KUNGL TEKNISKA HÖGSKOLAN
Royal Institute of Technology

How to meet the user requirements of room-based videoconferencing

Tobias Öbrink

Department of Teleinformatics

KUNGL TEKNISKA HÖGSKOLAN
Royal Institute of Technology

How to meet the user requirements of room-based videoconferencing

Tobias Öbrink

Distributed Multimedia Applications to support human communication and collaboration project at the Royal Institute of Technology

December 1999

TRITA–IT/R–99:05
ISSN 1103–534X
ISRN KTH/IT/R—99/05—SE

Department of Teleinformatics

Abstract

Videoconferencing is one of the most demanding services that a future network infrastructure may have to support. This is because of the combination of a need for interactivity and a high bandwidth demand. Videoconferencing might not demand as much interactivity as virtual reality or as much bandwidth as a hard disk backup, but still enough to make any network analyst pull his hair out by the roots trying to support both at the same time. The same dificulties applies inside the computer, where the same amount of data have to be moved multiple times between different devices within the same overall time constraints as the network. Thus, videoconferencing pose quite a challenge for both network and computer designers and in this report I try to push the limits of both.

The end users of a videoconferencing system are humans and the system should be designed to help the users to conduct a meeting where the participants are distributed in space. Transfer effects and association play an important role in the users' decision to adopt a new medium, so for a videoconferencing system to succeed it should provide an audio and video quality comparable to that of other services offered in similar environments.

The practical implications of the theories presented in this report is demonstrated in the design and implementation of a room-based videoconferencing system using standard computers interconnected by an IP-based network that offer better than broadcast TV resolution and still maintain a good interactivity. The system is based on consumer grade DV equipment and IEEE 1394 firewire commonly available in modern computers. I also show different ways to deal with multipoint scaling issues and multiprogramming overhead by analyzing the end-system of a room-to-room videoconferencing site.

Tests with the prototype seems to support earlier observations that; even if network bitrates increases rapidly, the computer design of today has problems to catch up. The designer of systems providing high-bandwidth, networked, real-time, interactive multimedia services like videoconferencing has to be aware of this trend.

Acknowledgements

First of all I would like to thank my wife, Analyn, for having patience with me during the last hectic months. Thanks also to my advisor Prof. Björn Pehrson for his support and to my secondary advisor and lab manager Fredrik Orava for not asking too many questions and protecting me from being assigned too much department duties. Thanks to Prof. Gerald Maguire, Prof. Larry Rowe, Prof. Dale Harris and Prof. Gunnar Karlsson for some interesting discussions, worthwhile input and pointers to good material. Thanks to Christian van den Branden Lambrecht for showing me DCTune. I would also like to thank Akimichi Ogawa for sharing source code and showing interest in my prototype implementation. And finally a big thanks to the guys in the lab: Jiang Wu, in the Bay Jing corner, for being a good friend. Jon-Olov Vatn for being the down-to-earth stabiliser and base station. Iyad Al-Khatib, in the Bay Ruth corner, for being the opposite of J-O. Yannis and Charis, in the greek corner, for being willing movie stars - chronnia polla on you. Enrico Pelletta, in the other half of the mafia corner, for being confused about everything. Ulf Bilting, in his own corner, for his boule excellence and for introducing me to the MICE et. al. culture that inspired this work.

Table of Contents

1. Introduction

For the last few years we have seen a trend where the transmission capacity of optical fibers and networking equipment is increasing faster (proportionally) than the computing power of processors. As a consequence the bandwidth available to the average user will increase faster than the commonly affordable computing power. Therefore it is likely that future telecommunication- and computer-systems will be optimized to save processing power in network nodes and hosts rather than link transmission capacity. How will such a future telecommunication- and computer-systems look like then?

One of the visions of the future is that of a global network containing very powerful (and expensive) servers serving a multitude of more or less intelligent client terminals over high bandwidth fiberoptic and wireless connections. Another vision is that of a multicast-supporting network with distributed applications migrating through the network to serve the user at the current location(s) of the user. These two visions are not necessarily contradictory and we will probably see a blend of both within most networks.

Along with the increase in network capacity we will naturally se a multitude of more or less new services being offered. The most common prediction that one can find in literature debating future services and their effect on the networks is that there will be a lot of video traffic. There are two reasons why this prediction is quite safe: First, because it takes a lot of bits to represent image data, not to say a rapid succession of images so even a relatively small number of video sessions will generate a lot of data. Secondly, because there are a lot of different networked video-based services that people have invented over the years, ranging from video on demand to media spaces, that are waiting for enabling technology.

The network requirements of those video-based services are quite different and I will not try to cover all of them. Instead I will concentrate on the service putting the highest demand on the network - one type of interactive video service called audio-video conferencing, or videoconferencing for short.

2. What's videoconferencing?

There are a lot of different audio- and video-based computerized communication tools and in many papers the term *videoconference* can stand for almost anything mediating audio and video. To avoid confusion and to show the relation between different groups of audio- and video-based computerized communication services I present a list of definitions that I've stumbled across in the literature. Especially the two first items are often confused with videoconferencing.

Videophony. Biparty telephony with motion video. Usually low to medium frame rate and a frame size large enough to show a talking head view at low to medium resolution, sometimes with the forwarded video in a smaller window superimposed on the received video. Videophones may be video-extended telephone sets, so called video dialtones, or a computer equipped with necessary hardware and software.

Video seminar distribution. Generally includes a speaker and his/her notes. Mainly one sender and many receivers at a certain time. The video stream showing the speaker has often the same properties as for videophony. The notes, consisting of a chalkboard or a slide show, is generally fixed and thus doesn't need a high frame rate. On the other hand the notes normally includes important details which calls for a need of high resolution images. Sun Microsystems' Forum [1] is a typical example.

Media space. A media space is a computer controlled network of audio and video equipment for collaboration between groups of people distributed in space. A media space is continually available and is therefore not a service that is only available at certain predetermined times [2].

Video-wall. Also called a video-window. A very large television screen is set into the wall of common rooms at different sites. The idea is that as people wander about the common room at one site, they can see and talk to people at the other site, giving a sense of social presence.

The term *videoconferencing* is an abbreviation of audio-video conferencing. The objective of a videoconference is to support a meeting between more than two remote participants. If *biparty,* the conference connects groups of people; if *multiparty,* it may connect a mixture of groups and individuals. Often documents need to be exchanged, either on paper or in projected- or in electronic form.

2.1. Room-based v.s. desktop videoconferencing

There are two main groups of videoconferencing - room based videoconferencing and desktop videoconferencing. To understand the differences between the two, one can take a look at the history.

The circuit-switched-based videoconferencing systems appeared in the 1980s. The first services were provided by *Public Telephone Operators* (PTOs) in dedicated meeting rooms, equipped with analog audio-visual devices, digitizers and compressor/decompressor systems as well as a connection to the PTO's internal circuit-switched network. In the second half of the 1980s videoconference products appeared on the market in the form of packaged systems with TV cameras, microphones, speakers, monitors and modules for digitization, compression and decompression. They were installed in private *videoconference studios* connected by leased lines or aggregated telephone connections. These packaged, stand-alone videoconference systems are called *video-codecs* and generally are connected directly to circuit-switched networks. Most offer an optional document video camera and some support two or more room cameras. The next development was the introduction of *rollabout systems* - a circuit videoconference system that can be moved between meeting rooms. Lighter and cheaper than static video-codecs. Rollabout systems generally offer fewer optional features than static video-codecs. The latest generation of circuit-switched videoconference systems, *desktop systems*, is provided in offices.

At about the same time, videoconference systems using packet-based network technology evolved from interpersonal systems, i.e. videophony, to full multiparty desktop systems, and can finally be used in room or mixed room/desktop environments. The first generation of products was, however, not provided with dedicated facilities such as camera prepositioning, document camera, or sophisticated audio handling. Most applications use the IP or IPX protocol. They may offer high resolution at low framerate, but audio equipment is either fair or medium quality and echo cancellation is seldom treated properly."

The main difference thus is the settings for which the applications are optimized for. One important note on the difference between desktop- and room-based videoconferencing is that desktop videoconferencing applications generally take advantage of having a window system to provide additional information, e.g *awareness* of which other people are attending, and ability to use other applications at the same time on the same machine. Thus screen real-estate is a factor that must be considered in the *Graphical User Interface* (GUI) design. In a room-to-room conference you are not limited to run all these features on the same screen or even the same machine. Finn et al (in the bibliography) shows several examples where people have taken advantage of the spatial properties of the room to provide more intuitive awareness and support for social cues than can be delivered on the limited screen real-estate of a typical desktop computer.

The last observation has to do with the typical system resources available in the two different cases. Many adverts for room-based systems use the principle of pooling of resources as justification for the higher price: A room-based system can serve more people at once, it's easier to protect the components from damage, theft and so forth, and it's easier to service. Therefore it's possible to spend more money on better equipment and fancy features. Desktop-based systems on the other hand is intended to be used in an ordinary desktop computer and sharing resources with a lot of other activities. In this case the money is in the mass market, so the desktop videoconference system should be easy to use, require minimal resources, be possible to run anywhere and it should be cheap enough for an average computer user.

2.2. Summary

The objective of a videoconference is to support a meeting situation where the participants are not all co-located. From history we see that there are two main groups of videoconferencing systems, room-based and desktop-based, where the room-based systems were the first to emerge due to technological constraints. Cameras, codecs and audio equipment was expensive and difficult to operate. Over the years, these constraints have been removed leading to the emergence of consumer-grade audio-video equipment and software codecs for PCs and laptops, enabling desktop videoconferencing.

In the work presented in this paper, technological constraints once again force a room-based design, although the technology no doubt will catch up soon to provides the service to the consumer market.

3. A short introduction to the human being

In my work I assume that humans are the final end-points in the communication and therefore it is natural to take an average human's abilities as the base from where to study the videoconferencing system. Much data on this can be found in literature on *Human-Computer Interaction* (HCI) and *Computer-Supported Cooperative Work* (CSCW) that in turn references to works in the fields of psychophysics, ergonomics, information science and technology, systems design and cognitive psychology as the source of information about the human abilities and limitations.

In this chapter I will present some fundamental properties of the average end user.

3.1. The senses used in videoconferencing

The senses used in a videoconferencing session is mainly sight and hearing and therefore a lot of the information in the literature covers the limitations of these senses.

3.1.1. Sight

Sight is a complex function that involves the optical transformations and reception of light as decribed by the physiology of the eye as well as a lot of parallell processing of the sensory data by the retinal and cortical layers to produce a visual perception.

