTP data packets. The version defined by this specification is two (2).

padding (P): 1 bit

> If the padding bit is set, this RTCP packet contains some additional padding octets at the end which are not part of the control information. The last octet of the padding is a count of how many padding octets should be ignored. Padding may be needed by some encryption algorithms with fixed block sizes. In a compound RTCP packet, padding should only be required on the last individual packet because the compound packet is encrypted as a whole.

reception report count (RC): 5 bits

> The number of reception report blocks contained in this packet. A value of zero is valid.

packet type (PT): 8 bits

Contains the constant 200 to identify this as an RTCP SR packet.

length: 16 bits

The length of this RTCP packet in 32-bit words minus one, including the header and any padding. (The offset of one makes zero a valid length and avoids a possible infinite loop in scanning a compound RTCP packet, while counting 32-bit words avoids a validity check for a multiple of 4.)

SSRC: 32 bits

The synchronization source identifier for the originator of this SR packet.

The second section, the sender information, is 20 octets long and is present in every sender report packet. It summarizes the data transmissions from this sender. The fields have the following meaning:

NTP timestamp: 64 bits

Indicates the wallclock time when this report was sent so that it may be used in combination with timestamps returned in reception reports from other receivers to measure round-trip propagation to those receivers. Receivers should expect that the measurement accuracy of the timestamp may be limited to far less than the resolution of the NTP timestamp. The measurement uncertainty of the timestamp is not indicated as it may not be known. A sender that can keep track of elapsed time but has no notion of wallclock time may use the elapsed time since joining the session instead. This is assumed to be less than 68 years, so the high bit will be zero. It is permissible to use the sampling clock to estimate elapsed wallclock time. A sender that has no notion of wallclock or elapsed time may set the NTP timestamp to zero.

RTP timestamp: 32 bits

Corresponds to the same time as the NTP timestamp (above), but in the same units and with the same random offset as the RTP timestamps in data packets. This correspondence may be used for intra- and inter-media synchronization for sources whose NTP timestamps are synchronized, and may be used by media- independent receivers to estimate the nominal RTP clock frequency. Note that in most cases this timestamp will not be equal to the RTP timestamp in any adjacent data packet. Rather, it is calculated from the corresponding NTP timestamp using the relationship between the RTP timestamp counter and real time as maintained by periodically checking the wallclock time at a sampling instant.

sender's packet count: 32 bits

The total number of RTP data packets transmitted by the sender since starting transmission up until the time this SR packet was generated. The count is reset if the sender changes its SSRC identifier.

sender's octet count: 32 bits

> The total number of payload octets (i.e., not including header or padding) transmitted in RTP data packets by the sender since starting transmission up until the time this SR packet was generated. The count is reset if the sender changes its SSRC identifier. This field can be used to estimate the average payload data rate.

The third section contains 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 packets from a single synchronization source. Receivers do not carry over statistics when a source changes its SSRC identifier due to a collision. These statistics are:

SSRC_n (source identifier): 32 bits

> The SSRC identifier of the source to which the information in this reception report block pertains.

fraction lost: 8 bits

> The fraction of RTP data packets from source SSRC_n lost since the previous SR or RR packet was sent, expressed as a fixed point number with the binary point at the left edge of the field. (That is equivalent to taking the integer part after multiplying the loss fraction by 256.) This fraction is defined to be the number of packets lost divided by the number of packets expected, as defined in the next paragraph. If the loss is negative due to duplicates, the fraction lost is set to zero. Note that a receiver cannot tell whether any packets were lost after the last one received, and that there will be no reception report block issued for a source if all packets from that source sent during the last reporting interval have been lost.

cumulative number of packets lost: 24 bits

> The total number of RTP data packets from source SSRC_n that have been lost since the beginning of reception. This number is defined to be the number of packets expected less the number of packets actually received, where the number of packets received includes any which are late or duplicates. Thus packets that arrive late are not counted as lost, and the loss may be negative if there are duplicates. The number of packets expected is defined to be the extended last sequence number received, as defined next, less the initial sequence number received.

extended highest sequence number received: 32 bits

> The low 16 bits contain the highest sequence number received in an RTP data packet from source SSRC_n, and the most significant 16 bits extend that sequence number with the corresponding count of sequence number cycles. Note that different receivers within the same session will generate different extensions to the sequence number if their start times differ significantly.

interarrival jitter: 32 bits

An estimate of the statistical variance of the RTP data packet interarrival time, measured in timestamp units and expressed as an unsigned integer. The interarrival jitter J is defined to be the mean deviation (smoothed absolute value) of the difference D in packet spacing at the receiver compared to the sender for a pair of packets. As shown in the equation below, this is equivalent to the difference in the "relative transit time" for the two packets; the relative transit time is the difference between a packet's RTP timestamp and the receiver's clock at the time of arrival, measured in the same units. If Si is the RTP timestamp from packet i, and Ri is the time of arrival in RTP timestamp units for packet i, then for two packets i and j, D may be expressed as

$$D(i,j)=(Rj-Ri)-(Sj-Si)=(Rj-Sj)-(Ri-Si)$$

The interarrival jitter is calculated continuously as each data packet i is received from source SSRC_n, using this difference D for that packet and the previous packet i-1 in order of arrival (not necessarily in sequence), according to the formula

$$J=J+(|D(i-1,i)|-J)/16$$

Whenever a reception report is issued, the current value of J is sampled. The jitter calculation is prescribed here to allow profile-independent monitors to make valid interpretations of reports coming from different implementations. This algorithm is the optimal first-order estimator and the gain parameter 1/16 gives a good noise reduction ratio while maintaining a reasonable rate of convergence.

last SR timestamp (LSR): 32 bits

The middle 32 bits out of 64 in the NTP timestamp received as part of the most recent RTCP sender report (SR) packet from source SSRC_n. If no SR has been received yet, the field is set to zero.

delay since last SR (DLSR): 32 bits

The delay, expressed in units of 1/65536 seconds, between receiving the last SR packet from source SSRC_n and sending this reception report block. If no SR packet has been received yet from SSRC_n, the DLSR field is set to zero. Let SSRC_r denote the receiver issuing this receiver report. Source SSRC_n can compute the round propagation delay to SSRC_r by recording the time A when this reception report block is received. It calculates the total round-trip time A-LSR using the last SR timestamp (LSR) field, and then subtracting this field to leave the round-trip propagation delay as (A- LSR - DLSR). This may be used as an approximate measure of distance to cluster receivers, although some links have very asymmetric delays.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|    RC   |   PT=RR=201   |             length            | header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     SSRC of packet sender                     |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                 SSRC_1 (SSRC of first source)                 | report
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ block
| fraction lost |       cumulative number of packets lost       |   1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           extended highest sequence number received           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      interarrival jitter                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         last SR (LSR)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   delay since last SR (DLSR)                  |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                 SSRC_2 (SSRC of second source)                | report
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ block
:                               ...                             :   2
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                  profile-specific extensions                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure A.3 - RTCP Receiver Report (RR) Header

The format of the receiver report (RR) packet is the same as that of the SR packet except that the packet type field contains the constant 201 and the five words of sender information are omitted (these are the NTP and RTP timestamps and sender's packet and octet counts). The remaining fields have the same meaning as for the SR packet. An empty RR packet (RC = 0) is put at the head of a compound RTCP packet when there is no data transmission or reception to report.

B MRT (Multicast Routing Tool)

In the following, a short overview is provided to the implementation of the multicast routing tool. First, a short summary of the developing process of MRT is given. Secondly, the usage of the program in order to model multicast networks as well as services is described. Finally, the possibility to include dynamically gathered measurement results is summarized.

B.1 Developing the MRT

MRT was developed using the JDK (Java development kit) version 1.2.2 of Sun Microsystems, Inc. The software has been tested using newer SDK versions as well. The AWT (abstract windowing toolkit) has been chosen instead of newer packets of the JFC (Java Foundation Classes) like SWING due to its stability. Additionally, AWT applications are running on most hardware and software platforms.

The implementation of the MRT has been done straight forward out of the UML specified model. All objects are defined in separate files. Each one is inheriting properties and capabilities from a more common object. For example, the object ETHERNET inherits functions from the object LINK, which inherits capabilities from the object NETWORK. An instance of such an object defines an existing device. The father of all objects is a project.

The MRT includes mechanisms to save the modeled multicast infrastructure to a file in order reload it at a later time. Currently, this dump is done in a binary form. A succeeding version should use XML (extended markup language) to allow the definition of the multicast network using third party tools, or just a text editor.

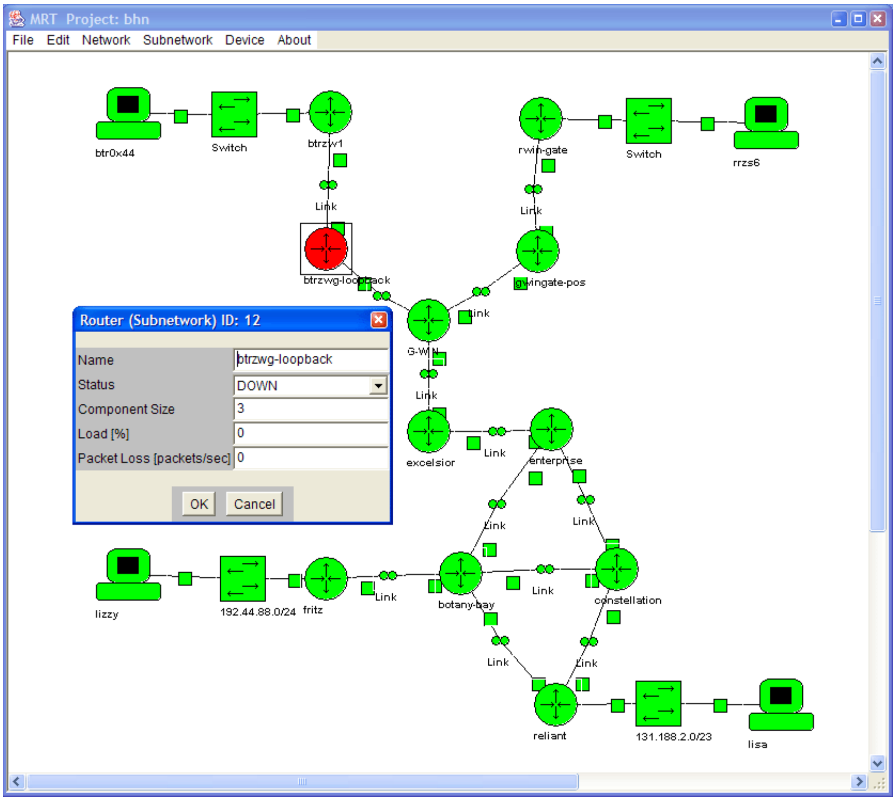
B.2 Usage of the MRT

A project is started with modeling the network infrastructure. All the hosts, routers, and networks have to be defined. Three steps are necessary to create a new object:

- the new object is created using the appropriate menu
- it can be placed freely on the desktop using the drag-and-drop principle
- the properties of the object have to be configured using a context menu

An example of the creation of a new router is shown in figure B.1. The properties dialog allows to configure the name and the IP address of the router as well as to initially set some quality of service parameters such as the state (UP / DOWN), the load, and the packet loss ratio.

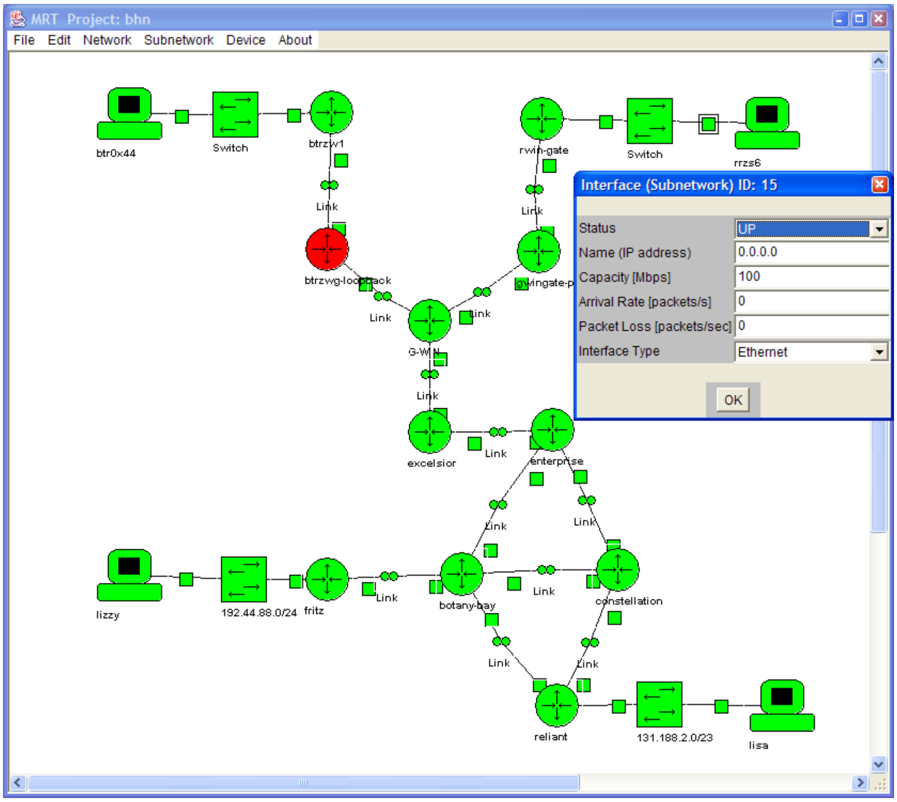
Another example, the definition of an interface, is shown in figure B.2. Interfaces are created using the context menu of any node. The connection to a link is done by dragging the interface on the link object.



Router (Subnetwork) ID: 12

Name	btrzwg-loopback
Status	DOWN
Component Size	3
Load [%]	0
Packet Loss [packets/sec]	0

Figure B.1 - Configuration of the Properties of a Router

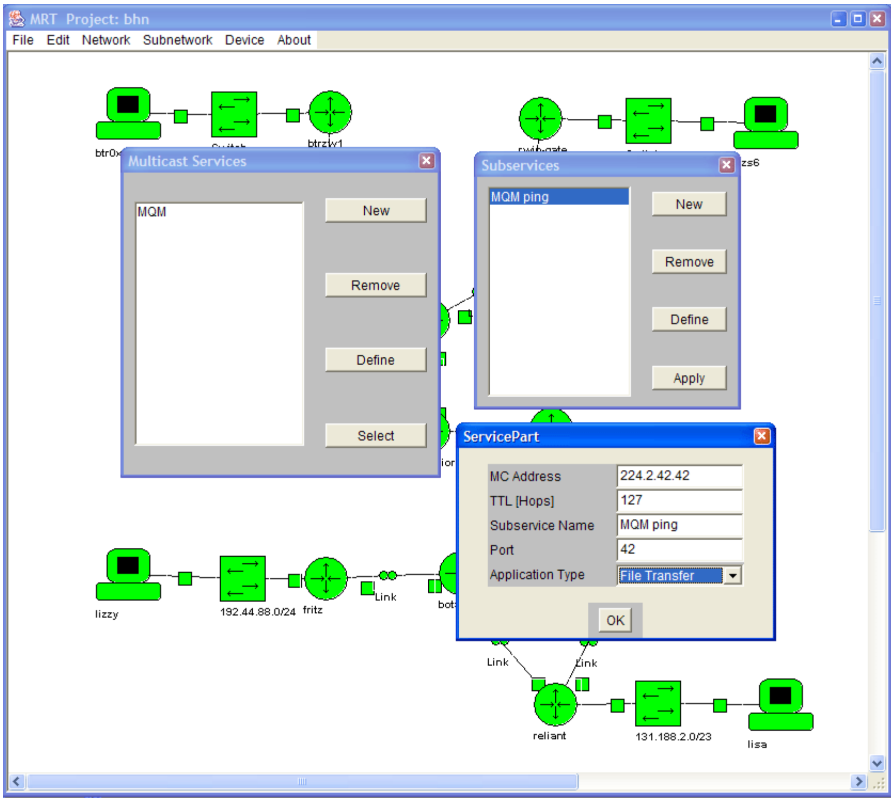


Interface (Subnetwork) ID: 15

Status	UP
Name (IP address)	0.0.0.0
Capacity [Mbps]	100
Arrival Rate [packets/s]	0
Packet Loss [packets/sec]	0
Interface Type	Ethernet

Figure B.2 - Configuration of the Properties of an Interface

After modeling the physical infrastructure of the IP multicast network, the services have to be defined. Figure B.3 shows an example of creating a service. The figure also demonstrates the configuration of the service parts and their properties such as the multicast group address and the application type.

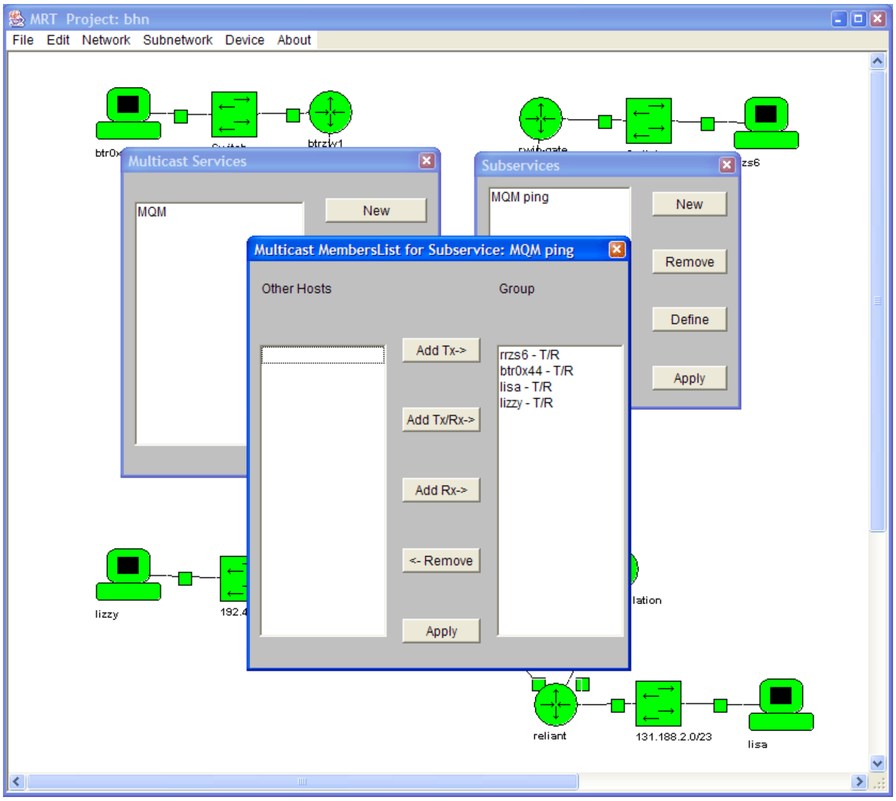


MC Address: 224.2.42.42
TTL [Hops]: 127
Subservice Name: MQM ping
Port: 42
Application Type: File Transfer

Figure B.3 - Definition of the Multicast Services

The association of the end systems to the multicast services respectively to the service parts is shown in figure B.4. A list of available hosts is provided in order to select single end systems as a participant of the particular service part. Each host can be a sender, a receiver, or both.

Based on the configured multicast service, the calculation of an optimum path through the network using the routing algorithm can be initiated. Figure B.5 shows the dialog window to select the service for which the multicast tree has to be analyzed. Additionally, it is possible to define a number of properties for the routing algorithm. For example, the delay weight specifies the importance of the delay for the following calculation process. Other weighting factors quantify the error ratio and the number of hops.