Only radiation of certain wavelengths is visible - those lying in the visible range: from about 250 nm to 780 nm. The human eye can discriminate between different wavelengths and each wavelength creates a different impression referred to as the *color sensation*. The eye is more sensitive to certain wavelengths than to others, implying that the human eye is more sensitive to certain colors than to others. For example yellow or yellow-green seems brighter than red or violet. This is a consequence of the distribution of the three types of cones in the retina.

A fact that is commonly used in perceptual video quality measurements is that human vision is more sensitive to contrasts than to general signal strength. This has to do with the center-surround organization of the visual neurons in the retina which makes the neurons fire if it receives light in a small central area while light in the larger surrounding area gradually inhibits the response.

Something that is overlooked in the design of today's videoconferencing systems is that sight is a three-dimensional sensation. To determine the distance to the point of focus, we use *binocular convergence*, that is the angle between the line of sight of each eye . We also use *binocular parallax* - the differences between the two images due to the space between the eyes, for the same purpose. By combining the two pieces of information, the brain creates a sensation of depth. Motion is also basic for visual perception. We move towards and from things, we look around them and move them to examine them closer. Movement also contributes with information about the three-dimensional layout of an object in form of *motion parallaxes*.

3.1.2. Hearing and speech

Sound consists of variations in the air pressure that are recorded by the ear and processed in the hearing center of the brain. The sound is characterized by its level (in dB) and its frequency (in Hz). Most people can hear sounds at levels between ca 20 to 100 dB, and frequencies between 20 to 15 000 Hz, which coincides with most most music contents. Normal speech uses between 30 to 70 dB and 100 to 5000 Hz although most of the actual speech content is located between 300 to 3400 Hz.

Another important factor is the speech melody. During a speech, sequences of phonemes follow silent periods. Typically 60% of speech consists of silent periods.

Normally, the human hearing filters out echoes of one's own voice arriving within a certain maximum time. Echoes of a speakers voice is distracting for both the speaker and the audience. The ITU-T has defined 24 ms as the upper limit of the one-way transit delay beyond which additional echo canceling techniques have to be employed.

The human is also able to determine the location of the source of a sound in a three-dimensional space. The first clue is based on the difference in intensity of the two stimuli presented to our ears. Likewise, the waveform will reach each of our ears at two distinct instants in time. The conjunction of the intensity and time difference produces an impression of lateralization. The outer ear also filters certain frequencies more than others, which helps detecting if the source is front, back, up or down from us. Sounds reverberate on surrounding objects and change when we move our head, which also helps in determining the position of the source. Three-dimensional sound helps to give situation awareness and to discriminate between participants in a conference.

3.2. Association and Transfer effects

The user's tolerance will generally be derived from their experience of comparable applications. For example, when viewing a movie-on-demand, a subscriber in the USA will compare the quality to that of NTSC over-the-air cable programs. Therefore, success with a new media is often dependent on whether the participants can use their existing norms. In the case of computer-mediated forms of communication, such as videoconferencing, we carry forward all our expectations and social norms from face-to-face communication.

3.3. Interaction

In an ordinary face-to-face meeting people interact and communicate with each other to achieve some goal. It is important to ensure that the videoconferencing system itself is as transient as possible and doesn't obstruct the process. The rules of face-to-face interaction are not conscious, so when they are broken we do not always recognize the true problem which inevitably causes confusion and a growing sense of unease. The two types of interaction in videoconferencing is

1. the interaction between the users and the system, and
2. the interaction between the users.

3.3.1. Man-machine interface

In the case of a room-based videoconferencing situation, the user interface consists of the room itself as well as the items in the room. Apart from the user interface design, the interaction between the users and the system also needs a short initialization and response time to be efficient and painless.

The *initialization delay* is a special case of response time in that it often involves physical actions and multiple steps. Therefore it can be allowed to be long, up to several tens of seconds, provided the user is kept updated on the progress of each step. The response time of the system after initialization should be under the *interactive threshold* of about one second.

3.3.2. Conversation

Conversation between people is a complex activity. It includes functions like interruptions and invitations to negotiate who should talk at a single moment, i.e. *turn-taking*. A common way to let others interrupt is to do a pause somewhat longer than it takes to breathe in, typically no more than a fraction of a second. If no one speaks up at that time the listeners should shut up until the next invitation. Most people regards untimely and repeated uninvited interruptions to their speech as rude.

Body language, such as facial expressions, gestures and mutual glances can also be used to invite someone else to take the floor. Tone of voice and eye contact are also crucial in enabling a smooth conversation.

3.3.3. Body language

Humans usually has broad experience of interpreting small nuances in *facial expressions*, *gestures* and *posture* and adapts their dialogue in reponse to these interpretations. If the speaker sees that the audience looks bewildered, then he can explain more in detail or ask the audience what causes the confusion. Socialpsychological studies has shown that ca 30% of a face-to-face conversation consists of *mutual glances*. Those glances are considered to have at least five functions; to guide the flow of conversation, to give feedback from the listener, to communicate feelings, to communicate the character of the relationship between the conversing people, and to mirror the status of the relationship between the conversing people. In short, sporadic direct eye contact is important in establishing a sense of engagement and social presence. Eye *gaze* is useful in establishing the focus of the conversation. If you say ‘now where does this screw go?’, there may be many screws, but if your colleague can see which one you are looking at then he/she is able to interpret which one you mean. In a similar but more direct way, we use our hands to indicate items of interest with *gestures*. This may be conscious and deliberate as we point to the item, or may be a slight wave of the hand or alignment of the body.

3.3.4. Personal space

When we converse with one another we tend to stand with our heads a fairly constant distance apart. We can accept people closer to us if they are at our sides or behind us than if we are facing them. These distances form a space called the *personal space*. The exact distance depends somewhat on context. A high level of noise may make people come closer just to be heard. Also if the conversants want to talk privately they tend to come closer. Personal space also differs across cultures, North Americans get closer than Britons, and southern Europeans and Arabs closer still. This can cause considerable problems during cross-cultural meetings.

3.4. The relation between video and audio

In a face-to-face meeting, most of the factual content is communicated through speech, while most of the feedback is in the form of body language. Thus most of the factual content is carried in the audio channel, while mechanisms for interaction can be carried in both the audio and video channel. The participants uses both audio and video media in a complicated pattern following partly subconscious rules, thus the relation between the audio and video channels is important.

The ear and the eye works very differently. The ear may be modelled as a differentiator, it is very easy to hear sounds from different sources even if they are intermixed. The eye works as an integrator, it is extremely difficult to recognize separate images if they are mixed, and it's difficult to see changes less than a few seconds long made in a familiar video clip. The consequence is that humans are much more sensitive to alterations of audio - than of visual signals and thus less tolerant of audio - than of video errors.

Synchronization between audio and video is another important factor in face-to-face communication. Synchronisation between audio and video is important to avoid misunderstandings about turn-taking and intent. Since video coding is generally more complex than audio coding, the coding delay is generally higher. Delayed video in relation to audio can be strenuous to look at for a longer time and don't give a full sensation of presence. For these two reasons trials with delayed audio to achieve lip synchronisation have been done. However, small audio delays have shown to seriously deteriorate participants ability to come to conclusion and also seriously lessen the participants' satisfaction with the conversation. Thus lip synchronization is highly desirable although not at the cost of additional audio delay.

4. Review of computer and communication support for videoconferencing

Videoconferencing is a *real-time service*. The data has to reach the destination withing a certain interval to have any value to the receiver. The audio and video in a videoconference are *continuous media*, i.e. the data should be delivered to the receiver with the same rate as it was generated by the sender. The audio and video data also has to be played out in the right order to be useful, and as we saw in section 3.4, audio-video synchronization is important. In this section I will briefly present the capabilities of todays computer and network hardware and software parts when it comes to supporting videoconferencing.

4.1. Computer systems architecture and real-time data

The computers today can be seen as multiprocessor systems since they generally consists of at least two busses interconnecting increasingly complex circuits. The development in *Digital Signal Processing* (DSP) technology further blur the distinction between processors and other circuitry. The consequence is that data can flow through the computer system without involving the *Central Processing Unit* (CPU). The two different architectures that I have looked at are:

- Bridge-based architecture.
- Shared memory architecture.

4.1.1. Bridge-based architecture

In this architecture only the CPU and main memory subsystem resides on the system bus. All other subsystems must reside on a peripheral bus, such as PCI or ISA, and communicate with the CPU and main memory via a *bridge circuit*. This is by far the most common computer systems architecture. The architectures of this type that I have looked into are

- Microsoft/Intel Windows BIOS [3, 4].
- Sun Microsystem’s UltraSPARC [5, 6].

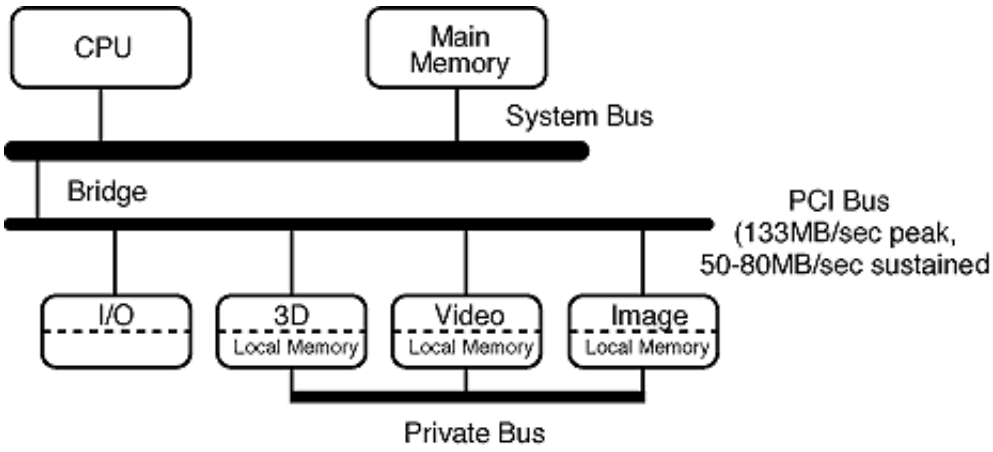

FIGURE 1. Microsoft/Intel Windows BIOS architecture [3].

In this architecture, the CPU has no real-time access to the data in the local buffers on the peripheral devices. The bridge is a shared resource and risks becoming a bottleneck. This makes it difficult to give any real-time guarantees for the transfer of data from one device to the next. The most common solution to this problem is to design stand-alone peripheral components that doesn't rely on the CPU or main memory and instead use a private bus to interconnect crucial devices as shown in Figure 1. The role of the platform machine is then reduced to power supply, and overseeing operations. Solutions consisting of this kind of specialized, stand-alone peripheral components tends to become quite expensive since you have to duplicate some of the platform resources on the peripheral devices, such as memory buffers and bus controllers in addition to the interfaces to the platform components. Another common problem is incompatible solutions from different vendors since the protocols used for communication between the devices over the private bus usually are closely guarded corporate secrets.

4.1.2. Shared memory architecture

In this architecture, the primary memory is shared by all devices, eliminating the need for on-board buffers and caches. To avoid complete chaos a memory controller keeps track of where different data is stored, who's writing to a certain buffer and who's reading from it. The only example of this architecture that I have is the SGI O2, built on their *Unified Memory Architecture* (UMA) architecture.

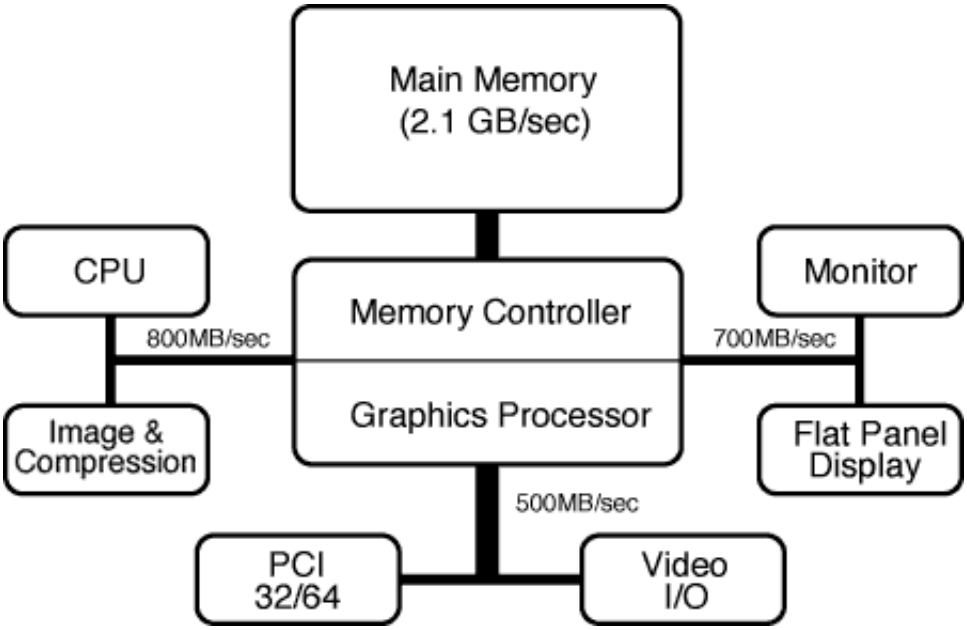

FIGURE 2. SGI Unified Memory Architecture [3].

The controller risks becoming a bottleneck, so it should match the hroughput of the memory. Another drawback with this shared memory type of system architecture is that it's difficult to introduce new special-purpose modules into the system if they don't fit well into the overall architecture. For example, if you insert a PCI-based network card in the system shown in Figure 2, it will reside on the same (slowest) bus as the Video I/O. If one would like to grab video from the Video I/O and then send it over the network the data would be transfered twice over the same bus and thus eliminating the whole point of the system, that is - no unnecessary copying of data.

4.2. Real-time support in operating systems

A typical multimedia application does not require any processing of audio and video to be performed by the application itself. Usually, data are obtained from a *source* and are forwarded to a *sink*. In such a case the data should take the shortest path through the system, i.e. copied directly from one device to another. Hence the application never actually touches the data. The system may have to support several such real-time *flows* together with sporadic non-real-time data transfers. The buses, the CPU and the main memory are resources shared by many processes and devices and the *Operating System* (OS) schedules which process is allowed to use which resource at a given time. The OS also takes care of a lot of other things as well, and there's a trend to put more and more functionality in the OS *kernel* to make the functions run faster. I will not go into more detail about that.

Other processes than the real-time ones initiated by the application itself are considered *competitive load*. Competitive load processes with long processing times may block a resource even though real-time processes are waiting, preventing them from meeting their deadlines. This can be avoided by using a technique called *preemption*. Preemption allows a higher priority process to grab a resource even if it is used by another process. *Priority-based scheduling* allows preemption. In a *Real-Time Environment* (RTE) processing of data is scheduled according to the inherent timing requirements of the data. One such scheduling algorithm is the *Earliest Deadline first* (EDF).

Advance allocation allows periodic processes to allocate resources in advance. Unfortunately, network traffic is seldom very regular, thus forcing the use of interrupts instead. *Interrupt handling* usually has highest priority and preempts any process that is running. so the overhead of interrupt handling should be kept at a minimum. Changing status of a process to/from running/ready causes a context switch, which is a costly operation. Thus the number of context switches should be minimized as well.

4.3. Multipoint distribution techniques

From the definition of videoconferencing in chapter 2 we see that it can be either biparty, the conference connects groups of people, or multiparty, the conference may connect a mixture of groups and individuals. For biparty, the setup is quite straightforward, while for multiparty you have some different distribution techniques to choose from. I have looked into the three currently most commonly used techniques - point-to-point mesh, reflector nodes and IP multicast. However, we have to take into account the human users as well. For multiparty videoconferences it has been found that if the meeting is *symmetrical* - that is, with balanced contributions from all participating sites - six to eight systems is a practical limit beyond which floor passing becomes difficult to manage. Beyond 12 sites anarchy becomes inevitable, unless some participants are very passive.

4.3.1. Point-to-point mesh

The earliest and still most robust way to provide multicasting is to set up a mesh of connections between the participants. Every end-system is directly connected to all the others, requiring $n^2 - n$ connections to fully interconnect n systems. IP-based point-to-point meshes are implemented in the Application Layer of the *International Standards Organization/Open Systems Interconnection* (ISO/OSI) reference model. *Integrated Services Digital Network* (ISDN) or similar circuit switched networks requires as many biparty connections (circuits) as remote sites for unidirectional transmission. A practical observed limit for ISDN circuits is one source and seven receivers. One advantage with point-to-point connections is that confidentiality is higher as the conference only involves the members specifically accepted and with which dedicated connections have been set up.

The obvious disadvantage of using a point-to-point mesh for multipoint distribution is that it is based on *sender-based copying*, i.e. each sender has to send identical copies to all other participants. Early versions of ISABEL [7] used point-to-point as well as Communique! [8].

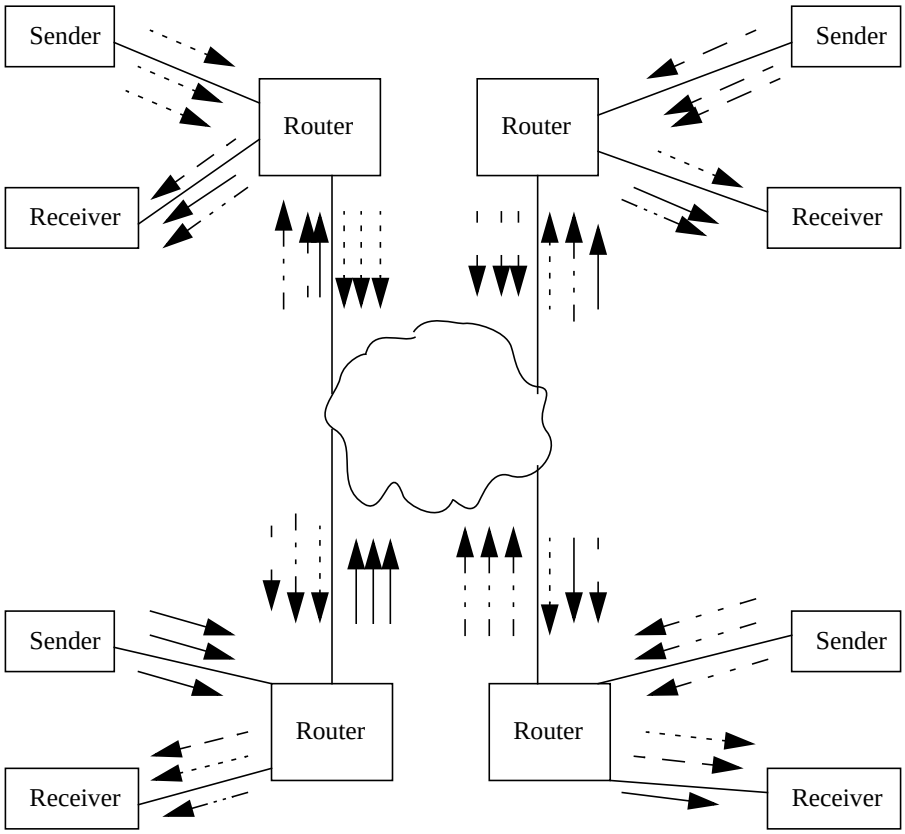

FIGURE 3. A point-to-point mesh use sender-based copying.

4.3.2. Reflector node

In circuit switched networks, a *Video Branch eXchange* (VBX), or videoconferencing hub (video-hub for short) is used to provide a star point which acts as a switch between all participating video-codecs. Video hubs can be switching digital signals directly, e.g. a *Multiparty Conferencing Unit* (MCU), or the signals can be converted to analog signals, called *video-switches* or video mixers. Video switches generally introduces an additional one second delay due to the D/A and A/D conversion. Advanced MCUs are capable of translating between different audio and video encoding and compression schemes. They are called *transcoder MCUs*. Usual video hubs allow for eight participating systems in multiparty conferences, some systems support up to 24 calls. Video hubs may be chained to form a cascade, thus extending the topology from a star to a tree scheme. Many video-hubs use *voice-activation* to only forward video from the party generating sound at a given moment.

The most commonly used videoconferencing applications today use ITU-T H.32x which supports multipoint conferencing through the use of a MCU. CU-SeeMe [9] is an example of a non-H.32x conferencing product using reflectors. The IP-based reflector approach is implemented in the Application Layer of the OSI model.