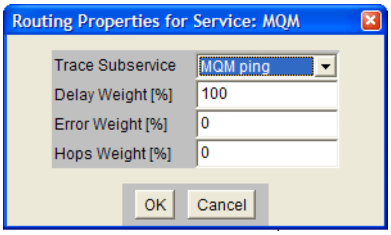


MQM

MQM ping

rrzs6 - T/R
btr0x44 - T/R
lisa - T/R
lizzy - T/R

Figure B.4 - Association of End Systems to a Service Part

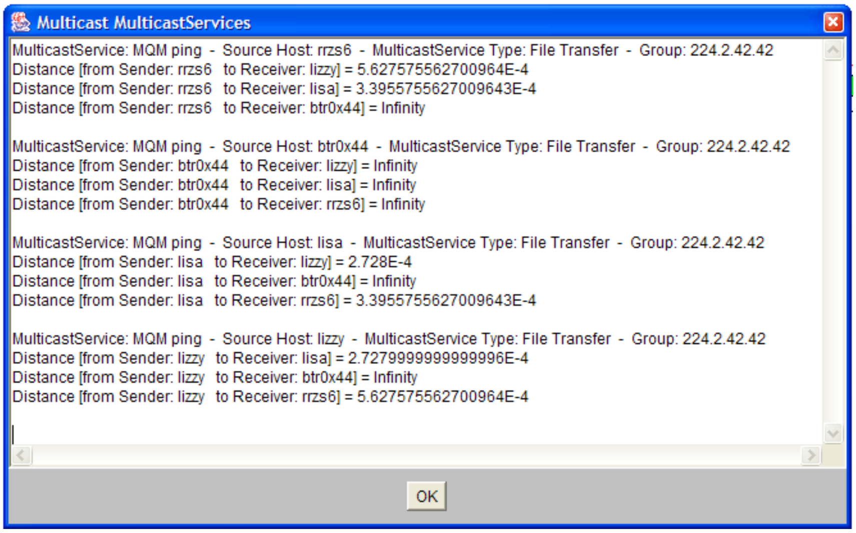


Trace Subservice	MQM ping
Delay Weight [%]	100
Error Weight [%]	0
Hops Weight [%]	0

Figure B.5 - Definition of the Routing Properties

The results of the routing algorithm are shown in figure B.6. The calculated distance from each sender towards to all receivers is numerically displayed. If no path has been found between two hosts, the distance is set to infinity.

In the shown example, the host btr0x44 is unreachable. Therefore, the distance from and to this host is infinity. All others are represented by a discrete number, which is describing the quality of the connection as well. The smaller the number is, the better is the quality of the particular path.



Multicast MulticastServices

```
MulticastService: MQM ping  -  Source Host: rrzs6  -  MulticastService Type: File Transfer  -  Group: 224.2.42.42
Distance [from Sender: rrzs6   to Receiver: lizzy] = 5.627575562700964E-4
Distance [from Sender: rrzs6   to Receiver: lisa] = 3.3955755627009643E-4
Distance [from Sender: rrzs6   to Receiver: btr0x44] = Infinity

MulticastService: MQM ping  -  Source Host: btr0x44  -  MulticastService Type: File Transfer  -  Group: 224.2.42.42
Distance [from Sender: btr0x44   to Receiver: lizzy] = Infinity
Distance [from Sender: btr0x44   to Receiver: lisa] = Infinity
Distance [from Sender: btr0x44   to Receiver: rrzs6] = Infinity

MulticastService: MQM ping  -  Source Host: lisa  -  MulticastService Type: File Transfer  -  Group: 224.2.42.42
Distance [from Sender: lisa   to Receiver: lizzy] = 2.728E-4
Distance [from Sender: lisa   to Receiver: btr0x44] = Infinity
Distance [from Sender: lisa   to Receiver: rrzs6] = 3.3955755627009643E-4

MulticastService: MQM ping  -  Source Host: lizzy  -  MulticastService Type: File Transfer  -  Group: 224.2.42.42
Distance [from Sender: lizzy   to Receiver: lisa] = 2.7279999999999996E-4
Distance [from Sender: lizzy   to Receiver: btr0x44] = Infinity
Distance [from Sender: lizzy   to Receiver: rrzs6] = 5.627575562700964E-4
```

Figure B.6 - Numerical Results of the Routing Algorithm

Finally, a graphical representation of the calculated path is displayed. The calculated path of the previously discussed example is shown in figure B.7.

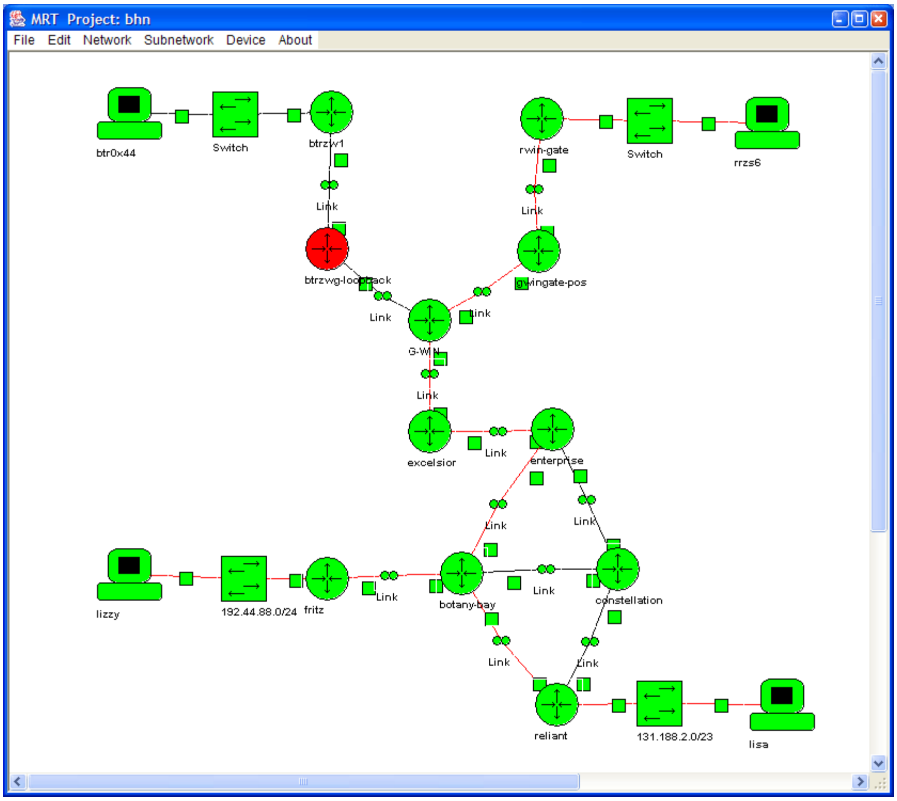


Figure B.7 - Graphical Representation of the calculated Path

So far, the MRT can be used to display multicast networks and to estimate an optimum path for a given multicast service based on statically modeled information.

B.3 Import of Dynamic Data

Measurement results can be imported in order to allow more actual calculations. Such results may be gathered from distributed measurement probes in the network as well as from the devices themselves. Currently, a simple text file oriented import function has been implemented.

The format of the import is very simple. Each row consists of six fields. The first one specifies the object type, the second field contains a name. All others define various states of the object. The following lines are the rows of such a file:

```
(1) Router  btrzwg-loopback DOWN    * * * *

(2) Router  botany-bay      UP      0 90 * *

(3) Host    lizzy           DOWN    * * * *
```

The meaning of the example lines is the following (an asterisk stands for an unchanged value):

(1) The router btrzwg-loopback is in the state DOWN.

(2) The router botany-bay is in the state UP. It has a zero packet loss ratio, but a utilization of 90%.

(3) The host lizzy is in the state DOWN.

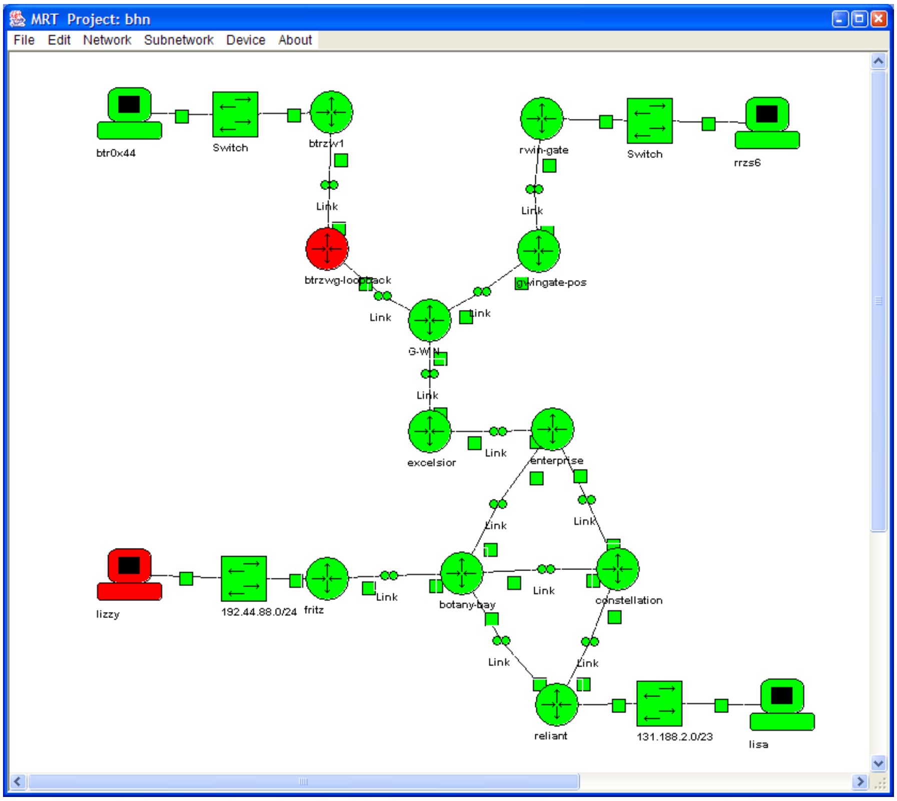
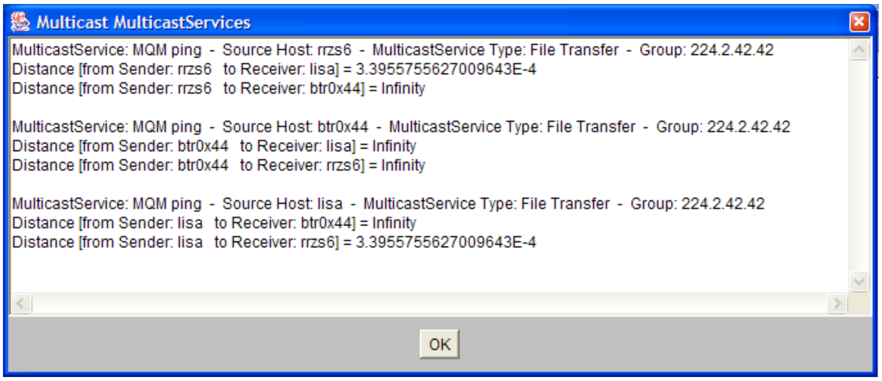


Figure B.8 - Display of the failing Devices

After importing the shown file, the MRT displays the new state of the network by changing the color of all failing devices to red. Figure B.8 shows a screenshot of this situation.

Using the routing algorithm again, new paths through the network can be calculated based on the changed conditions. The results are shown in figure B.9 and figure B.10 for the numerical and the graphical representation of the estimated quality between the communicating hosts and the used paths respectively.



```
MulticastService: MQM ping  -  Source Host: rrzs6  -  MulticastService Type: File Transfer  -  Group: 224.2.42.42
Distance [from Sender: rrzs6   to Receiver: lisa] = 3.3955755627009643E-4
Distance [from Sender: rrzs6   to Receiver: btr0x44] = Infinity

MulticastService: MQM ping  -  Source Host: btr0x44  -  MulticastService Type: File Transfer  -  Group: 224.2.42.42
Distance [from Sender: btr0x44   to Receiver: lisa] = Infinity
Distance [from Sender: btr0x44   to Receiver: rrzs6] = Infinity

MulticastService: MQM ping  -  Source Host: lisa  -  MulticastService Type: File Transfer  -  Group: 224.2.42.42
Distance [from Sender: lisa   to Receiver: btr0x44] = Infinity
Distance [from Sender: lisa   to Receiver: rrzs6] = 3.3955755627009643E-4
```

Figure B.9 - Numerical Results of the Routing Algorithm

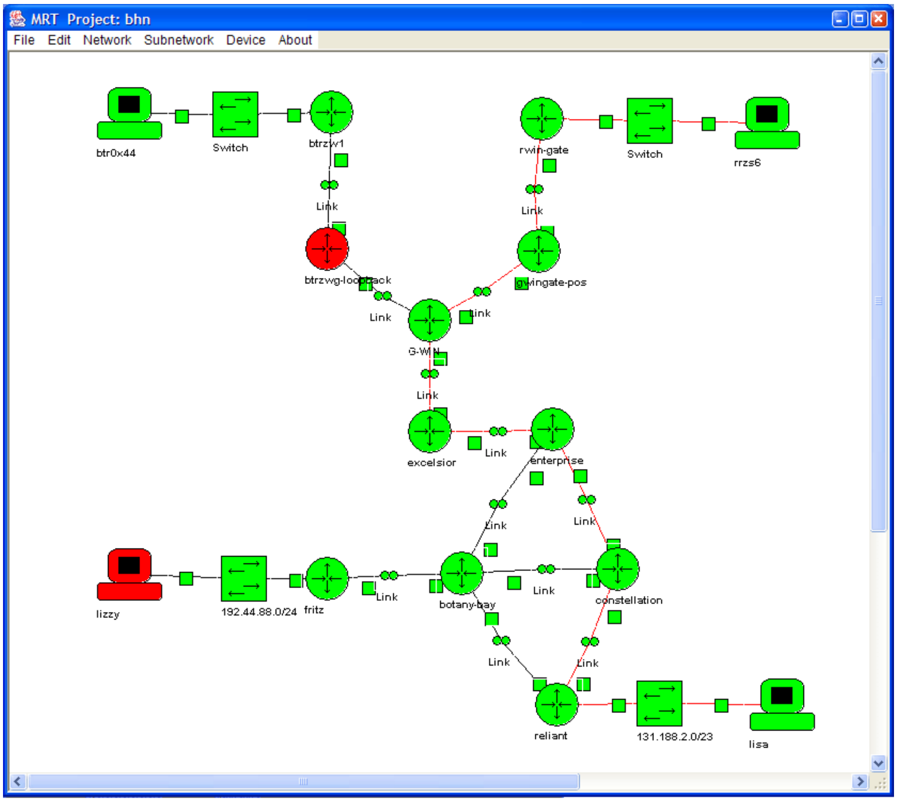


Figure B.10 - Graphical Representation of the calculated Path

C DNDF (Dummynet with Distribution Function)

In order to use DNDF to manipulate the behavior of a particular network, it is necessary to introduce the configuration of the underlying dummynet driver as well as that of DNDF itself. Additionally, the test environment to evaluate the IP impairment tool and the measurement results are discussed.

C.1 Configuring Dummynet

Dummynet strongly depends on the firewall code of FreeBSD. It uses its mechanisms to classify packets of individual streams. The packets are separated into different queues for which dummynet adds some new functionality in order to introduce a static delay or a configurable packet loss priority.

The following steps are required to configure dummynet:

(1) creating a queue

```
ipfw add pipe 1 ip from 192.44.88.0/24 to 131.188.3.150
```

A new queue (called pipe in the firewall code) is created and the traffic from any host in the network 192.44.88.0/24 to the host with the IP address 131.188.3.150 is assigned to it.

(2) applying dummynet features

```
ipfw pipe 1 config delay 300ms
```

An absolute delay of 300 ms is added to each packet flowing through in pipe 1.

```
ipfw pipe 1 config plr 0.3
```

A packet loss ratio of 0.3 is assigned to pipe 1. Therefore, every third packet on pipe 1 will be dropped.

C.2 Configuring DNDF

The mechanism to apply a variable delay to a packet stream works as follows: A kernel-internal table defines single delay values. The lookup of a specific value in this table is done using a random function. Each table entry has the same priority. An example of such a table including the resulting probability function is shown in the next subsection.

The configuration of the DNDF consists of several steps:

(1) A table specified in an ASCII file has to be created which specifies the values used to introduce a variable delay. This file can be generated using a text editor but also using special functions which create the table according to a predefined probability function such as the Gaussian function.

(2) The table has to be loaded into the kernel. A separate tool named cico exists for this task.

```
cico -l <filename>
```

(3) The DNDF function has to be turned on.

```
cico -a
```

C.3 Configuration used for the Examples

The setup to test the IP impairment tool is shown in figure C.1. The measurement equipment consists of a Smartbits 6000 traffic generator / traffic analyzer and the IP impairment tool running on a FreeBSD PC.

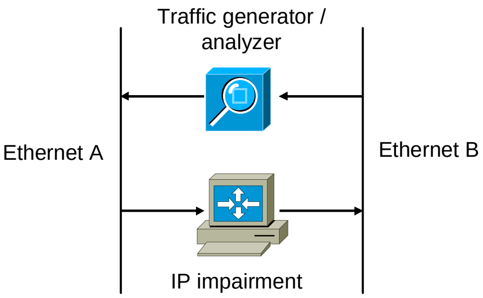


Figure C.1 - Lab Setup to Test the IP Impairment Tool

Flow #	Packet rate [pps]	Packet size [byte]	Bandwidth [kbps]	Application type
1	34	240	64	Voice over IP
2	68	300	160	MP3 Streaming
3	93	1.100	800	MPEG4 Streaming
4	596	1.100	5.000	MPEG2 Streaming
5	3055	429	10.000	best effort

Table C.1 - Definition of the configured Packet Flows

The Smartbits 6000 has been configured to produce five different data streams representing typical applications in the internet. The definitions of the flows are shown in table C.1.

The configuration of the IP impairment tool used for the examples appeared as follows: We created two queues, one for the best effort background traffic and one for the data stream to be manipulated. The association of the packets to each queue was performed using the destination IP address:

```
ipfw add pipe 1 ip from any to 129.192.5.5
```

Constant delays and a particular packet loss ratio were configured as shown previously.

The tables for the DNDF were created using a simple program which allows the construction of such a table using the Gaussian probability function. The minimum, the maximum, the average, and the variance can be specified using this tool. For the tests we used two different settings, whose functions are shown in figure C.2:

(1) Gaussian distribution, minimum 0 ms, maximum 500 ms, average 150 ms, variance 50

(2) Gaussian distribution, minimum 0 ms, maximum 500 ms, average 150 ms, variance 75

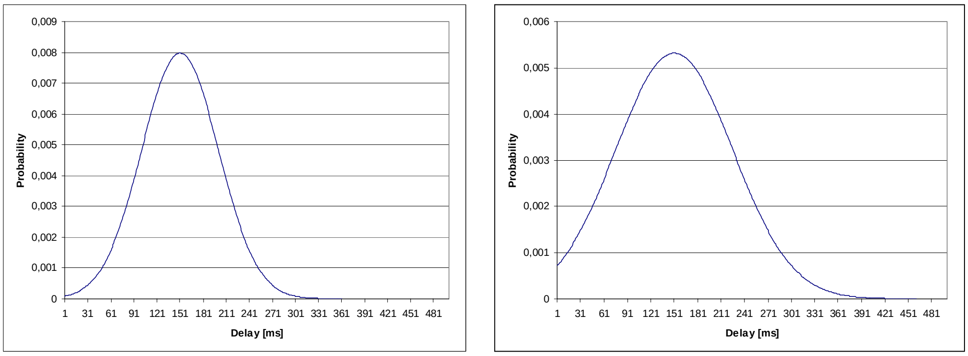


Figure C.2 - Distribution Functions for Jitter Tests (left: variance 50; right: variance 75)

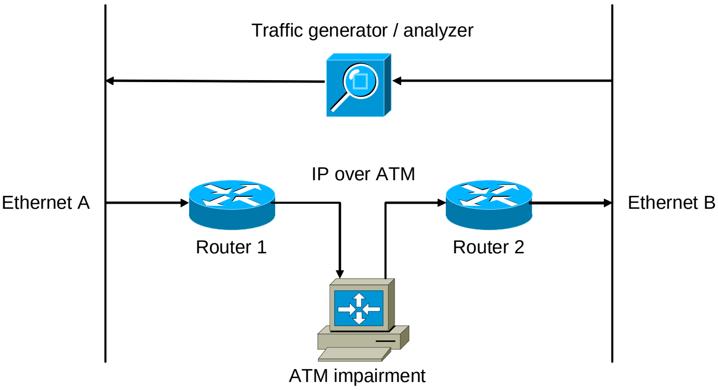


Figure C.3 - Lab Setup to for the Reference Measurements

In order to get a well-defined reference for the measurements, we used a commercially available ATM based impairment. Such a reference measurement was required in order to verify the correct behavior of our own implementation, the DNDF. The test setup is shown in figure C.3.

In this test, the ATM impairment tool was a GN-Nettest interWATCH 95000. The ATM impairment was employed to create a single reference measurement by introducing an absolute delay of 200 ms.

C.4 Measurement Results

The following figures show the results of the measurements to test the IP impairment tool as well as the reference tests. The results of the delay tests are presented using histograms and the illustrations of packet loss measurements show the number of lost packets as a function of time.

Figure C.4 shows the reference measurement without any impairment tool. Depending on the packet size, the delay oscillates around 0.2 ms and 0.45 ms, which are typical values for transmissions in a local area network.