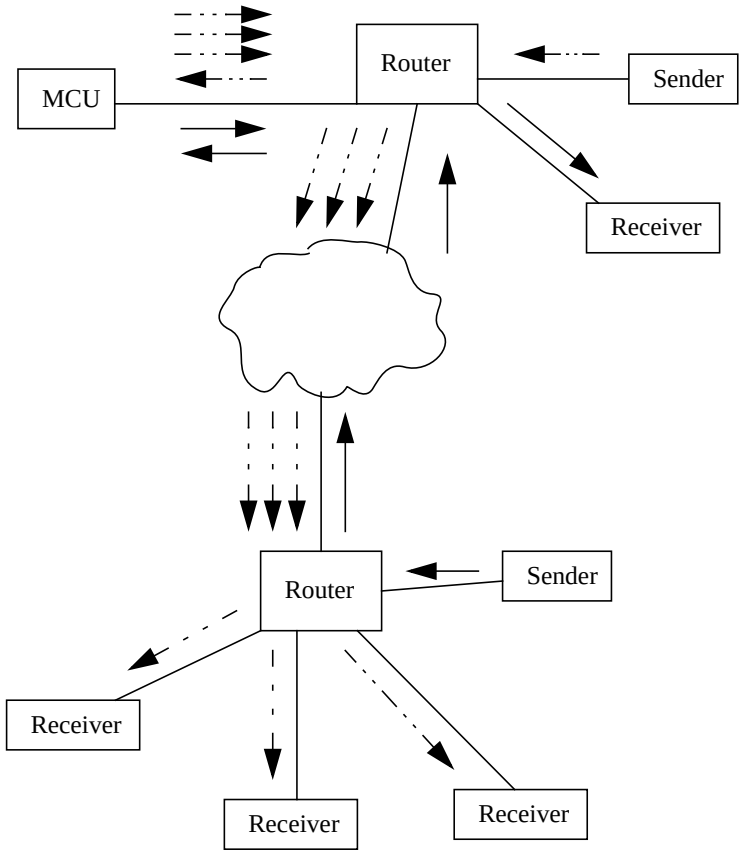

FIGURE 4. The effect of a misplaced reflector node.

4.3.3. IP multicast

A solution that is only available to IP-based systems is to use IP multicast support in the infrastructure to provide multipoint conferencing. IP multicast is implemented in the Network Layer of the OSI model and relies on the capability of the network to replicate, at certain internal points, the data emitted by a source. Replicated data should only be forwarded to the recipients which are part of the multicast group to minimize the number of network segments that has to be traversed by multiple copies of the same data. Depending on the network topology, the use of IP multicast instead of a MCU or a mesh helps avoiding unnecessary waste of bandwidth.

IP multicast has the potential to scale better to large numbers of participants in a conference, with respect to network- and end-host resources, than the reflector- and point-to-point mesh solutions. Although this is not so important in the videoconferencing case where floor handling sets an upper limit anyway. The MBone tools introduced in section 6.1 uses IP multicast.

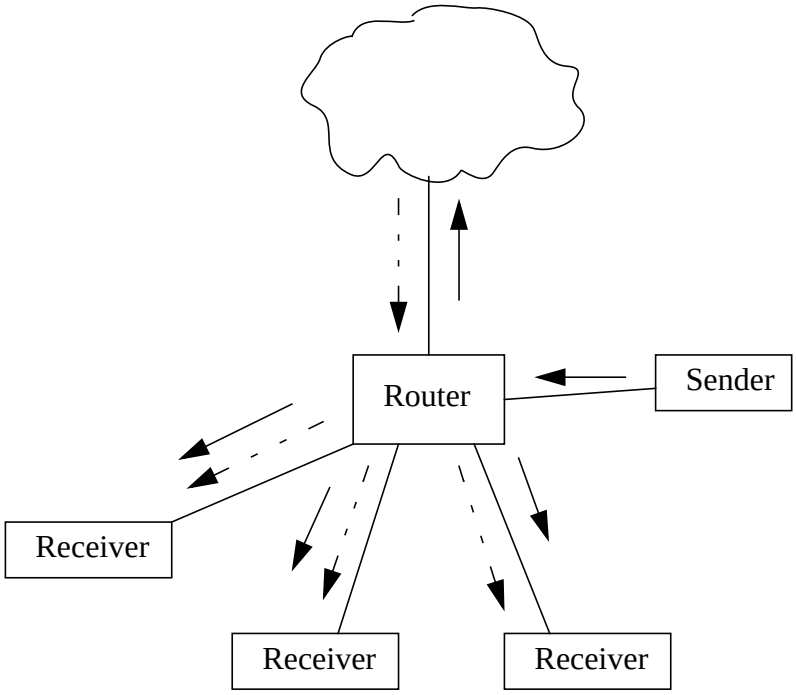

FIGURE 5. IP multicast use router-based copying.

A sender to an IP multicast group need not be a member of that group and the latest version of the *Internet Group Management Protocol* (IGMP) allows receivers to specify which senders it wants to listen to and which senders it doesn't want to listen to. Unfortunately IGMP is only used in the steps between sender/receiver and the nearest router, and thus cannot prune unwanted traffic from the whole system unless there is a corresponding functionality also in the multicast routing protocol used.

More information on IP multicast can be found in [10, 11, 12, 13, 14].

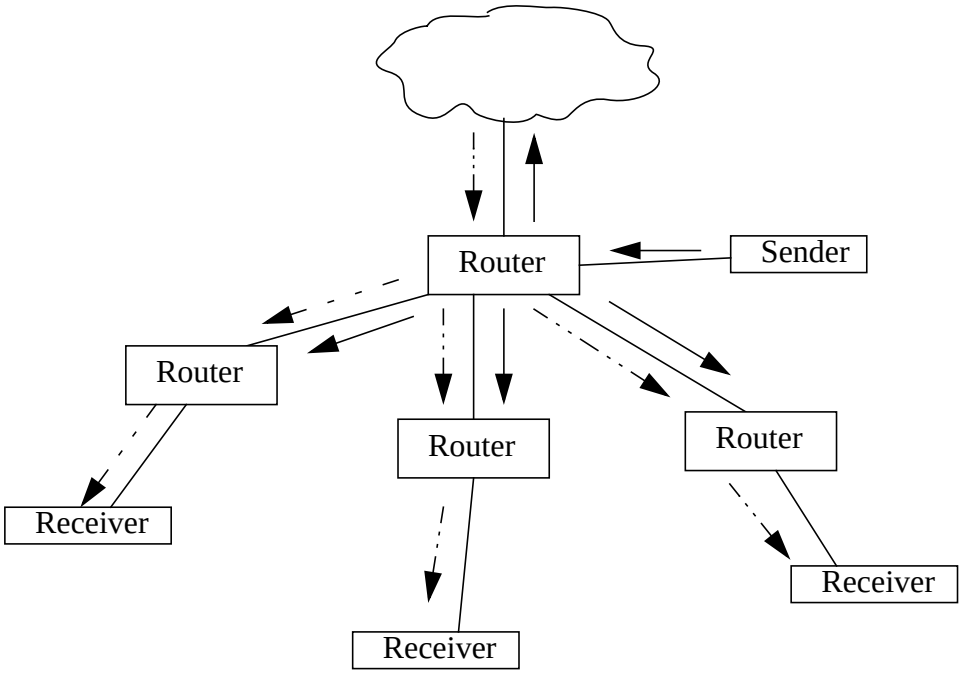

FIGURE 6. Drawback with IGMP v3 solution.

4.4. Real-time Transport Protocol (RTP)

The real-time transport protocol (RTP) [15] provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data. It runs over both multicast and unicast network services. RTP is accompanied by a real-time control protocol (RTCP) to monitor the data delivery and convey information about the participants in a session. Both RTP and RTCP are designed to be independent of the underlying transport and network layers although most applications are using RTP on top of UDP/IP. RTP is an application level protocol which means that it must be incorporated into applications using the protocol rather than in the TCP/IP protocol stack. RTP provides a timestamp field to allow payload-specific timing reconstruction at the receiving end and a sequence number to allow detection of lost and reordered packets.

RTP is designed to carry real-time data, that is if a certain datagram was delayed, you cannot wait forever for it. After a certain threshold, the contents of the late datagram have no value and will be discarded if it arrives. There isn’t always time to wait for retransmission so no ARQ method is included in the protocol.

The RTP protocol itself specifies those functions expected to be common across all types of real-time applications but it is also intended to be tailored through modifications and/or additions to the headers as needed. Most of the fields in the RTP header can have different uses depending on the type of data to be carried. For example the

RTP header includes a marker bit that can mark the beginning of a talk spurt or the end of a video frame, or it can be unused. All according to the needs of the application. Functions that are shared by a group of applications with similar demands are collected into a common RTP Profile. A RTP Profile, in turn, is usually complemented by a RTP Payload Format specification for each application describing additional functionality needed that is not shared by the rest of the group.

More on RTP can be found in [15, 16].

4.4.1. RTP A/V profile

The RTP profile relevant to this work is the RTP Profile for Audio and Video Conferences with Minimal Control [17], or the A/V profile for short. This RTP profile defines modifications to the RTP and RTCP headers suitable for audio and video data transmission and defines a set of payload type codes and their mapping to payload formats.

4.4.2. RTP Payload Formats

There are quite a few RTP Payload Formats today. H. Schulzrinne keeps a quite up to date listing at [18]. The RTP Payload Formats may specify additional header extensions to support partial decompression in the presence of loss or whatever information that the application needs apart from the payload itself. It usually defines the function of the marker bit and the timestamp field. M. Handley show the purpose of - and guidelines on how to write RTP Payload specifications in [19].

5. An introduction to audio and video Coding

The sound and light that humans utilize in their communication is in the analog domain. A multitude of sound and light sources contributes to the signals received and interpreted by the senses. In the computer and most telecommunication and computer networks of today, the information is in digital form. The main reasons for this is to avoid noise pick-up and signal losses. The digital representation of the media is also easier to manipulate, e.g. adding special effects and editing.

To represent an analog signal digitally we take samples of the signal amplitude at certain times, most often using fixed intervals. On the receiver side we use filters to recreate a smooth signal from the discrete samples. The *Nyquist theorem* guarantees that no information is lost in the interval between samples if we take at least twice as many samples per second as the highest frequency present in the signal. For performance reasons the samples can only be encoded as a limited number of discrete values and we thus lose information in round-off errors.

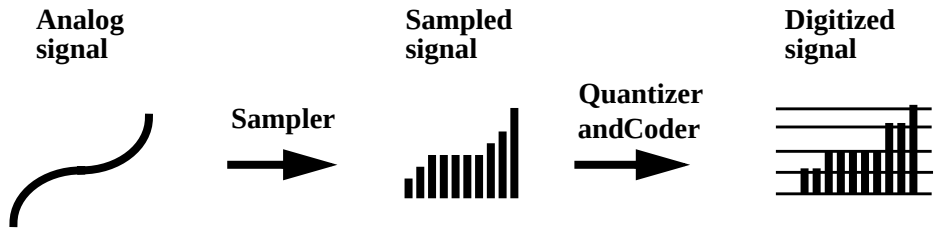

FIGURE 7. Digitization of an analog signal into a PCM encoded digital signal.

5.1. Audio coding

Raw sampling of the whole frequency band that a human can perceive accoding to the Nyquist theorem would need about 40 000 samples per second. This is the simplest form of encoding where samples of the signal's amplitude are quantized linearly over the spectrum. Taking into account that the sensitivity of the ear is not uniform over the spectrum one can adopt a non-linear quantization function and get a better resolution in the more important parts, or offer the same subjective quality using fewer bits per code. One can also apply more complicated algorithms to further reduce the transmission bandwidth needed. In the following sections I will briefly introduce some common audio coding standards.

5.1.1. Audio in Television

In Sweden, the analog audio is frequency modulated onto a carrier situated 5.5 MHz above the carrier for video. The peak frequency deviation is 50 kHz. The audio signal records ca 15 Hz to 15 kHz originally but the quality deteriorates quite fast.

In 1988 Sweden started using the *Near-Instantaneous Companded Audio Multiplex* (NICAM) system for broadcasting digital stereo audio multiplexed with the analog video and audio in the TV signal. The NICAM format use 14 bit codes sampled at 32 kHz and gives better quality than its analog counterpart [20].

5.1.2. Speech synthesis codecs

Speech synthesis coders use a model of the vocal tract to analyse and produce a code that describes the incoming speech. This generally gives the reconstructed speech an impersonal tone, but allows for very low bit rates. The incoming speech signal is divided into blocks of samples and each block is called a frame. The longer frame length, the lower bit rate, but poorer synthesised speech quality. Generally speaking, 20 msec frame length is used for most of current speech synthesis codecs; e.g.: QCELP-13 and EVRC for CDMA, US-1 for TDMA, RELP for GSM. And 25 msec frame length is used for low rate speech codec.

5.1.3. The ITU-T G.700-series of voice codecs

The International Telecommunication Union Telecommunication Standardization Sector (ITU-T) G.700-series of recommendations for speech coders records audio from the range of sound frequencies 300 to 3400 Hz where most of the speech content is located. The most well known is the G.711 *Pulse Code Modulation* (PCM) of voice frequencies with a sampling rate of 8 kHz and 8 bit logarithmic samples generating 64 kbits/s. Other commonly used codecs are G.723.1. that generates 5.3 or 6.3 kbits/s and G.729 that generates 8 kbits/s.

5.1.4. CD and DAT

The *Compact Disc-Digital Audio* (CD-DA) format is sampled at 44.1 kHz, and the *Digital Audio Tape* (DAT) formats are sampled at 32, 44.1 and 48 kHz respectively. The recorded frequency band covers the whole audio frequency band that a human can percieve (actually the 48 kHz DAT records a bandwidth of 24 kHz). The CD and DAT formats uses a linear 16 bit Pulse Code Modulation (PCM) encoding of the samples, and DAT also supports a non-linear 12 bit encoding. This gives an imperceptible quantization noise level. With stereo channels the 48 kHz, 16 bit DAT format needs more than 1.5 Mbps and 44.1 kHz, 16 bit CD generates 1.4 Mbps. The 32 kHz, 12 bit DAT format needs 768 kbits/s.

5.1.5. MPEG audio compression

MPEG-1 (ISO/IEC 11172-3) provides single-channel (’mono’) and two-channel (’stereo’ or ’dual mono’) coding at 32, 44.1, and 48 kHz sampling rate. The MPEG-1 Audio uses a fast Fourier transform followed by quantization according to a psychoacoustic model. The accuracy and complexity of the model is defined for three *Layers*. The predefined bit rates range from 32 to 448 kbit/s for Layer I, from 32 to 384 kbit/s for Layer II, and from 32 to 320 kbit/s for Layer III.

MPEG-2 BC (ISO/IEC 13818-3) provides a backwards compatible multichannel extension to MPEG-1; up to 5 main channels plus a *Low Frequent Enhancement* (LFE) channel can be coded; the bit rate range is extended up to about 1 Mbit/s; an extension of MPEG-1 towards lower sampling rates 16, 22.05, and 24 kHz for bit-rates from 32 to 256 kbit/s (Layer I) and from 8 to 160 kbit/s (Layer II & Layer III).

MPEG-2 AAC (ISO/IEC 13818-7) provides a very high-quality audio coding standard for 1 to 48 channels at sampling rates of 8 to 96 kHz, with multichannel, multilingual, and multiprogram capabilities. AAC works at bit rates from 8 kbit/s for a monophonic speech signal up to in excess of 160 kbits per second and channel for very-high-quality coding that permits multiple encode/decode cycles. Three profiles of AAC provide varying levels of complexity and scalability.

MPEG-4 Audio (ISO/IEC 14496-3) provides coding and composition of natural and synthetic audio objects at a very wide range of bit rates [21].

5.1.6. DVI audio compression

DVI is a proprietary technique for audio and video encoding developed by RCA in 1987 and acquired by Intel in 1988. In 1993 a software-only decoder became available as the product Indeo. Audio signals are digitized using 16 bits per sample and are either PCM-encoded or compressed using the Adaptive Differential Pulse Code Modulation (ADPCM) encoding technique. Different sampling frequencies are supported; 11025 Hz, 22050 Hz, and 44100 Hz for one or two PCM-encoded channels and 8268 Hz, 31129 Hz and 33075 Hz for ADPCM for one channel [22].

5.1.7. DV audio compression

In the HD Digital VCR Conference (DV) format [23], video and audio are coded together within a DV frame. There are many different encoding types but the one commonly used in consumer-grade video camcorders is the Standard-Definition Digital VCR (SD-DVCR) format. The audio coding is common for both formats, only the number of audio samples per frame differ. There are four different audio encoding modes defined in. 16 bits linear coding of one channel at 48, 44.1 or 32 kHz and 12 bits nonlinear coding of two channels at 32 kHz. I.e. about the same audio quality as for mono CD-DA or stereo DAT can be obtained. The bit rate is also the same - 768 kbits/s for a 32 kHz stereo or 48 kHz mono, and 512 to 705.6 kbits/s for the other two formats.

5.2. Video

Since the early days of moving pictures and cinemateques, *video* has been represented as a time-ordered sequence of images shown in rapid succession to give an impression of motion.

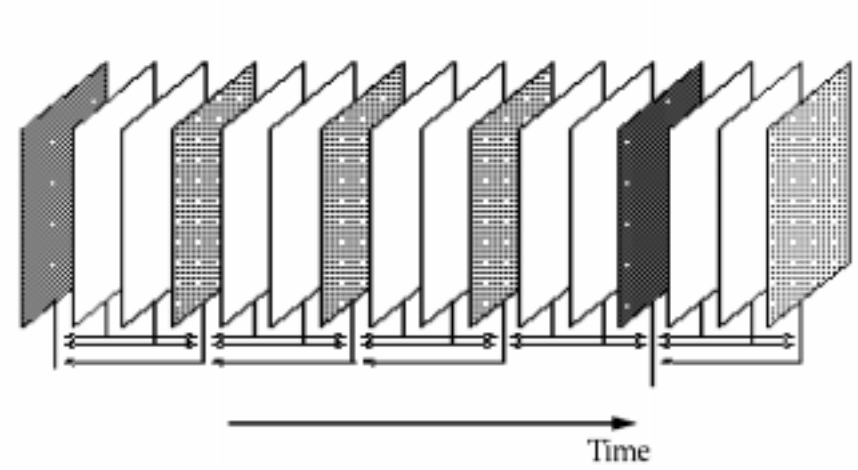

FIGURE 8. Video is a succession of images.

In movies the light is filtered by the film and the resulting image can be displayed on a white surface or can be viewed directly.

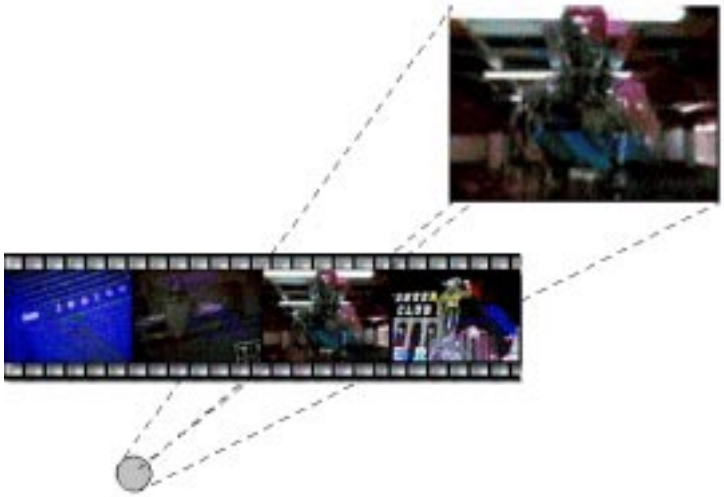

FIGURE 9. Video using photographic film.

Thus all *picture elements* (pixels) are projected simultaneously and the resolution (number of pixels) as well as the number of colors depends on the quality of the film. In television and digital video the representation is different.

5.2.1. Video in Television

There are three major analog TV broadcasting standards: the National Television Systems Committee (NTSC) standard that is used in North America and Japan. Sequentiel Coleur à Mémoire (SECAM) is used in France, part of Central and Eastern Europe, former USSR republics and part of the Arab peninsula. Phase Alternating Line (PAL) is used in the rest of Europe, part of Asia and part of Africa including South Africa. The analog TV broadcasting standards cover transmission of both audio and video signals. Here we will look briefly at the characteristics of the video part.

Thanks to the discipline of colorimetry we don’t have to sample the intensity of each wavelength that the eye can perceive - just three of them. For analog television, the incoming light is filtered into three images red, green and blue (RGB) and each of these images are linearized using *scan lines*. No specific screen dimensions are specified. Instead, the image dimensions are determined through the *aspect ratio* - the ratio between horizontal and vertical resolutions - and the number of scan lines. For the TV standards of today, the aspect ratio is 4:3 while the number of lines per frame differ. For NTSC each image consists of 525 lines of which 41 are blanked out during the vertical retrace interval, leaving 484 so called *active lines* visible on the screen. For SECAM and PAL, the corresponding numbers are 625 and 575 respectively.

Note that the number of active lines is the maximal vertical resolution which can be achieved without creating artificial lines using interpolation or some other scaling algorithm.

When the first TV systems were designed the image scan frequency had to be 50 Hz in Europe and 60 Hz in USA to avoid interference from residual frequencies of the electric power distribution network. These are still the most commonly used scan frequencies even if they have very little to do with human capabilities. For TV the images usually are *interlaced*, i.e. only every second line is scanned at each scan pass. This way you save bandwidth with little perceived quality degradation. Consequently, NTSC has an effective frame rate of 29.97 frames per second and SECAM and PAL has a frame rate of 25 frames per second, where a *frame* consists of two *fields*, each sampled at different instants in time.