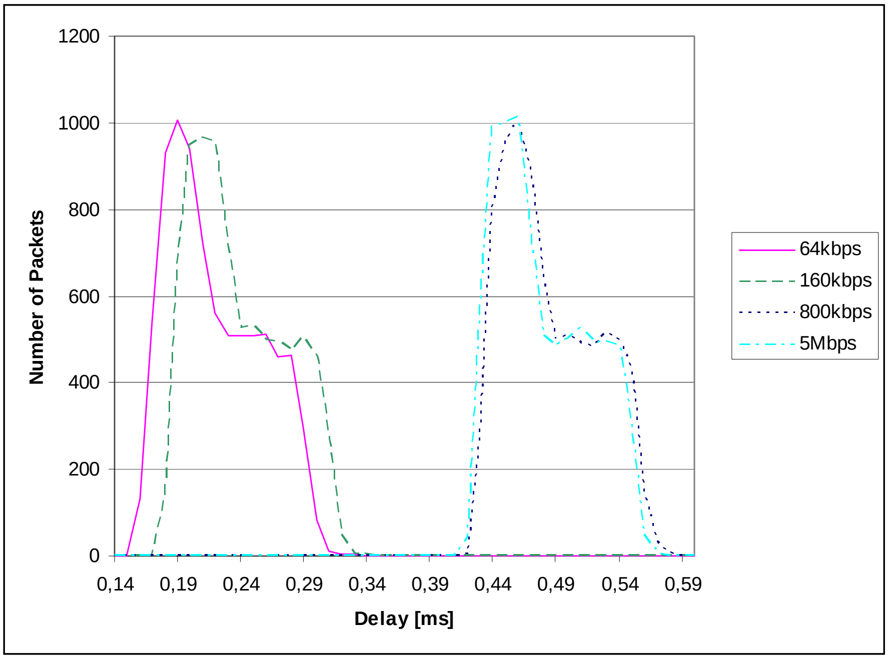


Figure C.4 - Reference Measurements without any Impairment Tool

The reference measurement using the ATM based impairment is shown in figure C.5. A constant delay of 0 ms has been configured. Due to the queuing effects in the involved routers, the resulting transmission delay is about 2.05 ms.

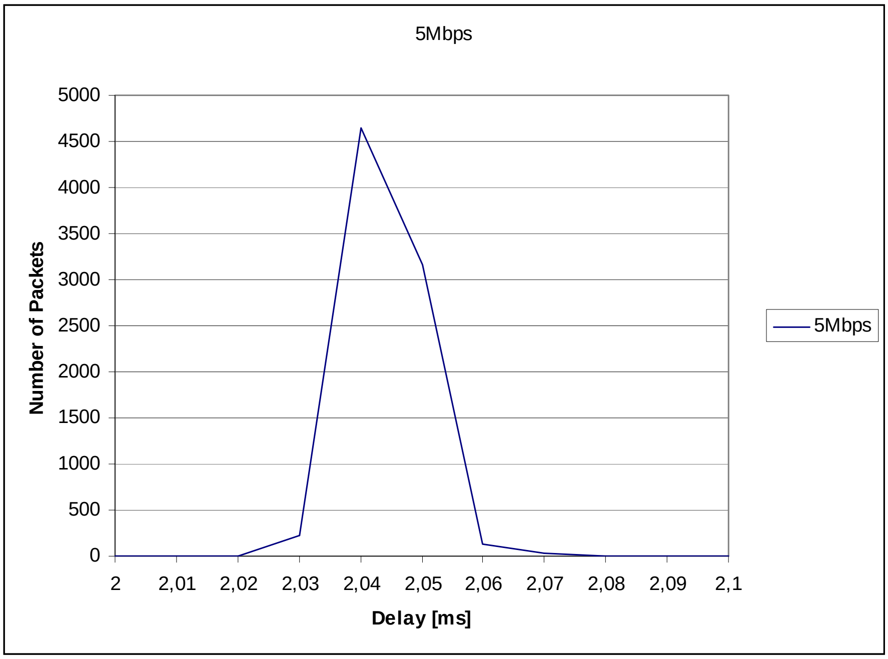


Figure C.5 - Reference Measurement using the ATM Impairment, Constant Delay of 0 ms

The last reference was taken by testing the IP impairment tool with a configured delay of 0 ms. The results, shown in figure C.6, show a resulting transmission delay of about 0.4 ms and 0.95 ms respectively depending on the packet size.

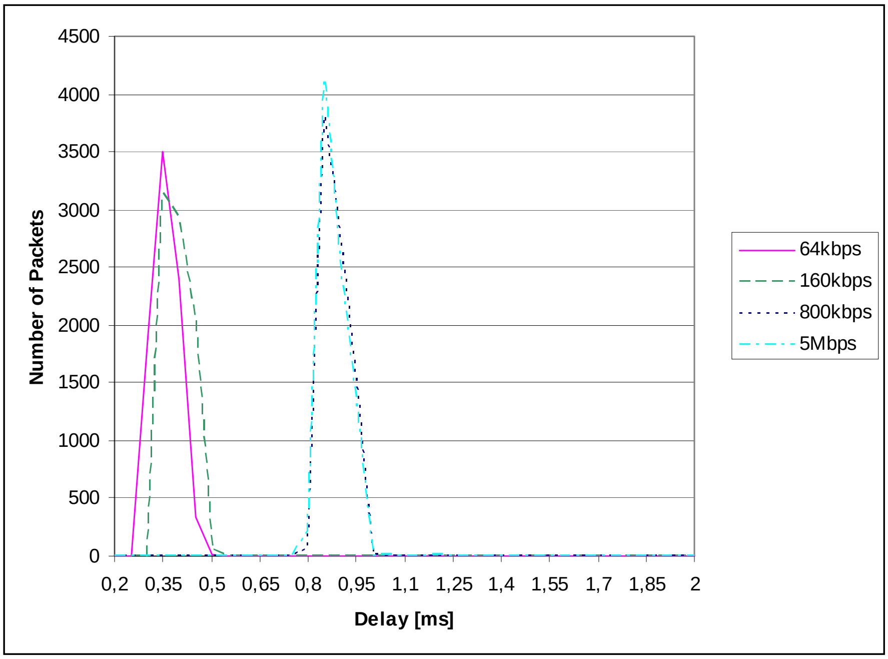


Figure C.6 - Constant Delay of 0 ms

To test the capabilities of dummynet to introduce a constant delay, two measurements were performed. First, the IP impairment tool was configured to a delay of 2 ms. Secondly, a latency of 200 ms was used.

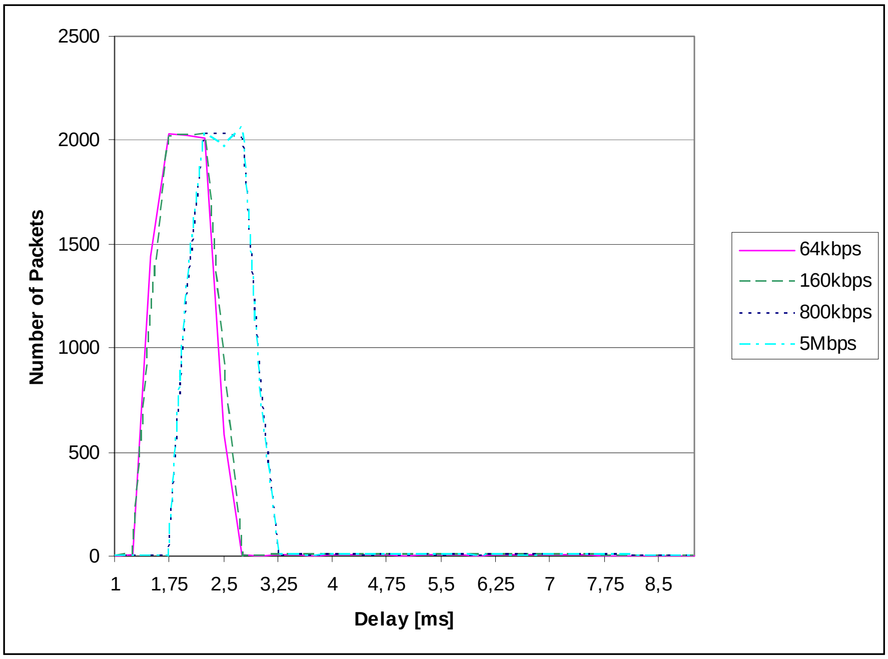


Figure C.7 - Constant Delay of 2 ms

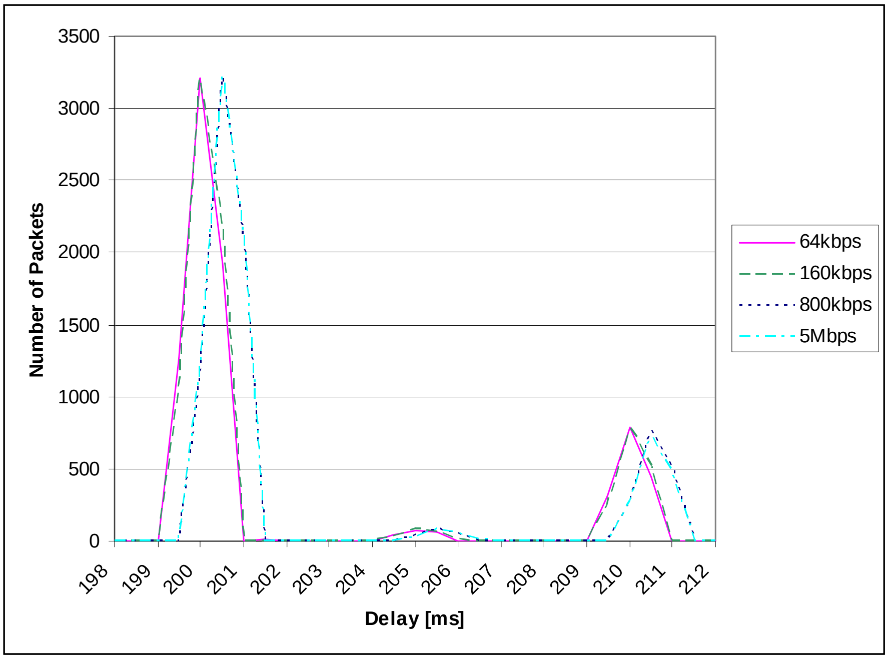


Figure C.8 - Constant Delay of 200 ms

The measured transmission delay ranges between 1.5 ms and 3 ms in the first case (shown in figure C.7) and between 199 ms and 202 ms in the second case (shown in figure C.8). Unfortunately, another peak appeared at about 210 ms in the second measurement. We decided to revoke this measurement. Typically, only the introduction of a variable delay and a specified packet loss ratio are required to test the applications behavior.

In order to evaluate the results of the last two measurements using the new IP based impairment tool, a reference measurement using the ATM impairment was carried out. A constant delay of 200 ms was configured for this test. The results are shown in figure C.9. The resulting latency oscillates very closely around 201.78 ms. The variance of the delay using the IP impairment tool is much broader than in the case of the ATM impairment. Nevertheless, the range is small enough for significant tests of multimedia applications.

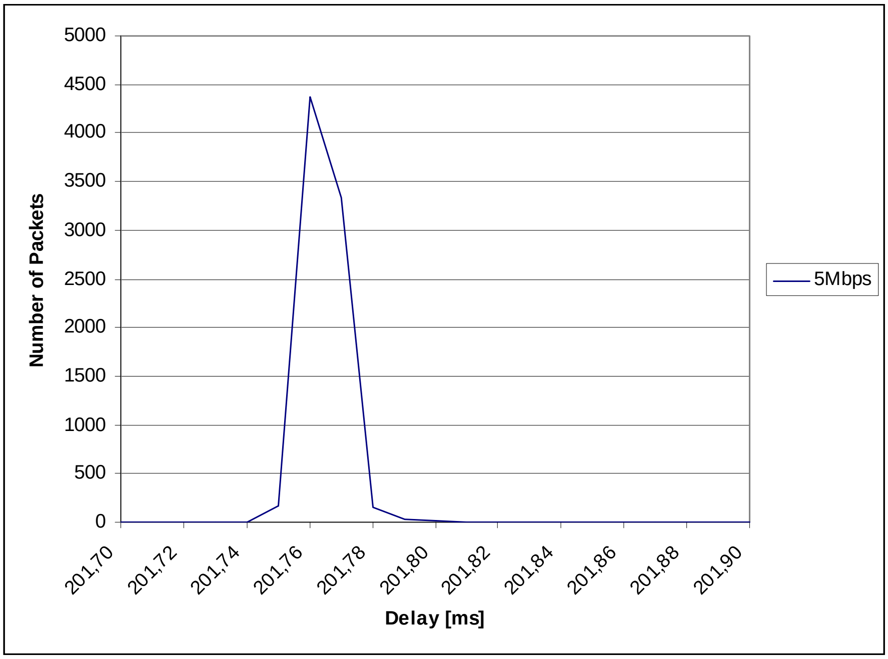


Figure C.9 - Constant Delay of 200 ms using the ATM Impairment

To follow the measurements of different constant delays, the capabilities of the IP impairment to introduce a specified packet loss ratio were examined. First, a packet loss ratio of 1% was applied. As expected, only single packets were dropped. The results are shown in figure C.10. Secondly, a loss ratio of 5% was tested. Thirdly, the same measurement was performed using a configured packet loss ratio of 10%. The results are shown in figure C.11 and figure C.12 for 5% and 10% respectively.

We found that the function to select packets in order to drop them, to achieve the configured packet loss ratio was not perfectly implemented. Dummynet tends to drop sequences of packets instead of single packets. The same effect is observed for large packet loss ratios. The impairment tool can be only used to test loss ratios greater than 1%.

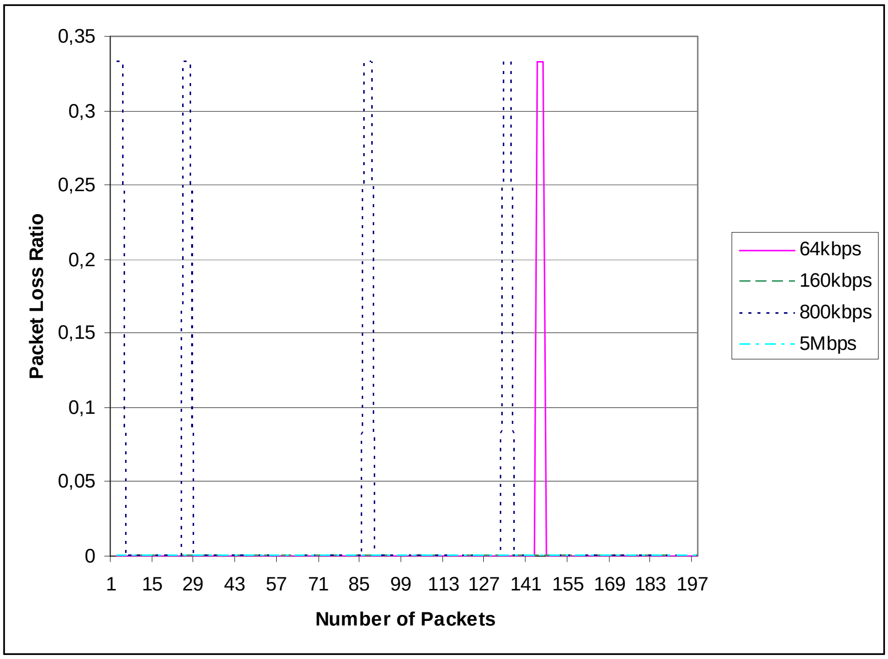


Figure C.10 - Packet Loss Ratio of 1%

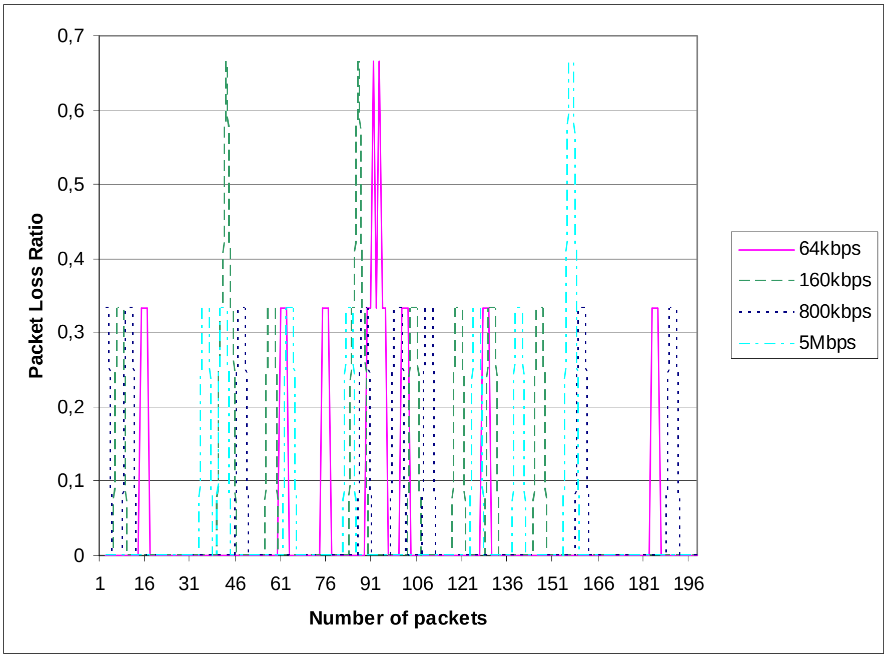


Figure C.11 - Packet Loss Ratio of 5%

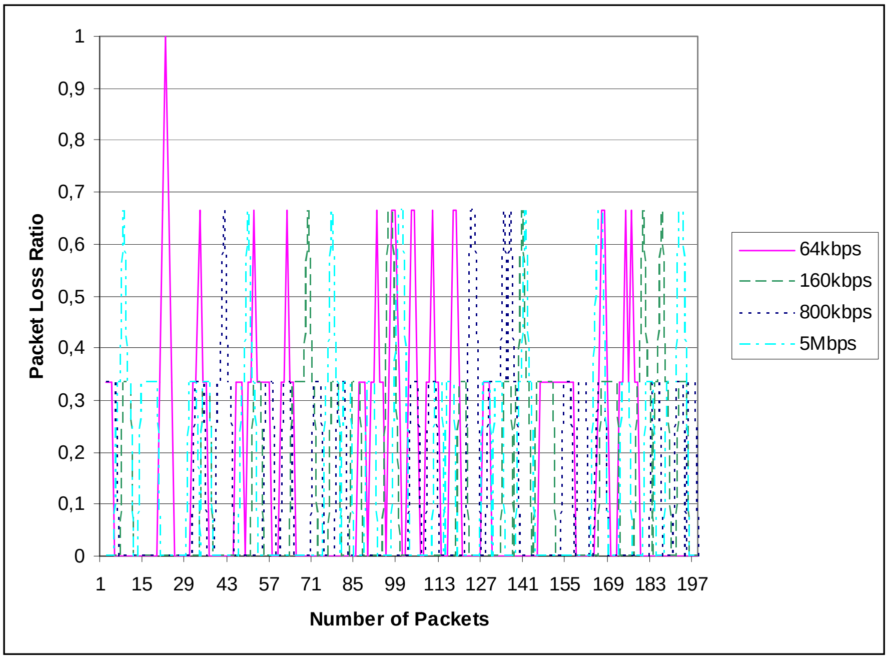


Figure C.12 - Packet Loss Ratio of 10%

Finally, the ability of the impairment tool to introduce a variable delay to a particular packet stream was tested. We used the probability distribution function described in the previous subsection. The results of both measurements are shown in figure C.13 and figure C.14 respectively.

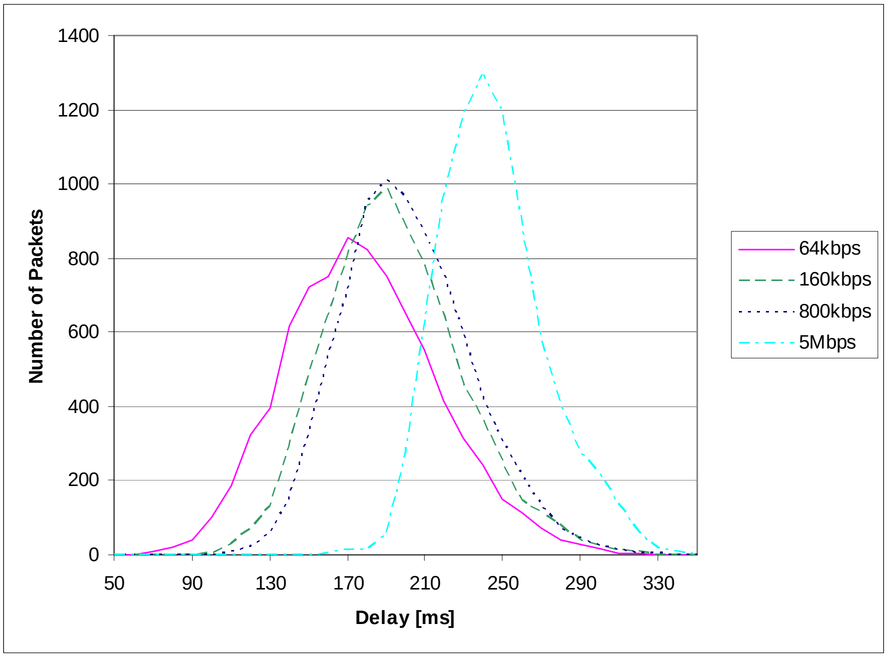


Figure C.13 - Variable Delay (min 0ms, max 500ms, avg 150ms, var 50)

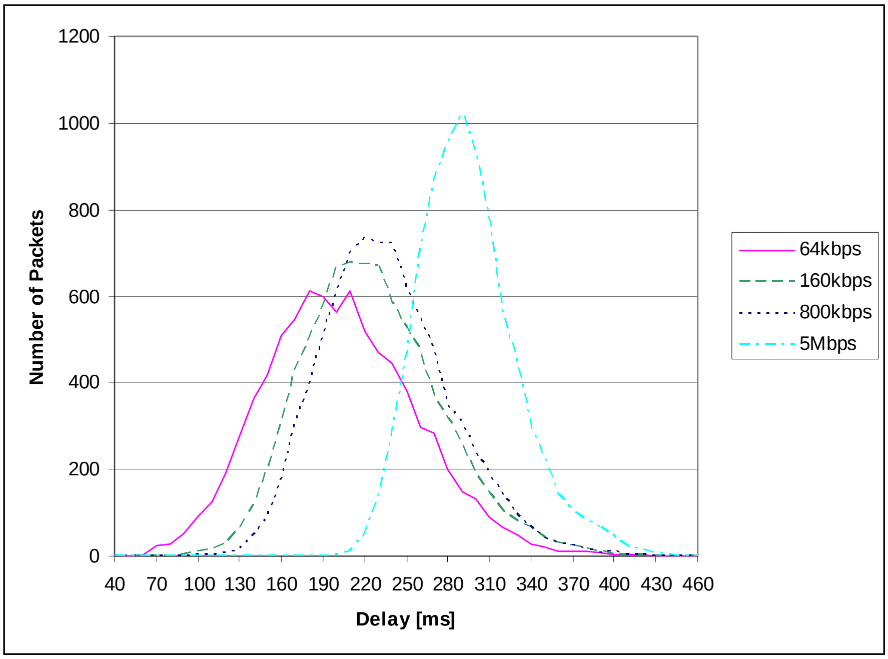


Figure C.14 - Variable Delay (min 0ms, max 500ms, avg 150ms, var 75)

In both cases, the results show a slight shift to higher delay values. A more dramatic shift can be determined for streams utilizing a higher bandwidth. In general, the variable delays introduced by the IP impairment tool are always comparable to the configured probability distribution functions. The shift can be adjusted by an adequate modification of the values in the kernel-table which describes the probability distribution function.

D MQM (Multicast Quality Monitor)

An overview of the implementation of the MQM is provided at this place. The prototypical implementation is rather simple. It consists of two separate programs:

(1) mqm_sender

The mqm_sender is used to initiate the MQM ping mechanism. It allows to send a single MQM ping request message to a configurable multicast group. The port number and the TTL can be specified as well.

The following actions are performed: a MQM packet is created, a sending timestamp is inserted and the packet is sent.

(2) mqm_receiver

The complexity of the mqm_receiver is a littly higher. This tool is intended to receive and to analyze MQM encoded packets as well as RTP and RTCP packets. At the start, the IP multicast address, the port number, and the TTL can be configured.

After the startup, the mqm_receiver starts listening on the specified multicast group for packets destined for the given port number as well as for port number + 1. The latter is used by RTP based streams for the control channel (RTCP). The performed actions depend on the type of the received packet:

MQM ping request: A MQM ping request has to be answered by a MQM ping response. Therefore, the mqm_receiver includes a reception timestamp into the MQM packet and sends it back to the multicast group. Additionally, a log message is produced including the calculated one-way delay from the originator to the local process.

MQM ping response: If a MQM ping response has been received, the MQM ping procedure is finished. The mqm_receiver creates an appropriate log message containing the IP addresses of the involved probes and the calculated delay values (OWD, RTT).

RTP packet / RTCP packet: The information in the RTP, respectively in the RTCP packets is decoded and corresponding log messages are created. For example, a log entry of a RTP packet consists of the IP address of the sender, the payload type, the sequence number, the timestamp, and the SSRC (compare appendix A).

A few examples of the usage of this prototypical implementation are provided in the following.

D.1 Analysis of MQM Ping Packets

The network configuration described in figure D.1 has been used for the examples shown in the following figures. The host lisa was sending MQM ping requests to the multicast group 224.42.42.42. All the shown probes, running on the hosts lisa, mc, and faui40p, have joined this group in order to catch the MQM messages and to react according to the message type.

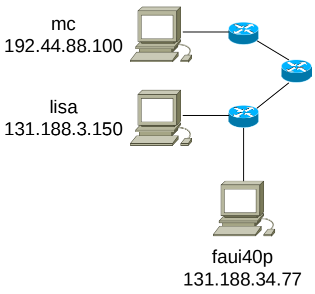
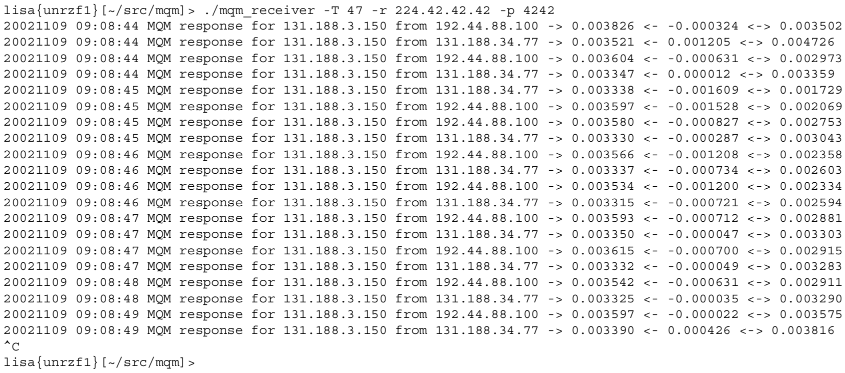


Figure D.1 - Network Configuration used for the MQM ping Examples

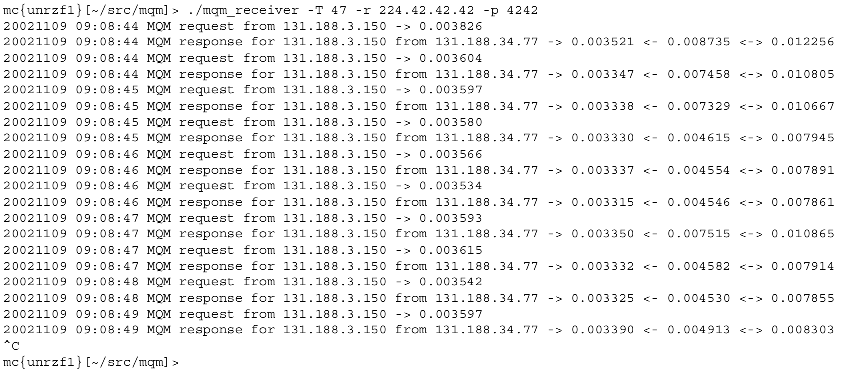
The host lisa was running the mqm_sender process, which has been invoked using the command: `mqm_sender -T 47 -s 224.42.42.42 -p 4242`. A single MQM ping request packet is sent to the given multicast group using the port number 4242 and a TTL of 47.

The numerical results are shown in figure D.2, figure D.3, and figure D.4 for the mqm_receivers running on lisa, mc, and faui40p respectively.



```
lisa{unrzf1}[~/src/mqm]> ./mqm_receiver -T 47 -r 224.42.42.42 -p 4242
20021109 09:08:44 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003826 <- -0.000324 <-> 0.003502
20021109 09:08:44 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003521 <- 0.001205 <-> 0.004726
20021109 09:08:44 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003604 <- -0.000631 <-> 0.002973
20021109 09:08:44 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003347 <- 0.000012 <-> 0.003359
20021109 09:08:45 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003338 <- -0.001609 <-> 0.001729
20021109 09:08:45 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003597 <- -0.001528 <-> 0.002069
20021109 09:08:45 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003580 <- -0.000827 <-> 0.002753
20021109 09:08:45 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003330 <- -0.000287 <-> 0.003043
20021109 09:08:46 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003566 <- -0.001208 <-> 0.002358
20021109 09:08:46 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003337 <- -0.000734 <-> 0.002603
20021109 09:08:46 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003534 <- -0.001200 <-> 0.002334
20021109 09:08:46 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003315 <- -0.000721 <-> 0.002594
20021109 09:08:47 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003593 <- -0.000712 <-> 0.002881
20021109 09:08:47 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003350 <- -0.000047 <-> 0.003303
20021109 09:08:47 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003615 <- -0.000700 <-> 0.002915
20021109 09:08:47 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003332 <- -0.000049 <-> 0.003283
20021109 09:08:48 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003542 <- -0.000631 <-> 0.002911
20021109 09:08:48 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003325 <- -0.000035 <-> 0.003290
20021109 09:08:49 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003597 <- -0.000022 <-> 0.003575
20021109 09:08:49 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003390 <- 0.000426 <-> 0.003816
^C
lisa{unrzf1}[~/src/mqm]>
```

Figure D.2 - MQM Packets received at Host lisa



```
mc{unrzf1}[~/src/mqm]> ./mqm_receiver -T 47 -r 224.42.42.42 -p 4242
20021109 09:08:44 MQM request from 131.188.3.150 -> 0.003826
20021109 09:08:44 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003521 <- 0.008735 <-> 0.012256
20021109 09:08:44 MQM request from 131.188.3.150 -> 0.003604
20021109 09:08:44 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003347 <- 0.007458 <-> 0.010805
20021109 09:08:45 MQM request from 131.188.3.150 -> 0.003597
20021109 09:08:45 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003338 <- 0.007329 <-> 0.010667
20021109 09:08:45 MQM request from 131.188.3.150 -> 0.003580
20021109 09:08:45 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003330 <- 0.004615 <-> 0.007945
20021109 09:08:46 MQM request from 131.188.3.150 -> 0.003566
20021109 09:08:46 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003337 <- 0.004554 <-> 0.007891
20021109 09:08:46 MQM request from 131.188.3.150 -> 0.003534
20021109 09:08:46 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003315 <- 0.004546 <-> 0.007861
20021109 09:08:47 MQM request from 131.188.3.150 -> 0.003593
20021109 09:08:47 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003350 <- 0.007515 <-> 0.010865
20021109 09:08:47 MQM request from 131.188.3.150 -> 0.003615
20021109 09:08:47 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003332 <- 0.004582 <-> 0.007914
20021109 09:08:48 MQM request from 131.188.3.150 -> 0.003542
20021109 09:08:48 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003325 <- 0.004530 <-> 0.007855
20021109 09:08:49 MQM request from 131.188.3.150 -> 0.003597
20021109 09:08:49 MQM response for 131.188.3.150 from 131.188.34.77 -> 0.003390 <- 0.004913 <-> 0.008303
^C
mc{unrzf1}[~/src/mqm]>
```

Figure D.3 - MQM Packets received at Host mc

```
faui40p{fodressl}[~/src/mqm]> ./mqm_receiver -T 47 -r 224.42.42.42 -p 4242
20021109 09:08:44 MQM request from 131.188.3.150 -> 0.003521
20021109 09:08:44 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003826 <- 0.002320 <-> 0.006146
20021109 09:08:44 MQM request from 131.188.3.150 -> 0.003347
20021109 09:08:44 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003604 <- 0.001060 <-> 0.004664
20021109 09:08:45 MQM request from 131.188.3.150 -> 0.003338
20021109 09:08:45 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003597 <- 0.001305 <-> 0.004902
20021109 09:08:45 MQM request from 131.188.3.150 -> 0.003330
20021109 09:08:45 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003580 <- 0.001080 <-> 0.004660
20021109 09:08:46 MQM request from 131.188.3.150 -> 0.003337
20021109 09:08:46 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003566 <- 0.001052 <-> 0.004618
20021109 09:08:46 MQM request from 131.188.3.150 -> 0.003315
20021109 09:08:46 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003534 <- 0.001085 <-> 0.004619
20021109 09:08:47 MQM request from 131.188.3.150 -> 0.003350
20021109 09:08:47 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003593 <- 0.001106 <-> 0.004699
20021109 09:08:47 MQM request from 131.188.3.150 -> 0.003332
20021109 09:08:47 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003615 <- 0.001079 <-> 0.004694
20021109 09:08:48 MQM request from 131.188.3.150 -> 0.003325
20021109 09:08:48 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003542 <- 0.001120 <-> 0.004662
20021109 09:08:49 MQM request from 131.188.3.150 -> 0.003390
20021109 09:08:49 MQM response for 131.188.3.150 from 192.44.88.100 -> 0.003597 <- 0.001080 <-> 0.004677
^C
faui40p{fodressl}[~/src/mqm]>
```

Figure D.4 - MQM Packets received at Host faui40p

The results of the measurement of the round-trip times which have been shown in the last few figures are summarized in the graph presented in figure D.5. Host lisa was sending MQM ping requests and analyzing the received response messages.

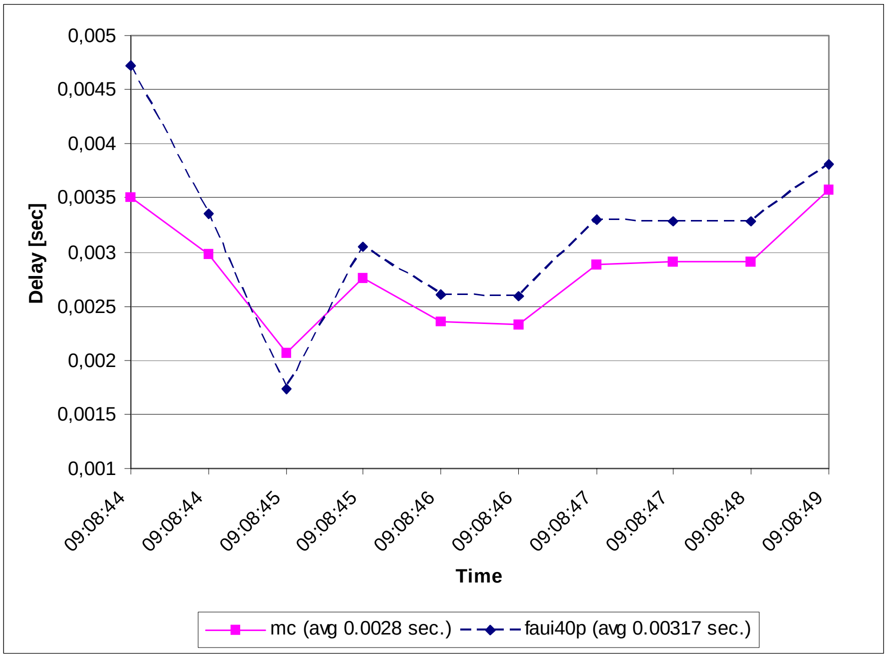


Figure D.5 - Results of the Delay Measurements

In addition to these values, the round-trip time between mc and faui40p can be calculated using the single OWD measurements. The result is shown in figure D.6.

The results of the single one-way delay measurements are not shown due to the imprecise values. The problem occurred due to a low synchronization of the different clocks.

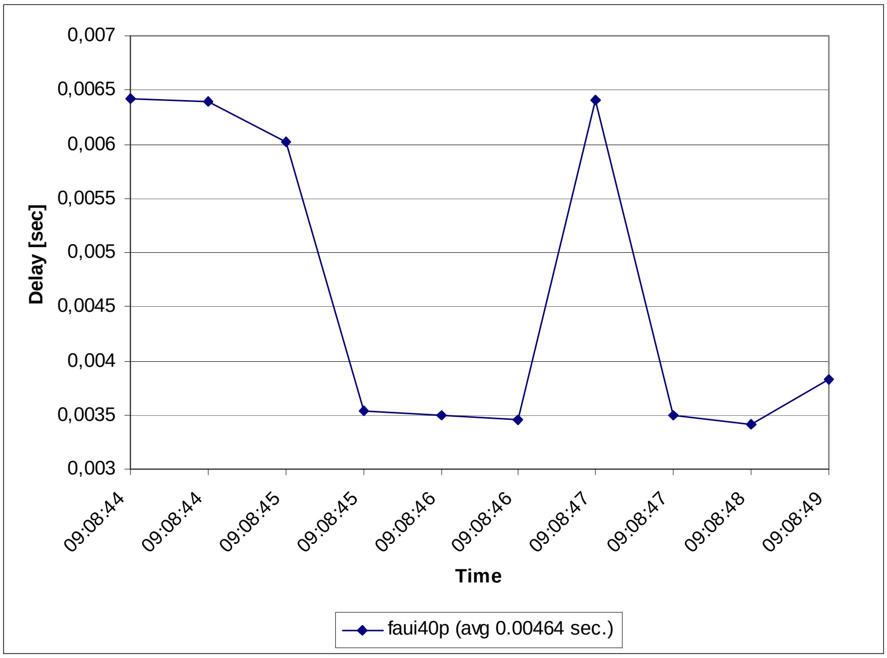


Figure D.6 - Calculation of the RTT using single OWD Measurements

D.2 Analysis of RTP Packets

In a second test, RTP packets have been received and analyzed by the mqm_receiver. The configuration used for this example is shown in figure D.7. The mqm_receiver was running on the host lisa.

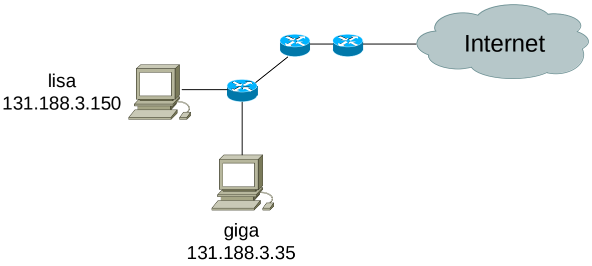
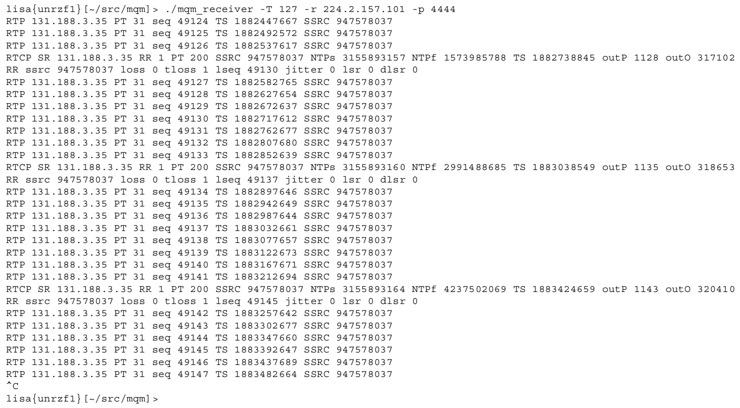


Figure D.7 - Network Configuration used for the RTP Examples

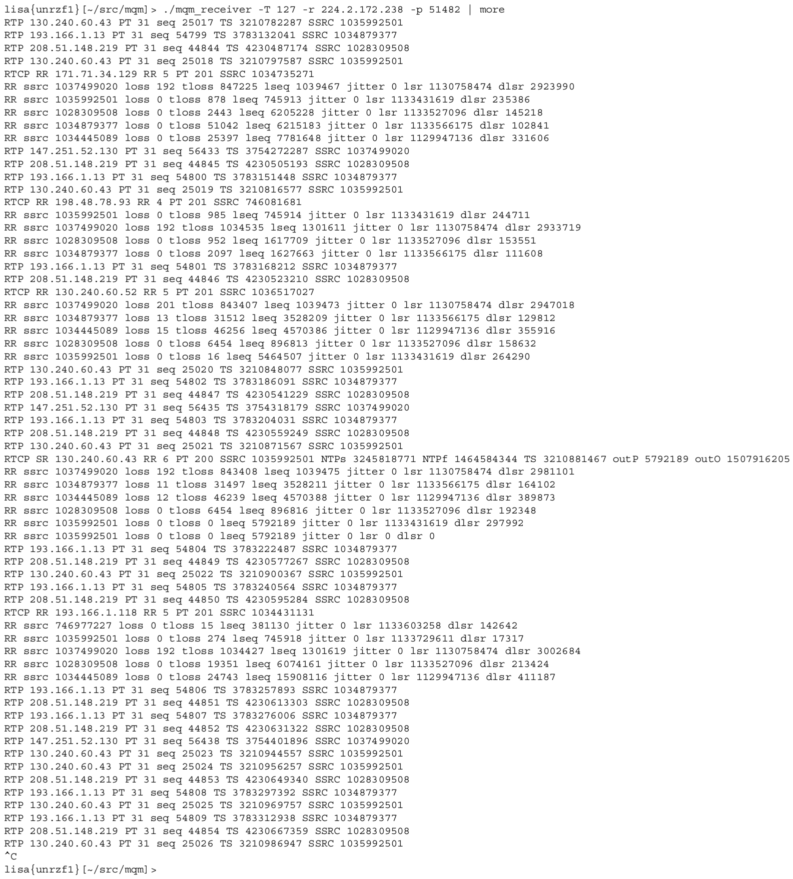
First, an RTP stream created by a local video server has been analyzed. The host giga is a server broadcasting recorded lectures from the project Uni-TV. In idle times, it transmits a low bandwidth "pause" video. The test was started during such a phase. The results of the analysis of the RTP and RTCP packets are shown in figure D.8.