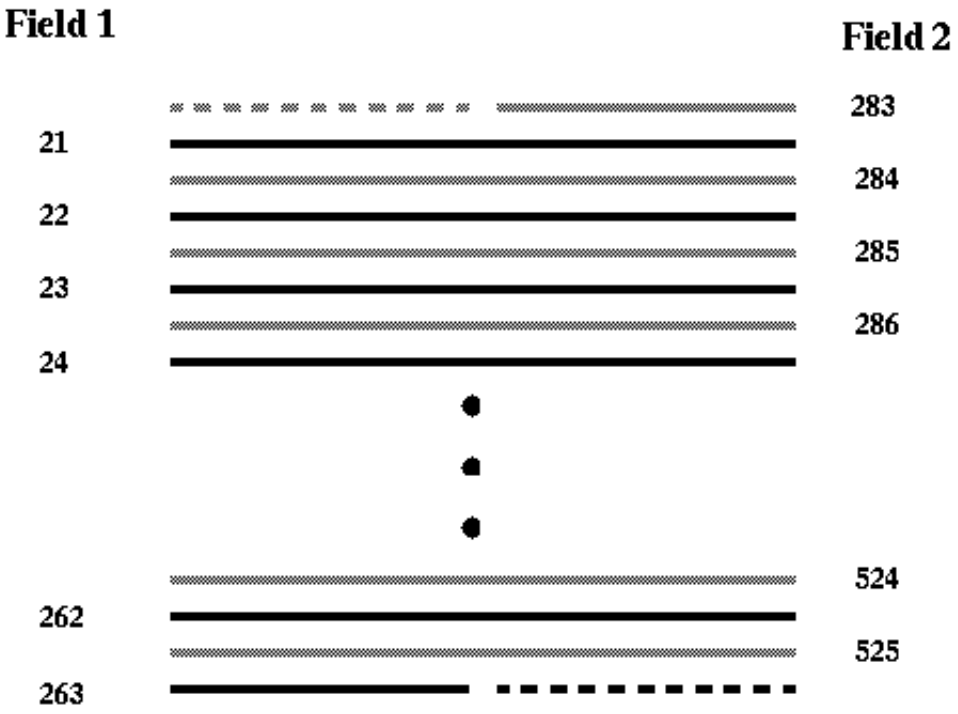

FIGURE 10. Active part of an interlaced NTSC frame.

The R, G and B signals is usually transformed into a *luminance* signal and two *chrominance* signals. This is to allow backward compatibility with old black and white TV sets where only the luminance signal is used. Also, since the human visual system is less sensitive to color than to luminance, the chrominance signals can be transmitted in narrower bandwidth and lower accuracy in an analog broadcasting network. The transform used in NTSC is called *YIQ* while the transform used in PAL is called *YUV*.

5.2.2. Digital Video

To digitize the analog video, the luminance and chrominance signals have to be sampled, quantized and coded. The International Telecommunication Union Radiocommunication Sector (ITU-R) has standardized the way in which current analog TV content may be digitized in it's recommendation BT.601. This recommendation defines a sampling of the incoming luminance signal at 13.5 MHz irrespectively of the original analog format. This means that it also samples the vertical and horizontal retraces and retains the same number of lines, field rate, field order and aspect ratio as the original analog signal. With this sampling rate the number of samples per line will be 858 for NTSC and 864 for SECAM and PAL. The ITU-R has adopted a common number of 720 *active samples* per line for all the original signals.

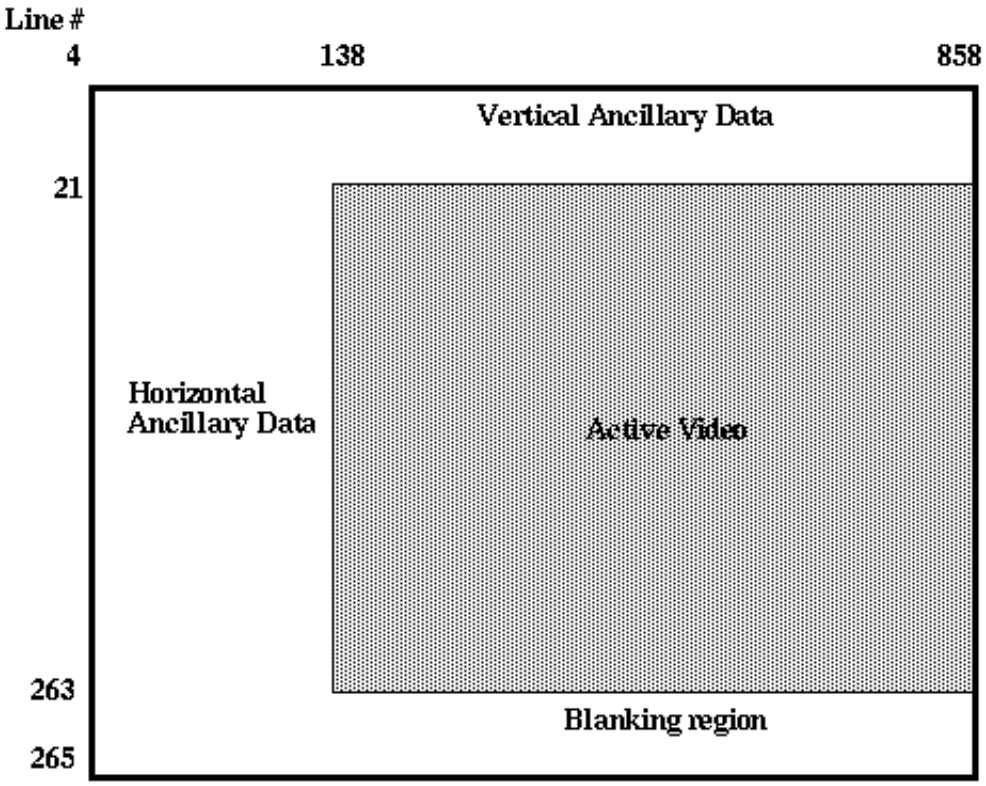

FIGURE 11. ITU-R BT.601 samples from NTSC field 1 [24].

Each sample is coded with 24 bits, one byte each for the luminance, denoted by *Y*, and the *color difference* signals, denoted by *Cb* and *Cr*. ITU-R BT.601 also defines different ways to *subsample* the color difference signals to further take advantage of the fact that the human visual system is less sensitive to color than to luminance and can thus reduce the bit rate. All video compression schemes that I have studied expects subsampled input. Below I will give an account of some common video encoding schemes.

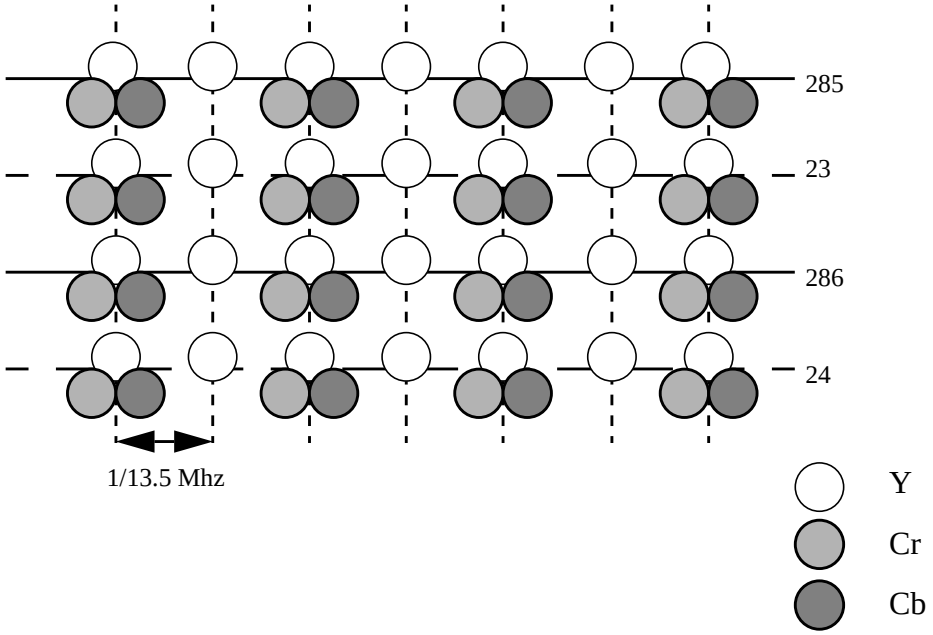

FIGURE 12. ITU-R BT.601 stream subsampled at 4:2:2.

5.2.3. ITU-T H.26x video compression

The first video compression standard developed by the ITU-T was the recommendation *H.261-Video codec for audiovisual services at px64 kbit/s* [25] aimed for use in videotelephony applications using ISDN as transport media. It is still widely used in H.320-based videoconferencing applications over ISDN and is also used in the MBone tools. The H.261 codec does it's own 4:1:1 subsampling of a ITU-R BT.601 stream with a sampling clock generating a maximum of 30000/1001 frames per second. The H.261 also defines it's own frame formats; *Common Intermediate Format* (CIF) and *Quarter-CIF* (QCIF). There is also an optional format called *Super-CIF* (SCIF) or *4CIF* that is seldom implemented. The picture quality varies depending on the bit rate used. A bit-rate of 384 kbits/s (6 ISDN B-channels) is commonly believed to provide sufficient picture quality for CIF format [26].

TABLE 1. Coded pixels in CIF-based frame formats [25].

	QCIF	CIF	4CIF
Luminance	176x144	352x288	704x576
Chrominance (Cb)	88x72	176x144	352x288
Chrominance (Cr)	88x72	176x144	352x288

Since H.261 can be seen as a subset of H.263 v.1 and v.2, there is no further development of the standard within the ITU-T.

H.262-Information technology - Generic coding of moving pictures and associated audio information: Video is the ITU-T notation for ISO/IEC International Standard 13818-2, also known as MPEG-2.

H.263-Video coding for low bit rate communication [27] was initially intended for use with H.324 videotelephony over *Public Switched Telephone Networks* (PSTN), but it has reached a wider acceptance and is now also used in MPEG-4 and H.323 videotelephony over Non-guaranteed Quality of Service Networks. H.263 provides better image quality at low bit rates with little additional complexity compared to H.261 [28]. The source coder can operate on five standardized picture formats: sub-QCIF, QCIF, CIF, 4CIF and 16CIF. The standard defines the maximum size of a frame in kbits, called BPPmaxKb, which means that the encoder should control its output bitstream. The minimum allowable value of the BPPmaxKb is defined for each of the source picture formats to guarantee a reasonably good image quality.

TABLE 2. Minimum value of BPPmaxKb [27].

Source format	BPPmaxKb
sub-QCIF	64
QCIF	64
CIF	256
4CIF	512
16CIF	1024

H.263 v. 2 -Video coding for low bit rate communication [29], also called H.263+, is an extension of H.263 with more optional modes and provision for customized frame sizes between 4 to 2048 pixels in X and Y with a step of 4 pixels granularity. Thus, using H.263v2 can be more computationally expensive than encoding with H.263.

5.2.4. MPEG video compression

Currently there are three finished MPEG video compression standards, MPEG-1, MPEG-2 and MPEG-4 version 1. MPEG-4 version 2 is in the final stage of standardization. The difference between these standards is the context for which the algorithms are optimized as will be explained below.