Each line starting with the keyword RTP represents a single received RTP packet. Only packets from a single source (giga) have been received. Additionally, RTCP packets have been analyzed. Rows starting with RTCP show the content of a received RTCP packet. Because no active receivers are online, only sender reports have been found. The SRs provide information about the total number of sent packet and the SSRC of this sender. In addition, they contain a number of receiver reports. In this case, only a single RR is shown which the receiver report from the sender itself is.



```
lisa{unrzf1}[~/src/mqm]> ./mqm_receiver -T 127 -r 224.2.157.101 -p 4444
RTP 131.188.3.35 PT 31 seq 49124 TS 1882447667 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49125 TS 1882492572 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49126 TS 1882537617 SSRC 947578037
RTCP SR 131.188.3.35 RR 1 PT 200 SSRC 947578037 NTPs 3155893157 NTPf 1573985788 TS 1882738845 outP 1128 outO 317102
RR ssrc 947578037 loss 0 tloss 1 lseq 49130 jitter 0 lsr 0 dlsr 0
RTP 131.188.3.35 PT 31 seq 49127 TS 1882582765 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49128 TS 1882627654 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49129 TS 1882672637 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49130 TS 1882717612 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49131 TS 1882762677 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49132 TS 1882807680 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49133 TS 1882852639 SSRC 947578037
RTCP SR 131.188.3.35 RR 1 PT 200 SSRC 947578037 NTPs 3155893160 NTPf 2991488685 TS 1883038549 outP 1135 outO 318653
RR ssrc 947578037 loss 0 tloss 1 lseq 49137 jitter 0 lsr 0 dlsr 0
RTP 131.188.3.35 PT 31 seq 49134 TS 1882897646 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49135 TS 1882942649 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49136 TS 1882987644 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49137 TS 1883032661 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49138 TS 1883077657 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49139 TS 1883122673 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49140 TS 1883167671 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49141 TS 1883212694 SSRC 947578037
RTCP SR 131.188.3.35 RR 1 PT 200 SSRC 947578037 NTPs 3155893164 NTPf 4237502069 TS 1883424659 outP 1143 outO 320410
RR ssrc 947578037 loss 0 tloss 1 lseq 49145 jitter 0 lsr 0 dlsr 0
RTP 131.188.3.35 PT 31 seq 49142 TS 1883257642 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49143 TS 1883302677 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49144 TS 1883347660 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49145 TS 1883392647 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49146 TS 1883437689 SSRC 947578037
RTP 131.188.3.35 PT 31 seq 49147 TS 1883482664 SSRC 947578037
^C
lisa{unrzf1}[~/src/mqm]>
```

Figure D.8 - Analysis of a RTP Stream from Host giga



```
lisa{unrzf1}[~/src/mqm]> ./mqm_receiver -T 127 -r 224.2.172.238 -p 51482 | more
RTP 130.240.60.43 PT 31 seq 25017 TS 3210782287 SSRC 1035992501
RTP 193.166.1.13 PT 31 seq 54799 TS 3783132041 SSRC 1034879377
RTP 208.51.148.219 PT 31 seq 44844 TS 4230487174 SSRC 1028309508
RTP 130.240.60.43 PT 31 seq 25018 TS 3210797587 SSRC 1035992501
RTCP RR 171.71.34.129 RR 5 PT 201 SSRC 1034735271
RR ssrc 1037499020 loss 192 tloss 847225 lseq 1039467 jitter 0 lsr 1130758474 dlsr 2923990
RR ssrc 1035992501 loss 0 tloss 878 lseq 745913 jitter 0 lsr 1133431619 dlsr 235386
RR ssrc 1028309508 loss 0 tloss 2443 lseq 6205228 jitter 0 lsr 1133527096 dlsr 145218
RR ssrc 1034879377 loss 0 tloss 51042 lseq 6215183 jitter 0 lsr 1133566175 dlsr 102841
RR ssrc 1034445089 loss 0 tloss 25397 lseq 7781648 jitter 0 lsr 1129947136 dlsr 331606
RTP 147.251.52.130 PT 31 seq 56433 TS 3754272287 SSRC 1037499020
RTP 208.51.148.219 PT 31 seq 44845 TS 4230505193 SSRC 1028309508
RTP 193.166.1.13 PT 31 seq 54800 TS 3783151448 SSRC 1034879377
RTP 130.240.60.43 PT 31 seq 25019 TS 3210816577 SSRC 1035992501
RTCP RR 198.48.78.93 RR 4 PT 201 SSRC 746081681
RR ssrc 1035992501 loss 0 tloss 985 lseq 745914 jitter 0 lsr 1133431619 dlsr 244711
RR ssrc 1037499020 loss 192 tloss 1034535 lseq 1301611 jitter 0 lsr 1130758474 dlsr 2933719
RR ssrc 1028309508 loss 0 tloss 952 lseq 1617709 jitter 0 lsr 1133527096 dlsr 153551
RR ssrc 1034879377 loss 0 tloss 2097 lseq 1627663 jitter 0 lsr 1133566175 dlsr 111608
RTP 193.166.1.13 PT 31 seq 54801 TS 3783168212 SSRC 1034879377
RTP 208.51.148.219 PT 31 seq 44846 TS 4230523210 SSRC 1028309508
RTCP RR 130.240.60.52 RR 5 PT 201 SSRC 1036517027
RR ssrc 1037499020 loss 201 tloss 843407 lseq 1039473 jitter 0 lsr 1130758474 dlsr 2947018
RR ssrc 1034879377 loss 13 tloss 31512 lseq 3528209 jitter 0 lsr 1133566175 dlsr 129812
RR ssrc 1034445089 loss 15 tloss 46256 lseq 4570386 jitter 0 lsr 1129947136 dlsr 355916
RR ssrc 1028309508 loss 0 tloss 6454 lseq 896813 jitter 0 lsr 1133527096 dlsr 158632
RR ssrc 1035992501 loss 0 tloss 16 lseq 5464507 jitter 0 lsr 1133431619 dlsr 264290
RTP 130.240.60.43 PT 31 seq 25020 TS 3210848077 SSRC 1035992501
RTP 193.166.1.13 PT 31 seq 54802 TS 3783186091 SSRC 1034879377
RTP 208.51.148.219 PT 31 seq 44847 TS 4230541229 SSRC 1028309508
RTP 147.251.52.130 PT 31 seq 56435 TS 3754318179 SSRC 1037499020
RTP 193.166.1.13 PT 31 seq 54803 TS 3783204031 SSRC 1034879377
RTP 208.51.148.219 PT 31 seq 44848 TS 4230559249 SSRC 1028309508
RTP 130.240.60.43 PT 31 seq 25021 TS 3210871567 SSRC 1035992501
RTCP SR 130.240.60.43 RR 6 PT 200 SSRC 1035992501 NTPs 3245818771 NTPf 1464584344 TS 3210881467 outP 5792189 outO 1507916205
RR ssrc 1037499020 loss 192 tloss 843408 lseq 1039475 jitter 0 lsr 1130758474 dlsr 2981101
RR ssrc 1034879377 loss 11 tloss 31497 lseq 3528211 jitter 0 lsr 1133566175 dlsr 164102
RR ssrc 1034445089 loss 12 tloss 46239 lseq 4570388 jitter 0 lsr 1129947136 dlsr 389873
RR ssrc 1028309508 loss 0 tloss 6454 lseq 896816 jitter 0 lsr 1133527096 dlsr 192348
RR ssrc 1035992501 loss 0 tloss 0 lseq 5792189 jitter 0 lsr 1133431619 dlsr 297992
RR ssrc 1035992501 loss 0 tloss 0 lseq 5792189 jitter 0 lsr 0 dlsr 0
RTP 193.166.1.13 PT 31 seq 54804 TS 3783222487 SSRC 1034879377
RTP 208.51.148.219 PT 31 seq 44849 TS 4230577267 SSRC 1028309508
RTP 130.240.60.43 PT 31 seq 25022 TS 3210900367 SSRC 1035992501
RTP 193.166.1.13 PT 31 seq 54805 TS 3783240564 SSRC 1034879377
RTP 208.51.148.219 PT 31 seq 44850 TS 4230595284 SSRC 1028309508
RTCP RR 193.166.1.118 RR 5 PT 201 SSRC 1034431131
RR ssrc 746977227 loss 0 tloss 15 lseq 381130 jitter 0 lsr 1133603258 dlsr 142642
RR ssrc 1035992501 loss 0 tloss 274 lseq 745918 jitter 0 lsr 1133729611 dlsr 17317
RR ssrc 1037499020 loss 192 tloss 1034427 lseq 1301619 jitter 0 lsr 1130758474 dlsr 3002684
RR ssrc 1028309508 loss 0 tloss 19351 lseq 6074161 jitter 0 lsr 1133527096 dlsr 213424
RR ssrc 1034445089 loss 0 tloss 24743 lseq 15908116 jitter 0 lsr 1129947136 dlsr 411187
RTP 193.166.1.13 PT 31 seq 54806 TS 3783257893 SSRC 1034879377
RTP 208.51.148.219 PT 31 seq 44851 TS 4230613303 SSRC 1028309508
RTP 193.166.1.13 PT 31 seq 54807 TS 3783276006 SSRC 1034879377
RTP 208.51.148.219 PT 31 seq 44852 TS 4230631322 SSRC 1028309508
RTP 147.251.52.130 PT 31 seq 56438 TS 3754401896 SSRC 1037499020
RTP 130.240.60.43 PT 31 seq 25023 TS 3210944557 SSRC 1035992501
RTP 130.240.60.43 PT 31 seq 25024 TS 3210956257 SSRC 1035992501
RTP 208.51.148.219 PT 31 seq 44853 TS 4230649340 SSRC 1028309508
RTP 193.166.1.13 PT 31 seq 54808 TS 3783297392 SSRC 1034879377
RTP 130.240.60.43 PT 31 seq 25025 TS 3210969757 SSRC 1035992501
RTP 193.166.1.13 PT 31 seq 54809 TS 3783312938 SSRC 1034879377
RTP 208.51.148.219 PT 31 seq 44854 TS 4230667359 SSRC 1028309508
RTP 130.240.60.43 PT 31 seq 25026 TS 3210986947 SSRC 1035992501
^C
lisa{unrzf1}[~/src/mqm]>
```

Figure D.9 - Analysis of RTP Streams from a number of Hosts in the Internet

For a second example, shown in figure D.9, the mqm_receiver has been used to analyze the packets received on a popular multicast group called "Places all over the world". At any time, there are a number of hosts transmitting videos to this group. During the running time of the mqm_receiver, RTP packets from four different sources have been seen. The analysis of the received RTCP packets provides information about the transmission quality between the single hosts. A RR contains a field for the number of lost packets as well as for the current jitter.

Bibliography

[1] A. Adams, R. Bu, R. Caceres, N. Duffield, T. Friedman, J. Horowitz, F. Lo Presti, S. Moon, V. Paxson, D. Towsley, "The use of end-to-end multicast measurements for characterizing internal network behavior," IEEE Communications Magazine, May 2000.

[2] Z. Albanna, K. Almeroth, D. Meyer, M. Schipper, "IANA Guidelines for IPv4 Multicast Address Assignments," RFC 3171, IETF, August 2001.

[3] K. Almeroth, M. Ammar, "Collecting and Modeling the Join/Leave Behavior of Multicast Group Members in the MBone," Proceedings of HPDC'96, Syracuse, New York, August 1996.

[4] K. Almeroth, M. Ammar, "Multicast Group Membership Collection Tool (mlisten)," Georgia Institute of Technology, September 1996. http://www.cc.gatech.edu/computing/Telecomm/mbone/.

[5] K. Almeroth, M. Ammar, "Multicast Group Behavior in the Internet's Multicast Backbone (MBone)," IEEE Communications, June 1997.

[6] K. Almeroth, "Managing IP Multicast Traffic: A First Look at the Issues, Tools, and Challenges," IP Multicast Initiative Summit, San Jose, California, USA, February 1999.

[7] K. Almeroth, L. Wei, "Justification for and use of the Multicast Routing Monitor (MRM) Protocol," Internet Draft, draft-ietf-mboned-mrm-use-00, IETF, February 1999.

[8] K. Almeroth, "The Evolution of Multicast: From the MBone to Inter-Domain Multicast to Internet2 Deployment," IEEE Network, Volume 14 Issue 1, January/February 2000.

[9] K. Almeroth, "A Long-Term Analysis of Growth and Usage Patterns in the Multicast Backbone (MBone)," Proceedings of IEEE Conference on Computer Communications (INFOCOM'00), Tel Aviv, Israel, March 2000.

[10] K. Almeroth, K. Sarac, L. Wei, "Supporting Multicast Management Using the Multicast Reachability Monitor (MRM) Protocol," Technical Report TR2000-26, University of California - Santa Barbara, May 2000.

[11] K. Almeroth, L. Wei, D. Farinacci, "Multicast Reachability Monitor (MRM)," Internet Draft, draft-ietf-mboned-mrm-01, IETF, July 2000.

[12] G. Almes, S. Kalidindi, M. Zekauskas, "A One-way Delay Metric for IPPM," RFC 2679, IETF, September 1999.

[13] G. Almes, S. Kalidindi, M. Zekauskas, "A One-way Packet Loss Metric for IPPM," RFC 2680, IETF, September 1999.

[14] W. Almesberger, "Linux Network Traffic Control - Implementation Overview," April 1999. http://icawww1.epfl.ch/linux-diffserv/.

[15] W. Almesberger, J. Salim, A. Kuznetsov, "Differentiated Services on Linux," Internet Draft, draft-almesgerber-wajhak-diffserv-linux-01, IETF, June 1999.

[16] S. Armstrong, A. Freier, K. Marzullo, "Multicast Transport Protocol," RFC 1301, IETF, February 1992.

[17] J. Atwood, R. Mukherjee, "RP Relocation in PIM-SM Multicast," Internet Draft, draft-atwood-pim-sm-rp-01, IETF, November 2001.

[18] B. Awerbuch, "Complexity of network synchronization," Journal of the ACM (JACM), Volume 32 Issue 4, October 1985.

[19] B. Awerbuch, A. Baratz, D. Peleg, "Cost-sensitive analysis of communication protocols," Proceedings of the ninth annual ACM symposium on Principles of distributed computing, Quebec City, Quebec, Canada, August 1990.

[20] A. Ballardie, "Core Based Trees (CBT version 2) Multicast Routing - Protocol Specification," RFC 2189, IETF, September 1997.

[21] A. Ballardie, "Core Based Trees (CBT) Multicast Routing Architecture," RFC 2201, IETF, September 1997.

[22] J. Bansemer, M. Eltoweissy, "On performance metrics for IP multicast routing," Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN 2000), Dallas, TX, 2000.

[23] T. Bates, Y. Rekhter, R. Chandra, D. Katz, "Multiprotocol Extensions for BGP-4," RFC 2858, IETF, June 2000.

[24] S. Bhattacharyya, C. Diot, L. Giuliano, R. Rockwell, J. Meylor, D. Meyer, G. Shepherd, "A Framework for Source-Specific IP Multicast Deployment," Internet-Draft, draft-bhattach-pim-ssm-00, IETF, July 2000.

[25] S. Bhattacharyya, C. Diot, L. Giuliano, R. Rockwell, J. Meylor, D. Meyer, G. Shepherd, B. Haberman, "An Overview of Source-Specific Multicast (SSM)," Internet Draft, draft-ietf-ssm-overview-03, IETF, March 2002.

[26] U. Black, "IP Routing Protocols - RIP, OSPF, BGP, PNNI, and Cisco Routing Protocols," Prentice Hall PTR, Upper Saddle River, New Jersey, 2000.

[27] S. Blake, D. Black, M. Carlson, E. Davis, Z. Wang, W. Weiss, "An Architecture for Differentiated Services," RFC 2475, IETF, December 1998.

[28] R. Boivie, N. Feldman, Y. Imai, W. Livens, D. Ooms, O. Paridaens, "Explicit Multicast (Xcast) Basic Specification," Internet Draft, draft-ooms-xcast-basic-spec-03, IETF, June 2002.

[29] J. Bolot, T. Turletti, I. Wakeman, "Scalable feedback control for multicast video distribution in the Internet," ACM SIGCOMM Computer Communication Review, Proceedings of the conference on Communications architectures, protocols and applications, Volume 24 Issue 4, October 1994.

[30] G. Booch, J. Rumbaugh, I. Jacobson, "The Unified Modeling Language User Guide," Addison Wesley, Reading, 1998.

[31] A. Bouch, A. Watson, M. Sasse, "QUASS - A Tool for Measuring the Subjective Quality of Real-Time Multimedia Audio and Video," Poster presented at HCI '98, Sheffield, England, 1-4 September 1998.

[32] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview," RFC 1633, IETF, June 1994.

[33] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification," RFC 2205, IETF, September 1997.

[34] R. Braudes, S. Zabele, "Requirements for Multicast Protocols," RFC 1458, IETF, May 1993.

[35] L. Breslau, S. Shenker, "Best-Effort versus Reservations: A Simple Comparative Analysis," ACM SIGCOMM Computer Communication Review, Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication, Volume 28 Issue 4, October 1998.

[36] I. Busse, B. Deffner, H. Schulzrinne, "Dynamic QoS Control of Multimedia Applications based on RTP," Proceedings to First International Workshop on High Speed Networks and Open Distributed Platforms, St. Petersburg, Russia, June 1995.

[37] J. Cadzow, "Foundations of digital signal processing and data analysis," Macmillan, New York, 1987.

[38] B. Cain, S. Deering, B. Fenner, I. Kouvelas, A. Thyagarajan, "Internet Group Management Protocol, Version 3," Internet Draft, draft-ietf-idmr-igmp-v3-11, IETF, May 2002.

[39] S. Casner, S. Deering, "First IETF internet audiocast," ACM SIGCOMM Computer Communication Review, Volume 22 Issue 3, July 1992.

[40] R. Chalmers, K. Almeroth, "Developing a Multicast Metric," Proceedings of IEEE Globecom, San Francisco, CA, USA, December 2000.

[41] K. Chen, M. Kutzko, T. Rimovsky, "Multicast Beacon Server v0.8," January 2002. http://dast.nlanr.net/projects/beacon/.

[42] M. Christensen, F. Solensky, "IGMP and MLD snooping switches," Internet Draft, draft-ietf-magma-snoop-02, IETF, June 2002.

[43] J. Chuang, M. Sirbu, "Pricing Multicast Communications: A Cost-Based Approach," Proceedings of the Internet Society INET'98 Conference, Geneva, Switzerland, July 1998.

[44] S. Cook, "The complexity of theorem-proving procedures," Proceedings of the third annual ACM symposium on Theory of computing, Shaker Heights, Ohio, USA, May 1971.

[45] T. Cormen, C. Leiserson, R. Rivest, "Introduction to Algorithms," MIT Press, Cambridge, Massachusetts, 1997.

[46] J. Crowcroft, K. Paliwoda, "A multicast transport protocol," ACM SIGCOMM Computer Communication Review, Symposium proceedings on Communications architectures and protocols, Volume 18 Issue 4, August 1988.

[47] J. Crowcroft, M. Handley, I. Wakeman, "Internetworking Multimedia," Morgan Kaufmann Publishers, San Francisco, California, 1999.

[48] Y. Dalal, R. Metcalfe, "Reverse path forwarding of broadcast packets," Communications of the ACM, Volume 21 Issue 12, December 1978.

[49] J. Davidson, J. Peters, "Voice over IP Fundamentals," Cisco Press, Indianapolis, IN, 2000.

[50] S. Deering, D. Cheriton, "Host Groups: A Multicast Extension to the Internet Protocol," RFC 966, IETF, December 1985.

[51] S. Deering, "Host Extentions for IP Multicasting," RFC 1112, IETF, August 1989.

[52] S. Deering, D. Cheriton, "Multicast routing in datagram internetworks and extended LANs," ACM Transactions on Computer Systems (TOCS), Volume 8 Issue 2, May 1990.

[53] S. Deering, "Multicast Routing in a Datagram Internetwork," Ph.D. thesis, Stanford University, December 1991.

[54] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, L. Wei, "An architecture for wide-area multicast routing," ACM SIGCOMM Computer Communication Review, Proceedings of the conference on Communications architectures, protocols and applications, Volume 24 Issue 4, October 1994.

[55] S. Deering, "Multicast routing in internetworks and extended LANs," ACM SIGCOMM Computer Communication Review, Volume 25 Issue 1, January 1995.

[56] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, L. Wei, "The PIM architecture for wide-area multicast routing," IEEE/ACM Transactions on Networking (TON), Volume 4 Issue 2, April 1996.

[57] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, A. Helmy, D. Meyer, L. Wei, "Protocol Independent Multicast Version 2 Dense Mode Specification," Internet Draft, draft-ietf-pim-v2-dm-03, IETF, June 1999.

[58] C. Demichelis, "Improvement of the Instantaneous Packet Delay Variation (IPDV) Concept and Applications," Proceedings to World Telecommunications Congress 2000, Birmingham, May 7-12, 2000.

[59] C. Demichelis, P. Chimento, "IP Packet Delay variation Metric for IPPM," Internet Draft, draft-ietf-ippm-ipdv-09, IETF, April 2002.

[60] E. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik 1, 1959.

[61] C. Diot, B. Levine, B. Lyles, H. Kassem, D. Balensiefen, "Deployment Issues for the IP Multicast Service and Architecture," IEEE Network Magazine, Volume 14 Number 1, January/February 2000.

[62] F. Dressler, S. Nägele-Jackson, K. Liebl, P. Holleczek, "Project Uni-TV," Dezember 1998. http://www.uni-tv.net/index_en.html.

[63] F. Dressler, U. Hilgers, S. Nägele-Jackson, K. Liebl, "Untersuchung von Dienstqualitäten bei echtzeitorientierten multimedialen Datenübertragungen," Multimedia und Automatisierung: Fachtagung der GI-Fachgruppe 4.4.2 Echtzeitprogrammierung, PEARL’99, Boppard, November 1999 / PEARL ’99, Workshop über Realzeitsysteme. Peter Holleczek (Hrsg.). Springer, Berlin, 1999 (Informatik aktuell).

[64] F. Dressler, U. Hilgers, "Routing mit QoS-Eigenschaften unter Linux," Echtzeitbetriebssysteme und LINUX: Fachtagung der GI-Fachgruppe 4.4.2 Echtzeitprogrammierung, PEARL 2000, Boppard, November 2000 / PEARL 2000, Workshop über Realzeitsysteme. Peter Holleczek (Hrsg.). Springer, Berlin, 2000 (Informatik aktuell).

[65] F. Dressler, U. Hilgers, P. Holleczek, "Voice over IP in Weitverkehrsnetzen?," Anwendungs- und System-Management im Zeichen von Multimedia und E-Business: Fachtagung der GI-Fachgruppe 3.4 Betrieb von Informations- und Kommunikationssystemen, BIK 2001, Tübingen, April 2001.

[66] F. Dressler, "How to Measure Reliability and Quality of IP Multicast Services?," Proceedings of 2001 IEEE Pacific Rim Conference on Communications, Computer and Signal Processing (IEEE PACRIM'01), Victoria, B.C., Canada, August 2001.

[67] F. Dressler, "QoS considerations on IP multicast services," Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Education, Science, and Medicine on the Internet (SSGRR 2002w), L'Aquila, Italy, January 2002.

[68] F. Dressler, "IP Multicast," Vorlesung im Rahmen der Vorlesung 'Grundzüge der Datenkommunikation', RRZE, Universität Erlangen-Nürnberg, July 2002.

[69] F. Dressler, "Advantages of VoIP in the german research network," Proceedings of 5th IEEE International Conference on High Speed Networks and Multimedia Communications (IEEE HSNMC 2002), Jeju Islands, Korea, July 2002.

[70] F. Dressler, "An Approach for QoS Measurements in IP Multicast Networks, MQM - Multicast Quality Monitor," Proceedings of Third International Network Conference (INC 2002), Plymouth, UK, July 2002.

[71] F. Dressler, "MQM - Multicast Quality Monitor," Proceedings of 10th International Conference on Telecommunication Systems, Modeling and Analysis (ICSTM10), Monterey, CA, USA, October 2002.

[72] F. Dressler, "TCP/IP Grundlagen," Vorlesung im Rahmen der Vorlesung 'Grundzüge der Datenkommunikation', RRZE, Universität Erlangen-Nürnberg, October 2002.

[73] H. Eriksson, "MBONE: the multicast backbone," Communications of the ACM, Volume 37 Issue 8, August 1994.

[74] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, L. Wei, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification," RFC 2362, IETF, June 1998.

[75] K. Fall, K. Varadhan, "The ns Manual (formerly ns Notes and Documentation)," The VINT Project, Collaboration between UC Berkeley, LBL, USC/ISI, Xerox PARC, April 2002. http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf.

[76] M. Faloutsos, A. Banerjea, R. Pankaj, "QoSMIC: Quality of Service sensitive Multicast Internet protoCol," ACM SIGCOMM Computer Communication Review, Proceedings of the ACM SIGCOMM '98 Conference on Applications, technologies, architectures, and protocols for computer communication, Volume 28 Issue 4, October 1998.

[77] A. Feldmann, A. Gilbert, P. Huang, W. Willinger, "Dynamics of IP traffic," ACM SIGCOMM Computer Communication Review, Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication, Volume 29 Issue 4, August 1999.

[78] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, F. True, "Deriving traffic demands for operational IP networks," ACM SIGCOMM Computer Communication Review, Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, Volume 30 Issue 4, August 2000.

[79] W. Fenner, S. Casner, "A ’traceroute’ facility for IP Multicast," Internet Draft, draft-ietf-idmr-traceroute-ipm-07, IETF, July 2000.

[80] B. Fenner, "Multicast Traceroute (mtrace) 5.1," December 1996. ftp://ftp.parc.xerox.com/pub/net-research/ipmulti/.

[81] B. Fenner, M Handley, R. Kermode, D. Thaler, "Bootstrap Router (BSR) Mechanism for PIM Sparse Mode, " Internet Draft, draft-ietf-pim-bsr-02, IETF, November 2001.

[82] B. Fenner, M. Handely, H. Holbrook, I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)," Internet Draft, draft-ietf-pim-sm-v2-new-05, IETF, March 2002.

[83] W. Fenner, "Internet Group Management Protocol, Version 2," RFC 2236, IETF, November 1997.

[84] D. Ferrari, "Client Requirements for Real-Time Communication Services," RFC 1193, IETF, November 1990.

[85] T. Friedman, R. Caceres, K. Almeroth, K. Sarac, "RTCP Reporting Extensions," Internet Draft, draft-friedman-avt-rtcp-report-extns-02, IETF, February 2002.

[86] M. Garey, D. Johnson, "Computers and Intractability: A Guide to the Theory of NPCompleteness," Freeman and Co., New York, 1979.

[87] S. Gorinsky, H. Vin, "The utility of feedback in layered multicast congestion control," 11th International workshop on on Network and Operating Systems support for digital audio and video, January 2001.

[88] R. Gotzhein, "Modellierung und Spezifikation von Diensten und Verhalten in verteilten Systemen," Dissertation, Universität Erlangen-Nürnberg, 1985.

[89] B. Grönvall, I. Marsh, S. Pink, "Caching: A multicast-based distributed file system for the internet," Proceedings of the seventh workshop on ACM SIGOPS European workshop, September 1996.

[90] D. Grossman, "New Terminology for Diffserv," Internet Draft, draft-ietf-diffserv-new-terms-02, IETF, November 1999.

[91] B. Haberman, J. Martin, "IGMPv3 and Multicast Routing Protocol Interaction," Internet Draft, draft-ietf-magma-igmpv3-and-routing-02, IETF, February 2002.

[92] S. Halabi, "Internet Routing Architectures," Second Edition, Cisco Press, Indianapolis, IN, 2001.

[93] M. Handley, "SDR: Session Directory Tool," University College London, November 1995. ftp://cs.ucl.ac.uk/mice/sdr/.

[94] M. Handley, J. Crowcroft, "Network text editor (NTE): A scalable shared text editor for the MBone," ACM SIGCOMM Computer Communication Review, Proceedings of the ACM SIGCOMM '97 conference on Applications, technologies, architectures, and protocols for computer communication, Volume 27 Issue 4. October 1997.

[95] M. Handley, "Multicast Address Allocation Protocol (AAP)," Internet Draft, draft-draft-handley-aap-00, IETF, December 1997.

[96] M. Handley, V. Jacobson, "SDP: Session Description Protocol," RFC 2327, IETF, April 1998.

[97] M. Handley, C. Perkins, E. Whelan, "Session Announcement Protocol," RFC 2974, IETF, October 2000.

[98] M. Handley, D. Thaler, R. Kermode, "Multicast-Scope Zone Announcement Protocol (MZAP)," RFC 2776, IETF, February 2000.

[99] M. Handley, I. Kouvelas, T. Speakman, L. Vicisano, "Bi-directional Protocol Independent Multicast (BIDIR-PIM)," Internet Draft, draft-ietf-pim-bidir-04, IETF, June 2002.

[100] M. Handley, V. Jacobson, C. Perkins, "SDP: Session Description Protocol," Internet Draft, draft-ietf-mmusic-sdp-new-10, IETF, May 2002.

[101] S. Hanna, B. Patel, M. Sah, "Multicast Address Dynamic Client Allocation Protocol (MADCAP)," RFC 2730, IETF, December 1999.

[102] V. Hardman, I. Kouvelas, "Robust Audio Tool (RAT)," University College London, 1995. http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/.

[103] V. Hardman, A. Sasse, M. Handley, A. Watson, "Reliable Audio for Use over the Internet," Proceedings of INET'95, Honolulu, Hawaii, June 1995.

[104] C. Hedrick, "Routing Information Protocol," RFC 1058, IETF, June 1988.

[105] G. Held, "Local Area Network Performance: Issues and Answers," Wiley Communications Technology, John Wiley & Sons (Sd), Juli 1994.

[106] U. Hilgers, F. Dressler, "Echtzeit-Datenverkehr über IP-basierte Datennetze," Echtzeitkommunikation und Ethernet/Internet: Fachtagung der GI-Fachgruppe 4.4.2 Echtzeitprogrammierung, PEARL 2001, Boppard, November 2001 / PEARL 2001, Workshop über Realzeitsysteme. P. Holleczek, B. Vogel-Heuser (Hrsg.). Springer, Berlin, 2001 (Informatik aktuell).

[107] E. Hellfritsch, "Projekt TKBRZL," August 1998. http://tkbrzl.bhn.de/.

[108] F. Hofmann, "Betriebssysteme: Grundkonzepte und Modellvorstellungen," 2. Auflage, B.G. Teubner, Stuttgart, 1991.

[109] H. Holbrook, D. Cheriton, "IP multicast channels: EXPRESS support for large-scale single-source applications," ACM SIGCOMM Computer Communication Review, Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication, Volume 29 Issue 4, August 1999.

[110] H. Holbrook, B. Cain, "Using IGMPv3 For Source-Specific Multicast," Internet Draft, draft-holbrook-idmr-igmpv3-ssm-00, IETF, July 2000.

[111] H. Holbrook, B. Cain, "Source-Specific Multicast for IP," Internet Draft, draft-ietf-ssm-arch-00, IETF, November 2001.

[112] C. Huitema, "Routing in the Internet," 2nd Edition, Prentice Hall PTR, Upper Saddle River, New Jersey, 1999.

[113] V. Jacobson, S. McCanne, "VAT: Visual Audio Tool," Lawrence Berkeley Laboratory (LBL), February 1992. ftp://ee.lbl.gov/conferencing/vat/.

[114] V. Jacobson, S. McCanne, "WB: Whiteboard Tool," Lawrence Berkeley Laboratory (LBL), July 1994. ftp://ee.lbl.gov/conferencing/wb/.

[115] S. Jagannathan, K. Almeroth, A. Acharya, "Topology Sensitive Congestion Control for Real-Time Multicast," Proceedings of Network and Operating System Support for Digital Audio and Video (NOSSDAV '00), Chapel Hill, North Carolina, USA, June 2000.

[116] R. Jain, "The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling," John Wiley & Sons, Inc., New York, 1991.

[117] M. Kanbara, H. Tanioka, K. Kinoshita, K. Murakami, "A Multicast Routing Algorithm for Multiple QoS Requirements," Proceedings of Third International Network Conference, INC 2002, Plymouth, UK, July 16-18, 2002.

[118] S. Kasera, G. Hjálmtýsson, D. F. Towsley, J. F. Kurose, "Scalable reliable multicast using multiple multicast channels," IEEE/ACM Transactions on Networking (TON), Volume 8 Issue 3, June 2000.

[119] S. Keramidis, "Eine Methode zur Spezifikation und korrekten Implementierung von asynchronen Systemen," Habilitation, Universität Erlangen-Nürnberg, Februar 1982.

[120] M. Kodialam, T. Lakshman, S. Sengupta, "Online Multicast Routing with Bandwidth Guarantees: A New Approach using Multicast Network Flow," ACM SIGMETRICS Performance Evaluation Review, Proceedings of the international conference on International Conference on Measurements and modeling of computer systems, Volume 28 Issue 1, June 2000.

[121] V. Kompella, J. Pasquale, G. Polyzos, "Multicast routing for multimedia communication," IEEE/ACM Transactions on Networking (TON), Volume 1 Issue 3, June 1993.

[122] R. Koodli, R. Ravikanth, "One-way Loss Pattern Sample Metrics," Internet Draft, draft-ietf-ippm-loss-pattern-05, IETF, July 2001.

[123] L. Kou, G. Markowsky, L. Berman, "A faster algorithm for steiner trees," Acta Informatica, Volume 15, 1981.

[124] I. Kouvelas, V. Hardman, A. Watson, "Lip Synchronisation for use over the Internet: Analysis and Implementation," Proceedings of IEEE Globecom '96, London, UK, November 1996.

[125] S. Kumar, P. Radoslavov, D. Thaler, C. Alaettinoglu, D. Estrin, M. Handley, "The MASC/BGMP architecture for inter-domain multicast routing," ACM SIGCOMM Computer Communication Review, Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication, Volume 28 Issue 4, October 1998.

[126] B. Levine, J. Crowcroft, C. Diot, J. Garcia-Luna-Aceves, J. Kurose, "Consideration of receiver interest for IP multicast delivery," Proceedings of IEEE INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Volume 2, 2000.

[127] X. Li, M. Ammar, S. Paul, "Video multicast over the Internet," IEEE Network Magazine, Volume 13 Issue 2, March 1999.

[128] C. Liu, "Multimedia Over IP: RSVP, RTP, RTCP, RTSP," http://www.cis.ohio-state.edu/~cliu/ipmultimedia/.

[129] B. Mah, "Measurements and Observations of IP Multicast Traffic," Technical Report UCB/CSD-94-858, University of California, Berkeley, CA, December 1994.

[130] J. Mahdavi, V. Paxson, "IPPM Metrics for Measuring Connectivity," RFC 2678, IETF, September 1999.

[131] D. Makofske, K. Almeroth, "MHealth: A Real-Time Multicast Tree Visualization and Monitoring Tool," Proceedings of Network and Operating System Support for Digital Audio and Video (NOSSDAV '99), Basking Ridge, New Jersey, USA, June 1999.

[132] D. Makofske, K. Almeroth, "MHealth: A Real-Time Multicast Tree Visualization and Monitoring Tool," http://www.nmsl.cs.ucsb.edu/mhealth/.

[133] A. Mankin, F. Baker, B. Braden, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib, L. Zhang, "Resource ReSerVation Protocol (RSVP) Version 1 Applicability Statement Some Guidelines on Deployment," RFC 2208, IETF, September 1997.

[134] A. Mankin, A. Romanow, S. Bradner, V. Paxson, "IETF Criteria for Evaluating Reliable Multicast Transport Protocols," RFC 2357, IETF, June 1998.

[135] R. Malpani, E. Perry, "mmon: ip multicast management," HP, Inc., September 2000. http://www.hpl.hp.com/mmon/.

[136] N. Maxemchuk, D. Shur, "An Internet multicast system for the stock market," ACM Transactions on Computer Systems (TOCS), Volume 19 Issue 3, August 2001.

[137] M. McBride, J. Meylor, D. Meyer, "Multicast Source Discovery Protocol Deployment Scenarios," Internet Draft, draft-ietf-mboned-msdp-deploy-00, IETF, February 2002.

[138] S. McCanne, V. Jacobson, "VIC: Video Conference Tool," Lawrence Berkeley Laboratory (LBL), November1994. ftp://ee.lbl.gov/conferencing/vic/.

[139] S. McCanne, V. Jacobson, M. Vetterli, "Receiver-driven layered multicast," Proceedings of SIGCOMM Symposium on Communications Architectures and Protocols. Palo Alto, CA, USA, August 1996.

[140] C. Mercer, "An Introduction to Real-Time Operating Systems: Scheduling Theory," Unpublished manuscript, 1992.

[141] D. Meyer, "Introduction to IP Multicast," NANOG Meeting, Dearborn, June 1998.

[142] D. Meyer, "Administrativley Scoped IP Multicast," RFC 2365, IETF, July 1998.

[143] D. Meyer, B. Fenner, "Multicast Source Discovery Protocol (MSDP)," Internet Draft, draft-ietf-msdp-spec-11, IETF, August 2001.

[144] P. van Mieghem, G. Hooghiemstra, R. van der Hofstad, "On the efficiency of multicast," IEEE/ACM Transactions on Networking (TON), Volume 9 Issue 6, December 2001.

[145] C. Miller, "Multicast Networking and Applications," Addision-Wesley, Reading, Massachusetts, 1999.

[146] D. Mills, "Network Time Protocol (Version 3) Specification, Implementation and Analysis," RFC 1305, IETF, March 1992.

[147] A. Morton, L. Ciavattone, G. Ramachandran, S. Shalunov, J. Perser, "Reordering Metric for IPPM," Internet Draft, draft-ietf-ippm-reordering-00, IETF, June 2002.

[148] J. Moy, "Multicast Extensions to OSPF," RFC 1584, IETF, March 1994.

[149] J. Moy, "OSPF Version 2," RFC 2328, IETF, April 1998.

[150] K. Nichols, S. Blake, F. Baker, D. Black, "Definitions of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC 2474, IETF, December 1998.

[151] B. Nickless, "IPv4 Multicast Unusable Group Addresses," Internet Draft, deaft-nickless-ipv4-mcast-unusable-01, IETF, June 2002.

[152] B. Nickless, "The IP Multicast Service Model," Internet Draft, draft-nickless-mcast-svc-model-00, IETF, February 2002.

[153] B. Nickless, "IPv4 Multicast Best Current Practice," Internet Draft, draft-ietf-mboned-ipv4-mcast-bcp-00, IETF, February 2002.

[154] C. Noronha, F. Tobagi, "Optimum Routing of Multicast Streams," Proceedings of IEEE Infocom ’94, Toronto, Canada, June 1994.

[155] C. Noronha, F. Tobagi, "Evaluation of Multicast Routing Algorithms for Multimedia Streams," Proceedings of IEEE International Telecommunications Symposium, Rio de Janero, Brazil, August 1994.

[156] K. Obraczka, "Multicast Transport Protocols: A Survey and Taxonomy," IEEE Communications, January 1998.

[157] J. Pansiot, D. Grad, "On routes and multicast trees in the Internet," ACM SIGCOMM Computer Communication Review, Volume 28 Issue 1, January 1998.

[158] M. Parsa, Q. Zhu, J. Garcia-Luna-Aceves, "An iterative algorithm for delay-constrained minimum-cost multicasting," IEEE/ACM Transactions on Networking (TON), Volume 6 Issue 4, August 1998.

[159] V. Paxson, "Towards a Framework for Defining Internet Performance Metrics," Proceedings of INET‘96, 1996.

[160] V. Paxson, "Measurements and Analysis of End-to-End Internet Dynamics," Ph.D. thesis, University of California, Berkeley, April 1997.

[161] V. Paxson, G. Almes, J. Mahdavi, M. Mathis, "Framework for IP Performance Metrics," RFC 2330, IETF, May 1998.