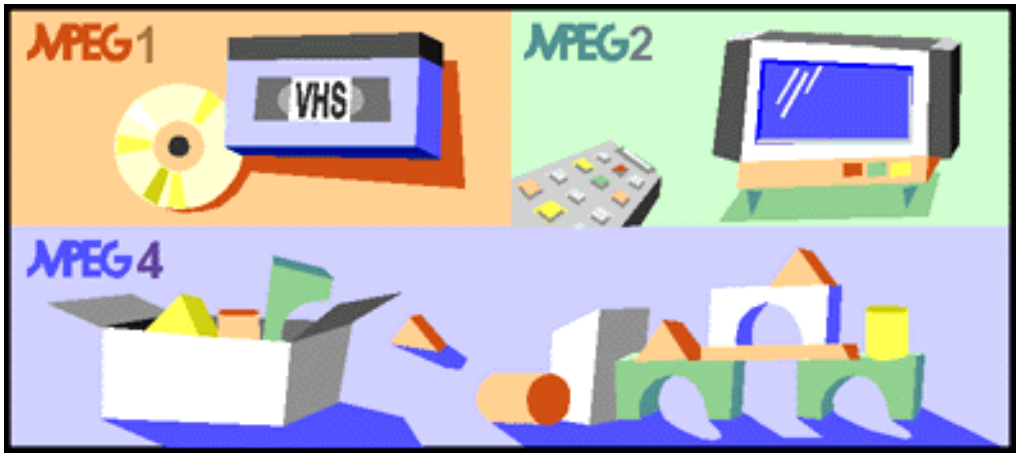

FIGURE 13. Relation of MPEG standards [21].

MPEG-1 Video (ISO/IEC 11172-2) is addressing the compression of video signals at about 1.5 Mbps [30] and was developed to operate principally from storage media operating at that bit-rate, e.g. CD-ROM. It is built upon experiences from H.261, JPEG and CMTT/2.

All MPEG-1 Video decoders should be capable of decoding a *Constrained Parameter Set* (CPS) bit stream:

- Horizontal size up to 720 pels
- Vertical size up to 576 pels
- Picture rate up to 30 fps
- Bit rate up to 1.86 Mbps

The most common format for MPEG-1 Video material is the *Source Input Format* (SIF) although higher resolution formats are possible. MPEG-1 uses 4:1:1 and 4:2:0 subsampling and progressive scan (non-interlaced) internally.

TABLE 3. MPEG-1 Video formats [30].

Format	Dimensions	bit-rate example
SIF	352x240 (NTSC), 352x288 (PAL)	1.2 - 3 Mbps at 30 fps
ITU-T BT.601	720x486 (NTSC), 720x575 (PAL)	5 - 10 Mbps at 30 fps
EDTV	960x486 (NTSC), 960x575 (PAL)	7 - 15 Mbps at 30 fps
HDTV	1920x1080	20 - 40 Mbps at 30 fps

MPEG-1 is not optimized for formats outside the CPS even if it has been used for such applications. MPEG-2 (ISO/IEC 13818-2) on the other hand was designed for formats above CPS with Digital TV in mind. Thus it is better suited for handling interlaced input and large resolution material.

TABLE 4. Typical MPEG-1 and MPEG-2 coding parameters [31].

	MPEG-1	MPEG-2
Standardized	1992	1994
Main Application	Digital video on CD-ROM	Digital TV (and HDTV)
Spatial Resolution	CIF	TV (HDTV)
Temporal Resolution	25 - 30 frames/s	50-60 fields/s (100-120 fields/s)
Bit Rate	1.5 Mbit/s	appr. 4 Mbit/s (20 Mbit/s)
Quality	comparable to VHS	comparable to NTSC/PAL

MPEG-2 has introduced the concept of “Profiles” and “Levels” to stipulate conformance between equipment not supporting the full specification. The most commonly implemented combination is the Main Profile at Main Level MP@ML which is used in e.g. DVD and Digital TV.

TABLE 5. Upper bound of parameters at each level for a NTSC feed[31].

Level	Format	Frame rate	bit-rate example
HIGH	1920x1152	60 frames/s	80 Mbit/s
HIGH 1440	1440x1152	60 frames/s	60 Mbit/s
MAIN	720x576	30 frames/s	15 Mbit/s
LOW	352x288	30 frames/s	4 Mbit/s

TABLE 6. Algorithms and functionalities supported with each profile [31].

Profile	Algorithmss	Representation
HIGH	Supports all functionality provided by the Spatial Scalable Profile plus the provision to support 3 layers with the SNR and Spatial scalable coding modes.	4:2:2 YUV
SPATIAL Scalable	Supports all functionality provided by the SNR Scalable Profile plus an algorithm for Spatial scalable coding (2 layers allowed).	4:2:0 YUV
SNR Scalable	Supports all functionality provided by the MAIN Profile plus an algorithm for SNR scalable coding (2 layers allowed).	4:2:0 YUV
MAIN	Non-scalable coding algorithm supporting functionality for coding interlaced video, random access and B-picture prediction modes.	4:2:0 YUV
SIMPLE	Incudes all functionality provided by the MAIN Profile but does not support B-picture prediction modes.	4:2:0 YUV

MPEG-4 Video version 1 (ISO/IEC 14496) [32] is a higher level protocol based on an object-oriented model. A *video scene* can be a composition of arbitrarily shaped video objects according to a script that describes their spatial and temporal relationship. A *video object* can be of any type of pixel-based video content, i.e. not just the kind of video that I described above but also computer-generated animation formats and still images. Since the *video content* can be of any of MPEG-1, MPEG-2, and H.26x as well as uncompressed video the following video formats are supported; SQSIF/SQCIF, QSIF/QCIF, SIF/CIF, 4SIF/4CIF, ITU-R BT.601 and ITU-R BT.709 as well as arbitrary sizes from 8x8 to 2048x2048. The *color space* can be any of monochrome, YCrCb and RGB where the pixel depth can be up to 12 bits per component. The chrominance subsampling schemes supported are 4:0:0, 4:2:0, 4:2:2 and 4:4:4. The maximum frame rate can take any value and frames can be either interlaced or progressive.

MPEG-4 Video version 1 is optimized for the following bit-rate ranges: < 64 kbits/s, 64 to 384 kbits/s and 384 to 4 Mbps. For some applications, bit-rates up to 50 Mbps for ITU-R BT.601 and 300 Mbps for ITU-R BT.709 are necessary.

5.2.5. DVI video compression

DVI video distinguishes two techniques with different resolutions and dissimilar goals [22].

Presentation-Level Video (PLV) has better quality at the expense of a a very complex asymmetric compression scheme. PLV is suitable for applications distributed on CD-ROMs. The frame format is 512x480.

Real-Time Video (RTV), can be used for interactive communication in the same manner as H.261. Frame format 256x240. In RTV, each pixel is 8 bits where the chrominance is subsampled at 4:1:0. I.e. one chrominance sample for each 4 luminance samples where the chrominance sample is taken alternatingly from Cb and Cr.

5.2.6. MJPEG video compression

The *Joint Photographic Experts Group* (JPEG, ISO DIS 10918-1 and ITU-T Recommendation T.81) standard [33, 34] defines a family of compression algorithms for continuous-tone, still images. This still image compression standard can be applied to video by compressing each frame of video as an independent still image and transmitting them in series. Video coded in this fashion is often called *Motion-JPEG* (MJPEG) [35].

JPEG accepts images in both RGB and YCbCr color spaces. The JPEG standard does not define the meaning or format of the components that comprise the image. Attributes like the color space and pixel aspect ratio must be specified out-of-band with respect to the JPEG bit stream. The *JPEG File Interchange Format* (JFIF) [36] is a defacto standard that provides this extra information using an application marker segment.

In [35] transport is defined for frame formats from 0x0 to 2040x2040 at steps of 8 pixels in X and Y. In JPEG itself there is no such upper limits defined, but the step size is the same.

5.2.7. DV video compression

I will concentrate on the SD-DVCR part of the video formats defined by the HD Digital VCR Conference in [23]. In this mode the video is coded using interlaced scan.

The SD-DVCR captures standard television frame formats for both NTSC and PAL. The analog feed is sampled according to ITU-R BT.601 and is subsampled at 4:1:1 for NTSC and 4:2:0 for PAL feed. The field-rate is 60 for NTSC and 50 for PAL and the frame formats are 720x480 for NTSC and 720x576 for PAL. The bit-rate of the compressed video is regulated to about 25 Mbps for both NTSC and PAL [37].

5.3. Audio and video distortion measurement methods

Unintentional artificial signals, so called *artefacts*, may have a significant impact on the perceived quality of an audio-video communication. Audio artefacts may be caused by delay distortion, intermodulation noise, crosstalk, or impulse noise in analog feed cables or introduced by the compression algorithm. A common source of video artefacts are the different codecs introducing contour-, blocking- and texture artifacts, mainly caused by quantization in combination with simplifications in the implementations of algorithms. According to [38], more than 4 affected video frames out of 1000 is unacceptable.

A common way to measure the amount of distortion of a signal due to different kinds of noise is the *Signal to Noise Ratio* (SNR). SNR is the ratio of the power in a signal to the power contained in the noise that is present at a particular point in the transmission. Typically, this ratio is measured at a receiver as it is at this point that an attempt is made to process the signal and eliminate the unwanted noise. For convenience, this ratio is often reported in decibels.

$$(SNR)_{dB} = 10\log\frac{signalpower}{noisepower} \quad \textbf{(EQ 1)}$$

In digital image coding the situation is somewhat more complicated since all the pixels can be thought of as representing a separate signal. Thus, computing a SNR for each of the pixels in, e.g. a ITU-R BT.601 NTSC frame, would result in 858*525 = 450 450 different values for each frame. What you would like to have is a single value describing the degree of distortion in the image. A common distortion measure is the *Mean Squared Error* (MSE).

$$MSE = \frac{1}{N}\sum_{k=0}^{N-1}(x_k - y_k)^2 \quad \textbf{(EQ 2)}$$

with which you can compute a *Peak-to-peak Signal to Noise Ratio* (PSNR)

$$(PSNR)_{dB} = 10\log\frac{x_p^{\ 2}}{MSE} \quad \textbf{(EQ 3)}$$

where x is the original, y is the received and x_p is the peak-to-peak value of the image data.

While these metrics are reasonable for measuring signal distortion they are very bad measures of perceptual quality. The SNR, for example may vary from a few dB up to more than 100 dB, mostly depending on the signal, while no noise is audible in any of these cases [39]. In [40] F. Bock, H. Walter and M. Wilde shows the failings of MSE.