[162] C. Perkins, I. Kouvelas, O. Hodson, V. Hardmann, M. Handley, J. Bolot, A. Vega-Garcia, S. Fosse-Parisis, "RTP Payload for Redundant Audio Data," RFC 2198, IETF, September 1997.

[163] J. Postel, "User Datagramm Protocol," RFC 768, IETF, August 1980.

[164] J. Postel, "Internet Protocol," RFC 791, IETF, September 1981.

[165] J. Postel, "Internet Control Message Protocol," RFC 792, IETF, September 1981.

[166] B. Quinn, K. Almeroth, "IP Multicast Applications: Challenges and Solutions," RFC 3170, IETF, September 2001.

[167] P. Radoslavov, D. Estrin, R. Govindan, M. Handley, S. Kumar, D. Thaler, "The Multicast Address-Set Claim (MASC) Protocol," RFC 2909, IETF, September 2000.

[168] M. Ramalho, "Intra- and inter-domain multicast routing protocols: A survey and taxonomy," IEEE Communications Surveys & Tutorials, Volume 3 Number 1, 2000.

[169] V. Rayward-Smith, "The computation of nearly minimal Steiner trees in graphs," International Journal of Mathematical Education in Science & Technology, Volume 14, 1983.

[170] Y. Rekhter, T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771, IETF, March 1995.

[171] J. Reynolds, J. Postel, "Assigned Numbers," RFC 1700, IETF, October 1994.

[172] L. Rizzo, "dummynet," http://www.iet.unipi.it/luigi/dummynet/.

[173] L. Rizzo, "A PGM Host Implementation for FreeBSD," http://info.iet.unipi.it/~luigi/pgm.html.

[174] J. Rosenberg, H. Schulzrinne, "Sampling of the Group Membership in RTP," Internet Draft, draft-ietf-avt-rtpsample-01, IETF, May 1999.

[175] G. Rouskas, I. Baldine, "Multicast Routing with End-to-End Delay and Delay Variation Constraints," IEEE Journal on Selected Areas in Communications, Volume 15 Issue 3, April 1997.

[176] L. Sahasrabuddhe, B. Mukherjee, "Multicast Routing Algorithms and Protocols: A Tutorial," IEEE Network, Volume 14 Number 1, January/February 2000.

[177] H. Salama, D. Reeves, Y. Viniotis, "Evaluation of multicast routing algorithms for real-time communication on high-speed networks," IEEE Journal on Selected Areas in Communications, Volume 15 Issue 3, April 1997.

[178] K. Sarac, K. Almeroth, "Multicast Reachability Monitoring Protocol (MRM) End Host Implementation," http://steamboad.cs.ucsb.edu/mrm/.

[179] K. Sarac, K. Almeroth, "Monitoring Reachability in the Global Multicast Infrastructure," Proceedings of International Conference on Network Protocols (ICNP), Osaka, Japan, November 2000.

[180] K. Sarac, K. Almeroth, "Supporting the Need for Inter-Domain Multicast Reachability," Proceedings of Network and Operating System Support for Digital Audio and Video (NOSSDAV '00), Chapel Hill, North Carolina, USA, June 2000.

[181] K. Sarac, K. Almeroth, "Supporting Multicast Deployment Efforts: A Survey of Tools for Multicast Monitoring," Journal of High Speed Networking - Special Issue on Management of Multimedia Networking, March 2001.

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

[183] H. Schulzrinne, "RTP Profile for Audio and Video Conferences with Minimal Control," RFC 1890, IETF, January 1996.

[184] H. Schulzrinne, A. Rao, R. Lanphier, "Real Time Streaming Protocol (RTSP)," RFC 2326, IETF, April 1998.

[185] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," Internet Draft, draft-ietf-avt-rtc-new-04, IETF, December 1999.

[186] H. Schulzrinne, S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control," Internet Draft, draft-ietf-avt-profile-new-10, IETF, August 2001.

[187] R. Sedgewick, "Algorithmen in C," 1. Auflage, Addison Wesley, 1992.

[188] S. Shalunov, B. Teitelbaum, "A One-way Active Measurement Protocol Requirements," Internet Draft, draft-ietf-ippm-owdp-reqs-02, IETF, June 2002.

[189] S. Shenker, C. Partridge, R. Guerin, "Specification of Guaranteed Quality of Service," RFC 2212, IETF, September 1997.

[190] T. Speakman, J. Crowcroft, J. Gemmell, D. Farinacci, S. Lin, D. Leshchiner, M. Luby, T. Montgomery, L. Rizzo, A. Tweedly, N. Bhaskar, R. Edmonstone, R. Sumansekera, L. Vicisano, "PGM Reliable Transport Protocol Specification," RFC 3208, IETF, December 2001.

[191] R. Steinmetz, "Multimedia-Technologie: Grundlagen, Komponenten und Systeme," 3te Auflage, Springer, Berlin, 2000.

[192] A. Swan, D. Bacher, L. Rowe, "rtpmon 1.0a7," University of Berkeley, January 1997. ftp://mm-ftp.cs.berkeley.edu/pub/rtpmon/.

[193] A. Tanenbaum, "Computer Networks," 3rd edition, Prentice Hall PTR, 1996.

[194] D. Thaler, B. Aboba, "Multicast Debugging Handbook," Internet Draft, draft-ietf-mboned-mdh-05, IETF, November 2000.

[195] D. Thaler, M. Handley, D. Estrin, "The Internet Multicast Address Allocation Architecture, "RFC 2908, IETF, September 2000.

[196] H. Uijterwaal, M. Kaeo, "One-Way Metric Applicability Statement," Internet Draft, draft-ietf-ippm-owmetric--as-00, IETF, July 2002.

[197] D. Waitzman, C. Partridge, S. Deering, "Distance Vector Multicast Routing Protocol," RFC 1075, IETF, November 1988.

[198] J. Walz, B. Levine, "A practical multicast monitoring scheme," University of Massachusetts, Computer Science Technical Report 30-2000, June 2000.

[199] B. Wang, J. Hou, "Multicast Routing and its QoS Extension: Problems, Algorithms, and Protocols," IEEE Network, Volume 14 Number 1, January/February 2000.

[200] N. Wang, C. Low, "On Finding Feasible solutions to Group Multicast Routing Problem, "Proceedings of IFIP International Conference of NETWORKING 2000, Paris France, May 2000.

[201] N. Wang, "The Steiner Tree Problem," August 2002. http://www.geocities.com/CollegePark/Residence/9200/research/steiner.htm.

[202] A. Watson, M. Sasse, "Multimedia Conferencing via Multicast: Determining the Quality of Service Required by the End User," Proceedings of International Workshop on Audio-Visual Services over Packet Networks (AVSPN '97), Aberdeen, Scotland, 15-16 September 1997.

[203] A. Watson, M. Sasse, "Measuring Perceived Quality of Speech and Video in Multimedia Conferencing Applications," Proceedings of ACM Multimedia '98, Bristol, England, 12-16 September 1998.

[204] Z. Whang, J. Crowcroft, "Quality-of-Service Routing for Supporting Multimedia Applications," IEEE Journal on Selected Areas in Communications, Volume 14 Numer 7, 1996.

[205] B. Williamson, "Developing IP Multicast Networks," Cisco Press, Indianapolis, USA, 2000.

[206] W. Willinger, M. S. Taqqu, R. Sherman, D. V. Wilson, "Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level," ACM SIGCOMM Computer Communication Review, Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication, Volume 25 Issue 4, October 1995.

[207] P. Winter, "Steiner Problem in Networks: A Survey," Networks, Volume 17, Number 2, 1987.

[208] R. Wittmann, M. Zitterbart, "Multicast Communication - Protocols and Applications," Academic Press, San Diego, California, 2001.

[209] J. Wroclawski, "The Use of RSVP with IETF Integrated Services," RFC 2110, IETF, September 1997.

[210] J. Wroclawski, "Specification of the Controlled-Load Network Element Service," RFC 2211, IETF, September 1997.

[211] S. Yan, M. Faloutsos, A. Banerjea, "QoS-aware Multicast Routing for the Internet: The Design and Evaluation of QoSMIC," IEEE/ACM Transactions on Networking (TON), Volume 10 Issue 1, February 2002.

[212] "40, 32, 24, 16 kbit/s adaptive differential pulse code modulation (ADPCM)," ITU-T Recommendation G.726, Telecommunication Standardizzation Sector of ITU, 1990.

[213] "Cisco IP/TV Administration and Configuration Guide," Cisco, Inc. http://www.cisco.com/univercd/cc/td/doc/product/webscale/iptv/iptv34/adm_gd/index.htm.

[214] "MPEG1 - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s," International Organisation for Standardisation, ISO/IEC 11172-1-3, 1993.

[215] "MPEG2 - Generic coding of moving pictures and associated audio information," International Organisation for Standardisation, ISO/IEC 13818-1-3, 2000.

[216] "Multicast Routing Monitor," Cisco, Inc. http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t5/mrm.pdf.

[217] "Pulse code modulation (PCM) of voice frequencies," ITU-T Recommendation G.711, Telecommunication Standardizzation Sector of ITU, 1988.

[218] "Video codec for audiovisual services at p x 64 kbit/s," ITU-T Recommendation H.261, Telecommunication Standardizzation Sector of ITU, 1990.

[219] "Video coding for low bit rate communication," ITU-T Recommendation H.263, Telecommunication Standardizzation Sector of ITU, 1993.

List of Figures

List of Tables

Index

Acronyms

AAP	address allocation protocol
AS	autonomous system
ASIC	application specific integrated circuit
ATM	asyncronous transfer mode
BGP	border gateway protocol
BHN	Bayrisches Hochschulnetz (Bavarian university network)
BSR	bootstrap router
CBQ	class-based queuing
CDV	cell delay variation
CoS	class of service
DiffServ	differentiated services
DNDF	dummynet with distribution function
DR	designated router
DVMRP	distance vector multicast routing protocol
FDDI	fiber distributed data interface
FIFO	first-in, first-out
G-WiN	Deutsches Wissenschaftsnetz (German research network)
GPS	global positioning system
IANA	internet assigned numbers authority
ICMP	internet control message protocol
IETF	internet engineering task force
IGMP	internet group management protocol
IntServ	integrated services
IP	internet protocol
IPDV	IP delay variation
IPPM WG	IP performance metrics working group
ISP	Internet Service Provider
ITU	international telecommunication union
LAN	local area network

MAAS	multicast address allocation server
MADCAP	multicast address dynamic client allocation protocol
MASC	multicast address-set claim
MBone	multicast backbone
MBGP	multiprotocol extensions for BGP
MJPEG	motion joint photographic experts group
MPEG	moving pictures experts group
MQM	multicast quality monitor
MRM	Multicast Reachability Monitor
MRT	multicast routing tool
MSDP	multicast source discovery protocol
MTU	maximum transmission unit
MZAP	multicast-scope zone announcement protocol
NOC	network operation center
NTP	network time protocol
OSPF	open shortest path first
OWD	one-way delay
PCM	pulse code modulation
PGM	pragmatic general multicast protocol
PHB	per hop behavior
PIM	protocol independent multicast
PIM-DM	PIM dense-mode
PIM-SM	PIM sparse-mode
QoS	quality of service
RED	random early discard
RFC	request for comments
RP	rendezvous-point
RPT	RP-tree
RPF	reverse path forwarding
RSVP	resource reservation protocol
RTCP	RTP control protocol
RTP	real-time transport protocol

RTSP	real-time streaming protocol
RTT	round-trip time
SAP	session announcement protocol
SDH	synchronous digital hierarchy
SDP	session description protocol
SLA	service level agreement
SNMP	simple network management protocol
SPG	Steiner tree problem in graphs
SPT	shortest path tree
SSM	source specific multicast
TCP	transmission control protocol
TKBRZL	Telekonferenz der Bayrischen Rechenzrentrumsleiter
TOS	type of service
TTL	time to live
UDP	user datagram protocol
UML	uniform modeling language
UTC	universal time
VoD	video on demand
VoIP	voice over IP
WWW	world wide web
Xcast	explicit multicast

Überwachung von Multicast-Netzwerken für zeitsynchrone Kommunikation

Inhaltsverzeichnis

Kurzfassung

Das Internet wird im zunehmenden Maße für die Übertragung von multimedialem Inhalt genutzt. Echtzeitanwendungen wie Videokonferenzen und TV-Broadcasts stehen dabei im Vordergrund. Die Entwicklung und Implementierung von IP Multicast ist die Grundlage für die sinnvolle Nutzung dieser neuen Dienste. Weiterhin wird eine minimale Dienstgüte für die Übertragung dieser hochqualitativen Multimedia-Daten vorausgesetzt.

Leider ist IP Multicast noch immer eine sehr komplizierte und fehleranfällige Technologie. Nur wenige ISPs sind in der Lage, diese Techniken korrekt und stabil einzusetzen. Auch sind bisher wenige oder keine Mechanismen in den großen Backbone-Netzen implementiert, die eine bessere Dienstgüte als die bei IP-Netzen übliche best effort Übertragung anbieten oder sogar garantieren.

Eine der wichtigsten Aufgaben ist es also, Werkzeuge zur Überwachung der Multicast-Netze zu entwickeln. Ziel dieser Tools ist neben der Ermittlung der Verfügbarkeit die Messung der aktuell nutzbaren Dienstgüte.

Im Rahmen dieser Arbeit wird ein neues Konzept beschrieben, das die Messung der Verfügbarkeit und der Dienstgüte in verteilten Multicast-Netzen ermöglicht. Ein wesentliches Augenmerk bei der Spezifikation ist die Skalierbarkeit der Methode. Um diese sicher zu stellen, werden neue Mechanismen entworfen, die die Messungen auch in sehr großen und verteilten Umgebungen ermöglichen.

Es hat sich herausgestellt, daß die Meßmethoden sowie die ideale Verteilung der Meßstationen eng gekoppelt sind mit dem konkreten Ziel, der Sicherstellung der Funktionalität und der Qualität bestimmter Multicast-Dienste.

Auf diesen Aspekt wird im Rahmen der Arbeit eingegangen. Ein neues Modellkonzept wird beschrieben, welches die Integration von Informationen über die zur Verfügung stehende Netzwerk-Infrastruktur sowie über die wichtigsten Multicast-Dienste erlaubt. Basierend auf den modellierten Daten kann ein ebenfalls implementierter Routing-Algorithmus die für die aktiven bzw. geplanten Multicast-Dienste benötigten Teile des Netzes identifizieren sowie Informationen bereitstellen, welche konkreten Messungen für die optimale Bestimmung der möglichen Dienstgüte durchzuführen sind.

Zusammenfassend kann man sagen, daß ein Rahmenwerk definiert und gestestet wird, welches die genannten Zielstellungen erfüllt. Anhand von prototypischen Implementierungen wird die Funktionalität im Labor und an realen Netzen überprüft.

Einleitung

Im Internet ist IP Multicast eine der Technologien, die die zukünftige Entwicklung moderner Multimediaanwendungen am entscheidendsten beeinflußt. Die Nutzung des Internets hat bisher drei wesentliche Zyklen erlebt. Wenn das Netz zu Beginn nur von Experten bedient werden konnte, haben sich schon sehr bald Anwendungen wie die elektronische Post (EMail) und Diskussionsforen wie die NetNews als Hauptanwendung etabliert. Eine neue Phase in der Geschichte des Internet begann mit der Entwicklung des World Wide Web (WWW). Mit der immer einfacheren Bedienung durch graphische Bedienoberflächen und den umfassenden Nutzungsmöglichkeiten des Hypertext-orientierten WWW startete der Durchbruch des Internet in der privaten Heimanwendung.

Der nächste Schritt ist bereits im Kommen. Dank neuer Technologien, die die verfügbare Bandbreite sowohl der Backbone-Netze der großen ISPs (Internet Service Provider) als auch der Anschlüssen im Heimbereich um ein vielfaches steigern, wird die Nutzung des Internet auch für multimedialen Inhalt wie Radio- oder Fernsehübertragungen in Echtzeit interessant. Genau hier ist der Schnittpunkt mit der Entwicklung von IP Multicast zu sehen, da diese Anwendungen immer eine große Anzahl von Teilnehmern gleichzeitig erreichen wollen, aber Ressourcen im Netz und bei den Servern gespart werden sollen.

Die Anfänge von IP Multicast begannen mit dem Aufbau des MBone, dem Multicast-Backbone [73], etwa 1992. Den Grundstein für diese Entwicklungen legte Steve Deering mit dem Host-Modell für IP Multicast [52], [55], welches detailliert in seiner Ph.D. Thesis [53] beschrieben wurde. Im Gegensatz zu den ersten Tagen der Einführung von Multicast, gibt es heute deutlich ausgefeiltere Protokolle und stabilere Implementierungen.

Vor der Entwicklung von Multicast hatten Entwickler von Applikationen für das Internet die Wahl zwischen Unicast- und Broadcast-Übertragungen. Unter einer Unicast-Verbindung versteht man den Transport von Paketen über das Netz von einem Sender zu exakt einem Empfänger. Das benutzte IP-Protokoll ist in [164] beschrieben. Gerade in lokalen Netzen findet man oft die Anforderung, mehrere bzw. alle anderen Stationen gleichzeitig anzusprechen. Für die Lösung dieses Problems wurde die Broadcast-Übertragung eingeführt. Ein Paket von einem einzelnen Sender wird hier allen Teilnehmern des Netzes zugestellt.

Für einzelne Anwendungen gibt es einen entsprechenden Bedarf auch über die Grenzen lokaler Netz hinweg. Schon 1978 wurde ein Vorschlag zur Lösung dieser Aufgabe gemacht [48], der später auch beim Entwurf von IP Multicast Pate stand.

Die Vorteile der Unicast-Übertragung liegen klar auf der Hand. Die meisten Applikationen basieren auf einer direkten Host-zu-Host-Kommunikation. Sollen allerdings mehrere Ziele gleichzeitig angesprochen werden, so sind entsprechend viele parallele Verbindungen aufzubauen. Das führt zu einem potentiellen Flaschenhals in der Leistungsfähigkeit des Serversystems bzw. zu einer Überlastung des Netzanschlusses dieses Systems.

Sollen beispielsweise n parallele Videoströme mit je 5 Mbps verschickt werden, so läßt ein 100 Mbps-Anschluß theoretisch maximal 20 Ströme zu. Wird das Video via Broadcast verschickt, ist das Serversystem nicht mehr überlastet, es kann jedoch zu Engpässen im Netz führen, da der Datenstrom an alle Systeme verteilt wird und nicht nur an die, die ihn auch tatsächlich empfangen wollen. Eine Überlastsituation kann so auch an völlig unbeteiligten Endsystemen auftreten, die die nicht erwarteten Pakete nach einer Analyse verwerfen müssen.

Diese Überlegungen führten zur Entwicklung von IP Multicast. Das Prinzip dieser Übertragung ist, Datenpakete nur einmal in das Netz zu verschicken und dem Netz die Aufgabe zu überlassen, die Pakete an den richtigen Stellen zu vervielfältigen und nur den Stationen zuzustellen, die am Empfang der Daten interessiert sind.

Man unterscheidet grundsätzlich Protokolle und Technologien für das Intra-Domain-Multicast-Routing und das Inter-Domain-Multicast-Routing [92], [112]. Über das Protokoll IGMP (Internet Group Management Protocol, [51]) teilen Endsysteme ihrem direkt angeschlossenen Router mit, daß sie am Empfang von Paketen für eine spezielle Multicast-Gruppe interessiert sind. Routing-Mechanismen sorgen dann für den Transport von Multicast-Paketen zu interessierten Empfängern.

Die Multicast Routing-Protokolle [34] unterteilt man in zwei wesentliche Klassen. Dense-mode-Protokolle arbeiten nach dem "push"-Prinzip, d.h. Daten werden an alle Systeme verteilt, es sei denn, es wird eine Mitteilung verschickt, daß die Übertragung nicht erwünscht wird (z.B. weil keine Empfänger für die Daten aktiv sind). Sparse-mode-Protokolle hingegen operieren nach dem "pull"-Prinzip. Multicast-Datenströme werden durch ein explizites Anforderungssystem bestellt. Protokolle beider Klassen unterscheiden sich durch ihre Skalierbarkeit und Komplexität.

In den Anfängen von IP Multicast wurde ein logisches Netz über dem Internet aufgebaut, welches nur der Übertragung von IP Multicast diente, das MBone. Über Tunnelkonstrukte wurden einzelne Multicast-fähige Netze miteinander verbunden. Diese alten Strukturen werden heute durch ein natives Multicast-Routing ersetzt [8]. Noch offene Fragen nach der Sicherheit und Skalierbarkeit werden von aktuellen Entwicklungen wie IGMPv3 [38] oder SSM (Source Specific Multicast, [111], [24], [25]) beantwortet.

IP Multicast wird bereits seit einigen Jahren intensiv für verschiedene Anwendungen eingesetzt. An vorderster Stelle stehen Multimedia-Applikationen [121], [127] wie Video-Konferenzen, wie z.B. die Telekonferenz der bayrischen Rechenzentrumsleiter (Projekt TKBRZL, [107]), TV-Broadcasts, z.B. im Rahmen des Projekts Uni-TV [62], die Übermittlung von Wertpapierkursen [136] oder Spezialanwendungen wie die Zeitsynchronisation via NTP (Network Time Protocol, [146]).

Aufgrund der relativ hohen Komplexität, unvollständiger bzw. sogar fehlerhaften Implementierungen ist IP Multicast für die Betreiber der Backbone-Netze, die ISPs, sehr aufwendig zu betreiben [61]. Es gibt noch immer nur wenige verfügbare Mitarbeiter mit entsprechendem Know-How. Von den Nutzern von IP Multicast, also den Kunden der ISPs, wird aber gerade im Multicast-Umfeld eine besonders hohe Qualität erwartet, da die primären Anwendungen Multimedia-Applikationen mit hohen Ansprüchen an die zur Verfügung gestellte Dienstgüte des Netzes, vor allem aber an seine Echtzeitfähigkeit, sind. Entstanden ist ein Henne-Ei-Problem. Die ISPs sind nicht bereit, sehr viel Geld in die Ausstattung ihrer Netze und die Ausbildung ihrer Mitarbeiter zu investieren, bevor es nicht eine Reihe von zahlenden

Kunden gibt. Die Kunden hingegen wollen die bestehenden Netze aufgrund der mangelnden Dienstgüte, dazu gehört auch die reine Funktionalität des IP Multicast-Routings, nicht für wichtige IP Multicast-Applikationen forcieren. Darunter leidet auch die Entwicklung neuer Dienste.

Die Lösung dieser Fragestellung sieht man heute in der Entwicklung von automatisierten Möglichkeiten, Fehler und Problemstellen schnell zu finden und einzugrenzen. Ziel sind vollautomatische Funktionstests, sowie die Analyse der vorhandenen Dienstgüte im Netz bzw. sogar die Vorhersage der zu erwartenden Dienstqualität [67]. Seit 1999 wird an entsprechenden Tools gearbeitet [6]. Zu nennen sind hier vor allem MHealth [131], die Kombination des mtrace-Tool [80] mit einer graphischen Oberfläche, der Multicast Reachability Monitor (MRM, [11]) und ein Projekt des NLANR (National Laboratory for Applied Network Research), dem Multicast Beacon [41].

Das Problem aller dieser Ansätze ist, daß sie leider unvollständig sind, d.h. nicht alle relevanten Dienstgüteparameter werden untersucht. Zumeist sind nur einfache Erreichbarkeitstests möglich und es stehen nur eingeschränkte Möglichkeiten zur Lokalisierung von Problemstellen zur Verfügung.

In der vorliegenden Arbeit wird ein neuer Ansatz beschrieben, der aufgrund einer komplett neuartigen Problemanalyse versucht, die Unvollständigkeit bisheriger Implementierungen zu vermeiden. Die primäre Idee ist es, ausgehend von den wichtigsten Anwendungen im Netz, die vorhandene Multicast-Infrastruktur zu testen. Dabei werden Prüfungen der Funktionalität, d.h. der Erreichbarkeit im Netz sowie Dienstgütemessungen, durchgeführt. Ziel ist es auch, soweit es anhand vorhandener Informationen und Meßdaten möglich ist, Vorhersagen über die zu erwartende Qualität geplanter Anwendungen zu erstellen.

Die Arbeit ist folgendermaßen aufgebaut: Zuerst wird eine Einführung in die Grundlagen von IP Multicast gegeben, eine Analyse der potentiellen Probleme durchgeführt und eine Aufstellung der Ziele der Arbeit gegeben. Dazu beschreibt Kapitel 2 "Multicast Infrastructure" die aktuell eingesetzten Multicast-Routing-Protokolle und Strategien für Intra-Domain- und Inter-Domain-Multicast-Routing. Ebenfalls werden hier bekannte Multicast-Netze analysiert und beschrieben. Beispiele sind das Campus-Netz der Universität Erlangen-Nürnberg, das Bayrische Hochschulnetz (BHN), das Deutsche Forschungsnetz (G-WiN) und das Multicast-Routing im globalen Internet. Ein Überblick über typische Applikationen, wie Audio- und Videotools, Collaboration-Anwendungen u.ä. sowie über die typischen Dienstformen in Multicast-Netzen, die Broadcasts und die Konferenzen rundet das Kapitel ab.

In Kapitel 3 "Quality of Service in Multicast Networks" werden die wesentlichen Dienstgüteparameter für Datenübertragungen über das Internet vorgestellt. Dies sind im einzelnen die Erreichbarkeit, die Paketverlustrate, die absolute Verzögerung, das Delay und die Varianz der Verzögerung, der Jitter. Mehrere Arbeitsgruppen beschäftigen sich mit der Möglichkeit, diese Parameter zu messen, ganz voran die IPPM WG (IP Performance Measurement Working Group) der IETF (Internet Research Task Force). Dazu wurden für IP Unicast Metriken und Meßverfahren entwickelt, um die Dienstgüte einer Verbindung zu bestimmen. Für IP Multicast gibt es erste Ansätze und Ideen, die an dieser Stelle genauer untersucht werden sollen. Mit einem Überblick über Verfahren zur Sicherung und Garantie von

Dienstgüteeigenschaften von Übertragungen über das Internet soll das Kapitel beschlossen werden. Genannt seien hier beispielhaft die "Integrated Services Architecture", die "Differentiated Services Architecture" (beides sind Arbeitsgruppen der IETF), sogenannte layered Multicast-Übertragungen oder die Entwicklung von Protokollen für die gesicherte Übertragung von Multicast-Paketen.

Im Rahmen von Kapitel 4 "QoS Requirements of IP Multicast Applications" werden Meßmethoden und deren Ergebnisse vorgestellt, die der Analyse des Verkehrsverhaltens von multimedialen Multicast-Applikationen dienen. Dazu wird ein Überblick über mögliche Testverfahren gegeben. Man unterscheidet im wesentlichen zwischen objektiven und subjektiven Messungen sowie zwischen Messungen in einer Laborumgebung und solchen im realen Netz. Mangels geeigneter Instrumente für die Analyse von Multicast-Applikationen im Labor, speziell für die Beeinflussung der Dienstgüteeigenschaften eines Netzes, wurde ein IP-basiertes Impairmenttool entwickelt.

Die Präsentation des neuen Ansatzes zur Messung und Analyse der Dienstgüteparamter in einem Multicast-Netz ist in drei Schwerpunktthemen gegliedert. In Kapitel 5 "Modeling IP Multicast Networks and Services" wird in objektorientierter Ansatz zur Darstellung von IP-Multicast-Netzen inklusive der darüber genutzten Multicast-Dienste vorgerstellt. Das Modell, spezifiziert in UML (Uniform Modeling Language), erlaubt das Einbinden von statischen als auch von dynamischen, d.h. gemessenen Daten und Informationen einzelner Netzkomponenten. Der Aufbau des Objekt-Modells folgt dem TCP/IP-Schichtenmodell. Das Modell erlaubt durch integrierte Routing-Algorithmen die Berechnung bester Wege durch das Netz basierend auf den Eigenschaften des Netzes und den Anforderungen der genutzten Dienste. Nutzungsszenarien werden ebenso präsentiert wie Eigenschaften der prototypischen Implementierung in JAVA.

In Kapitel 6 "Definition of a Metric for Multicast Services" wird eine Metrik vorgestellt, die es erlauben soll, die Gegebenheiten in einem Multicast-Netz mit den Anforderungen von Applikationen zu vergleichen. Ein weiteres Ziel einer solchen Metrik ist die Erarbeitung von SLAs (Service Level Agreements), die Kunden mit ihren ISPs abschließen können, die die Minimalleistungen des Netzes in Bezug auf die zur Verfügung gestellte Dienstgüte repräsentieren. Potentielle Parameter der Metrik und eine prinzipielle Berechnungsgrundlage runden dieses Kapitel ab.

Kapitel 7 "Multicast Quality Monitor (MQM)" beschreibt ein neu entwickeltes Programm zur Messung von Dienstgüteeigenschaften in einem Multicast-Netz. Basierend auf den Erfahrungen aus der Analyse der bereits existierenden Ansätze und dem Modell aus Section 5 werden Meßverfahren entworfen und implementiert, die es erlauben, auch komplexe Multicast-Netze zu überwachen. Neben dem Aufbau und den grundlegenden Prinzipien des Tools werden die einzelnen Meßverfahren, gegliedert nach Verfügbarkeitsmessungen und Dienstgütemessungen, detailliert beschrieben. Einige dieser Verfahren wurden bereits in anderen Entwicklungen vorgestellt und genutzt, einige, wie z.B. die Bestimmung des One-Way-Delays für IP Multicast, sind derzeit einzigartig. Einem neuartigen Entwurf folgt auch die Kommunikation und Datenhaltung zwischen den Meßinstrumenten.

Abschließende Betrachtungen über den Status der Arbeit und ein Ausblick auf weitere Forschungsschwerpunkte sind in Kapitel "Zusammenfassung" zusammengestellt.

Zusammenfassung

Die Nutzungsszenarien des Internets haben sich in den letzten Jahren deutlich verändert. Multimediale Dienste prägen die Landschaft moderner Netzwerk-Applikationen. In diesem Rahmen nimmt auch die Bedeutung von IP Multicast stark zu. Typische Anwendungen in einem Multicast-basierten Umfeld sind etwa Videokonferenzen und Broadcast-Übertragungen von multimedialem Inhalt. Sehr bekannt sind schon heute Radiosender, die ihren Ausstrahlungsbereich durch die Nutzung des Internets deutlich vergrößern wollen. Ein zukunftsträchtiger Bereich sind auch die diversen, teilweise weltweit ausgestrahlten Fernsehsendungen.

Alle diese Anwendungen erwarten einen zuverlässigen und, aufgrund der Anforderungen durch die Übertragung von Multimediadaten vorausgesetzten, hochqualitativen Übertragungsdienst. Die Aufgabe der Service-Provider, der Hardware-Hersteller und der Entwickler neuer Protokolle und Mechanismen sind aufgefordert, Netzwerke bereitzustellen, die diesen Anforderungen genügen.

Die Betrachtung heutiger Multicast-Netzwerke macht jedoch deutlich, daß es aufgrund der hohen Komplexität der Protokolle und der Konfiguration der Netzkomponenten hier deutliche Engpässe gibt. Ein weiterer Kritikpunkt ist die fehlende bzw. mangelhafte Bereitstellung einer entsprechenden Dienstgüte für ausgewählte Applikationen.

Ursprünglich war das Internet nur für eine "best effort" Übertragung vorgesehen, d.h. Daten wurden so gut wie möglich transportiert. Dieser Mechanismus liegt auch heute noch allen IP basierten Netzen zugrunde. Neue Ansätze und Entwicklungen erlauben es, Daten im Netzwerk zu priorisieren und ihnen eine bevorzugte Behandlung zukommen zu lassen. Andere Konzepte versuchen durch Ressourcenreservierungen eine minimale Ende-zu-Ende Dienstgüte zu garantieren. Die Implementierung dieser Ideen geht leider nur sehr langsam voran. Oftmals werden auch die versprochenen Dienstgütemerkmale nicht erreicht bzw. eingehalten. Analysewerkzeuge für die Untersuchung von Muliticast-Netzwerken werden benötigt.

Ziel dieser Arbeit war es, ein Rahmenwerk zu definieren, welches es erlaubt, Analysen von Multicast-Netzen durchzuführen. Zum einen soll die Verfügbarkeit der Multicast-Netze bzw. grundsätzlich die Erreichbarkeit von einzelnen Systemen via IP Multicast ermittelt werden. Die Ergebnisse müssen mit Tests des normalen Netzes, d.h. der unicast IP-Konnektivität verglichen werden, um Unterscheidungen von allgemeinen Netzstörungen und solchen im Multicast-Routing zu erlauben. Zum anderen sind Messungen der Übertragungsqualität im Netz zwingend notwendig. Die Ergebnisse dieser Tests erlauben es, die aktuelle Dienstgüte im Netz zu bestimmen und auch Vorhersagen zu treffen, die das Verhalten von erwarteten Übertragungen vorhersagen.

Zu Beginn wurden typische Multicast-Netze ausgewertet, um potentielle Problemstellen zu analysieren. Weiterhin wurden existierende Analysemethoden und Tools untersucht, in wie weit diese bereits die genannten Ziele erreichen. Es stellte sich heraus, daß keines der Werkzeuge

einen umfassenden Überblick über die Verfügbarkeit und die Qualität von IP Multicast-Netzen ermöglicht. Auch haben alle Ansätze Probleme mit der Skalierbarkeit. Sehr große Netze können nicht bzw. nur sehr eingeschränkt untersucht werden.

Im Rahmen der Arbeit wurde ein neuer Ansatz vorgestellt, diese Messungen durchzuführen. Unter dem System mit dem Namen Multicast Quality Monitor (MQM) verbirgt sich eine Meßumgebung, die es erlaubt die folgenden Untersuchungen an IP Multicast-Netzen durchzuführen.

Für den Test von Erreichbarkeit und Verfügbarkeit wurde ein neuer Multicast-Ping-Mechanismus eingeführt, der es erlaubt, die Erreichbarkeit von einzelnen Meßstationen im Multicast-Netz zu untersuchen. Die grundsätzliche Idee ist es, periodisch Nachrichten zu verschicken, die von allen Empfängern entsprechend beantwortet werden. Anhand der empfangenen Pakete läßt sich so ein Erreichbarkeitsgraph erstellen. Die perodische Ausführung der Aktionen ist zwingend notwendig, um die Einträge für das Multicast-Routing in den Netzwerk-Komponenten aufzufrischen. Die Verfügbarkeit ergibt sich aus den einzelnen Erreichbarkeitstests über eine Zeitspanne.

Für die Untersuchung der Dienstgüte in einem Multicast-Netzwerk werden zwei verschiedene Mechanismen genutzt. Die Laufzeit, also das Delay, kann mit demselben Methoden erfolgen, die für die Erreichbarkeitstests eingesetzt werden. Dazu erhalten die einzelnen Datenpakete, welche im vorgeschlagenen MQM Format kodiert sind, zusätzlich Zeitinformationen. Zeitstempel werden beim Verschicken der MQM Ping-Request-Nachricht, bei der entsprechenden Antwort und beim Empfang dieser Antwort ermittelt. Anhand dieser Daten kann das One-Way-Delay und die Round-Trip-Time berechnet werden. Die Berechnung des One-Way-Delays basiert dabei natürlich auf Zeitinformationen zweier verschiedener Rechner. Genaue Werte können nur ermittelt werden, wenn die Uhren sehr präzise synchronisiert wurden. Typischerweise werden GPS-Empfänger für diese Aufgabe genutzt.

Der zweite Teil der Messungen wird anhand von kontinuierlichen Datenströmen durchgeführt. Dabei werden RTP kodierte Pakete benutzt, die für die Messungen wichtige Informationen wie Sequenznummern und Zeitstempel enthalten. Anhand dieser Daten lassen sich die Varianz des Delays, der Jitter, die Paketverlustrate sowie die Raten der vertauschten bzw. duplizierten Pakete berechnen. Als Quellen für die zu analysierenden RTP-Ströme kommen aktive Multimedia-Übertragungen in Frage. Der Vorteil dieser Meßmethode ist, daß das Netz nicht durch zusätzlichen Datenverkehr, welcher nur für die Messungen benötigt wird, belastet wird. Ist kein solcher Dienst bzw. keiner mit den entsprechenden Merkmalen wie z.B. der gewünschten Datenrate aktiv, so erlaubt es der Multicast Quality Monitor, Datenströme zu simulieren. Der Empfang und die Analyse der Daten bleibt dabei unverändert.

Eine der wichtigsten Fragestellungen bei der Entwicklung des MQM war die Skalierbarkeit. Alle einzelnen Mechanismen wurden im Hinblick auf ihr Verhalten untersucht, wenn sehr große Netze analysiert werden sollen.

Der MQM Ping Mechanismus ist so aufgebaut, daß schon zwei Ping-Nachrichten von verteilten Meßstationen für eine komplette Analyse des Netzes ausreichen. Anhand der gewonnenen Daten aus den Antworten auf die beiden Anfragen, läßt sich ein vollständiger Erreichbarkeitsgraph inklusive ermittelter Delay-Werte berechnen.

Die Analyse der Übertragungsqualität durch das Multicast-Netz anhand von RTP-Strömen kann, bzw. soll, basierend auf aktiven Multicast-Anwendungen stattfinden. Außerdem erlaubt der spezifizierte Beacon-Mechanismus dynamisch aktivierte Tests, d.h. alle RTP-basierten Messungen lassen sich zentral steuern, so daß keine unnötigen, lang laufenden Sende- und Empfangsprozesse vorausgesetzt werden müssen. Der Einfluß auf das Netz und damit die aktiven Dienste kann durch den Beacon-Mechnismus minimiert werden.

Generell ist es unmöglich, alle potentiellen Verbindungen im Internet auf Verfügbarkeit und Qualität hin zu untersuchen. Ein Lösungsansatz wurde mit dem Modell für Multicast-Netze und Multicast-Dienste vorgeschlagen. Dieses Modell erlaubt es, die Netzwerkinfrastruktur von IP Multicast-Netzen detailliert zu beschreiben. Anhand der modellierten Informationen und Meßdaten bzw. Simulationsreihen können optimale Wege zwischen verschiedenen Endsystemen und die mögliche Übertragungsqualität berechnet werden. Das Modell erlaubt weiterhin die Einbindung von Informationen über die genutzten Multicast-Dienste. Gemeinsam mit diesen Daten kann die Analyse des Netzes deutlich vollständiger erfolgen. Vorschläge für die Platzierung der Meßstationen sind ebenso möglich wie die Empfehlung geeigneter Meßverfahren. Die Anzahl von Einzeluntersuchungen kann hierdurch deutlich reduziert werden bei gleichzeitiger Erhöhung des Informationsgehaltes.

Zusammenfassend läßt sich also sagen, daß erfolgreich ein Rahmenwerk spezifiziert wurde, welches umfangreiche Analysen des Verkehrsverhaltens von IP Multicast-Netzen erlaubt. Alle typischen Dienstgütemerkmale moderner Multimedia-Applikationen wurden integriert, die Skalierbarkeit des Systems in hohem Maße gesichert und Voraussetzungen für Erweiterungen für zukünftige, derzeit unbekannte Anforderungen geschaffen.

Ein Ausblick auf potentielle Verbesserungen soll die Arbeit abschließen. Wichtig für den praktischen Einsatz ist eine breite Anwendung des Systems in realen Netzen und entsprechend geeignete Zugriffsmöglichkeiten für die Nutznießer der Daten, die Endnutzer und die Netzwerkadministratoren. Um das zu erreichen, sind die prototypischen Implementierungen zu erweitern und es sind Abfrage- und Präsentationselemente hinzuzufügen. Eine Standardisierung der vorgestellten Verfahren und Methoden könnte dem hier spezifizierten Verfahren zu breiterer Nutzung verhelfen.

Curriculum Vitae

Personal Details

- born in Dresden, Germany on December 2, 1971
- Nationality: German

School Education

1978-1988	Elementary School (Oberschule), Dresden, Germany
1988-1990	Secondary School (Erweiterte Oberschule), Dresden, Germany
1990	Abitur

Military Service

1996-1997	Zivildienst, Department for Obstetrics and Gynecology
	University of Erlangen-Nuremberg, Germany

University Education

1990-1998	Study of Computer Science, University of Erlangen-Nuremberg, Germany
1998	MS in Computer Science (Dipl. Inf.), Department for Computer Science,
	University of Erlangen-Nuremberg, Germany
	Thesis Title: Monitoring of ATM Networks (Netzmonitoring auf ATM-Ebene)
	Advisors: Dr. P. Holleczek, Prof. F. Hofmann
1999-2003	Ph.D. studies of Computer Science, University of Erlangen-Nuremberg, Germany

Employment History

since 1998	Researcher (wiss. Ang.), Regional Computing Center in Erlangen (RRZE),
	University of Erlangen-Nuremberg, Germany

Affiliations / Memberships

since 2000	Member, MBONE Deployment Working Group (mboned), IETF
since 2001	Member, Multicast & Anycast Group Membership Working Group (magma), IETF
since 2001	Member, ACM (Association for Computing Machinery)
since 2002	Member, IEEE (Institute of Electrical and Electronics Engineers, Inc.)
since 2002	Member, IEEE Computer Society