5.3.1. Subjective opinion scores

Since the end-points of the communication are humans, sooner or later all measurement results must be mapped to subjective ratings of perceived quality. I have looked at two commonly used methods, called *Mean opinion score* (MOS) and the *Degradation mean opinion score* (DMOS) although several other scales are used as well as can be seen in [41, 42].

MOS is a method for obtaining subjective ratings of perceived quality on a five-grade scale ranging from 1, Bad, to 5, Excellent. The MOS value is extracted from the results of an *Absolute Category Rated* (ACR) test performed on 20 to 60 untrained persons.

The DMOS is a method for obtaining subjective ratings of perceived quality degradation compared to an original. DMOS uses a five-grade scale ranging from 1, Very annoying, to 5, Inaudible. The DMOS value is extracted from the results of an *Degradation Category Rated* (DCR) test performed on 20 to 60 untrained persons.

These subjective rating methods are commonly used to measure perceived audio quality, although the methods used are criticized for not being fair enough [42].

5.3.2. Perceptual model-based

Measuring subjective opinion scores tend to be both expensive and lengthy. The subjective nature of the tests also put high demands on test population selection, minimizing exposure time, and test material selection among other things to achieve a statistical significance and rule out unwanted interferences. To avoid all this fuzz, there are efforts to develop measurement tools that mimic human senses to produce some values according to *perceptual metrics,* that in turn can be mapped to subjective rating scores.

Their general operation is often based on a few key aspects of how human perception works. As we saw in section 3.1.1, the human vision is more sensitive to contrasts than to general signal strength, e.g. gaussian noise is acceptable to quite high levels while contiguous errors, such as block artifacts in video is perceived as very annoying. The same also applies to audio, where artificial tones is perceived as more disturbing than gaussian noise. The consequence is that humans are more sensitive to some artefacts than others.

All perceptual models of human vision that I have looked at exploited this *contrast sensitivity property.* Besides from gaussian noise and contiguous errors, there are a multitude of other artefacts, as we saw above. In a perceptual distortion measurement tool artefacts are assigned weights according to their visibility or audibility which is then used in combination with a signal distortion measure to produce an approximation of the perceptual distortion in the image.

In [40, 43, 44] you can find some recent developments in perceptual models and distortion measurement for video. Within the International Telecommunication Union (ITU-R), a task group (TG 10/4) is working on the development of an objective measurement system, based on perceptual models for audio testing.

6. Review of the State of the Art in videoconferencing

I haven't been able to find one specialized IP-based product comparable to the early H.320-based videoconferencing studios. Instead, the tendency seems to be to use some existing desktop videoconferencing tool on workstations for audio and video communication between rooms. Thus, even if the aim of this thesis is to concentrate on the room-based setting, it is necessary to look at the State of the Art in both desktop and room-based videoconferencing over IP-based networks. I also included some commonly used videoconferencing solutions using transport over circuit-switched networks (ISDN) as a comparison.

6.1. The MBone tools

The *Multicast Backbone* (MBone) was initially a virtual network backbone built on top of the unicast Internet using IP-in-IP tunnels bridging together multicast-enabled subnets. The reason for this was that at that time, 1991-1992, production Internet routers could not carry IP-multicast traffic. Thirty-two isolated multicast sites spread over four countries were connected and the network was, among other things, used to audiocast the 23rd *Internet Engineering Task Force* (IETF) meeting. The MBone still exists today, but is gradually replaced by IP-multicast support in Internet routers since late 1996. During the five years of intense research, the initial audio tool (vt) was complemented by several audio, video, whiteboard and session directory tools of which I will present the two most commonly used today.

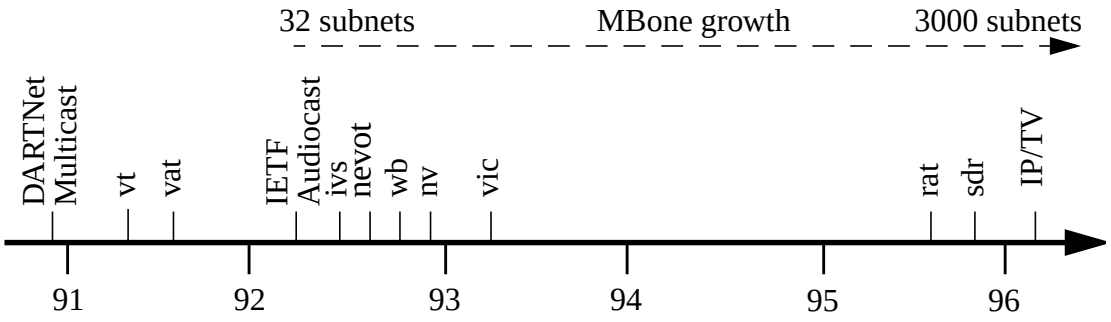

FIGURE 14. MBone Timeline [45].

6.1.1. VIC

The UCB/LBNL *VIdeoConference* tool (VIC) is one of the most successful desktop video tools for packet-based networks. Unlike many of it's predecessors, VIC is highly optimized for a typical Internet environment with lossy connections and low-end desktop computers. The source code is freely available and modular as described in [45]. This makes VIC a good platform for prototyping. The MASH research group at the University of California, Berkeley [46] is extending the TCL/C++ architecture of VIC into a multimedia networking toolkit called the *MASH shell*, or simply *mash*, while the *University College London* (UCL) is continuing the development of the original VIC, keeping a revision series of their own.

VIC was designed with a flexible and extensible architecture to support heterogeneous environments and configurations. For example, in high bandwidth settings, multi-megabit full-motion JPEG streams can be sourced using hardware assisted compression, while in low bandwidth environments like the Internet, aggressive low bit-rate coding can be carried out in software [47].

VIC uses RTP version 2 for video transport and for gathering awareness information and network statistics. To provide confidentiality to a session, VIC implements end-to-end encryption using the Data Encryption Standard (DES).

At writing time, the current version from UCL is version 4 [48]. Supported input video formats are *Square pixel SIF* (320x240 for NTSC feed, 384x284 for PAL) and CIF. It can also do X11 screen capture. The following codecs are supported:

- Software H263
- Software H263+
- Software and hardware JPEG
- Software BVC encoding
- Raw YUV packetizer/codec
- Software and hardware H.261 encoding
- Software nv and nvdct
- Software and hardware cellB

Audio-video synchronisation is supported with RAT version 3.2.

6.1.2. RAT

The *Robust-Audio Tool* (RAT) developed at UCL allows users to participate in audio conferences over the internet. Just as VIC, RAT is based on IETF standards, using RTP version 2 above UDP/IP as its transport protocol.

RAT features sender based loss mitigation mechanisms and receiver based audio repair techniques to compensate for packet loss, and load adaption in response to host performance. These features are a result of experiences using the previously available audio conferencing applications over the MBone for remote language teaching [49]. Over the years, more and more features have been added and the sound quality has improved significantly compared to the previous audio tools. The current version is 4.0 [48] and supports sampling rates of 8,16,32,48 kHz, mono and stereo, and can do sample format conversion with alternative quality/cost options. The codecs supported are

- Wide Band ADPCM Speech Codec (16kHz, 64 kbits/s),
- G726-2/3/4/5
- VDVI
- L16 (128 kbits/s, 16 bits linear)
- PCM (64 kbits/s, 8 bits mu-law)
- DVI (32 kbits/s, 16 bits differential)
- GSM (13.2 kbits/s)
- LPC (5.8 kbits/s)

RAT can also do 3D positioning of audio sources and support lip synchronization with VIC via a *conference bus*.

6.2. Other desktop videoconferencing tools

There are other IP-based videoconferencing tools that have developed in a different way from the MBone tools, being unicast-based and using either point-to-point meshes or reflector nodes for multipoint distribution.

6.2.1. Communique!

The MDL Communique! [8] is a desktop videoconferencing application using RTP version 1 over UDP/IP transport. It supports uncompressed L8 audio and a proprietary audio compression algorithm called Insoft1. Audio and video can be synchronized and both a rudimentary software echo cancellation and silence suppression is available. The documentation mentions support for IP multicast transport, but we haven't been able to make it work during the testing. Video codecs supported are CellB, JPEG, DVE2, Indeo and H.261. Which codecs one really can use differs somewhat depending on machine, operating system and audio-video hardware.

6.2.2. CU-SeeMe

CU-SeeMe was originally developed at Cornell University [9] to produce a similar functionality as the MBone tools on the Apple Macintosh platform. At that time it provided a 160x120 4-bit grayscale image. Later it was commersialized by White Pine Software and a MS Windows version was developed. CU-SeeMe uses a reflector node for multipoint distribution that has later been extended to interface with the MBone tools and applications based on the ITU-T recommendation H.323. The reflector node can use IP multicast for downstream traffic but the default configuration uses unicast both upstream and downstream. The MS Windows version of the CU-SeeMee client also supports conferences using IP multicast without any reflector involved. The WhitePine CU-SeeMe includes three software video codecs; MJPEG, H.263 and the original grayscale codec from Cornell. The MJPEG and H.263 codecs can handle 8, 16 or 24 bit colors and video of QSIF/QCIF size at a few frames per second. The audio codecs supported are G.723.1 (5.3 and 6.4 kbits/s), DigiTalk (8.5 kbits/s), Delta-Mod (16 kbits/s) and DVI (32 kbits/s) [50].

6.2.3. The DIT/UPM ISABEL

ISABEL was a one year project devoted to put together two different ATM developments: RECIBA from Telefonica in Spain and RIA from Telecom Portugal. ISABEL is also the name of a cooperative work application developed on top of the ISABEL network infrastructure to demonstrate new usage of the emerging ATM technology [7]. It was also used in the RACE and ACTS summerschools 1993-1997. The ISABEL application runs on a Sun Workstation equipped with a Parallax Xvideo capture board generating MJPEG encoded video ranging from QSIF to 4SIF formats. Audio formats used are 8 - 16 bit PCM, the 8 bit format can be compressed using GSM or G.721 codecs. It uses UDP/IP for transport. of voice and video and no loss recovery algorithms, so it needs a good quality link to work. A conference is initiated by a server to which additional clients may connect to participate in the conference. From the server, each client obtains the addresses of the other clients in the conference enabling them to set up and manage a mesh of point-to-point links. Thus ISABEL uses a mix of point-to-point mesh and reflector node technology to provide multipoint distribution. There is also a pure reflector-node-based version where each sender send to the server which in turn distributes the audio and video data using IP multicast [51]. Except from the audio and video it also incorporates a lot of X11-based shared collaboration tools.

6.3. The ITU-T H.32x visual telephones

The ITU-T H.32x series of recommendations cover visual telephony over different network characteristics. They are commonly described as *umbrella standards* since they utilize services defined in other standards.

TABLE 7. ITU-T Video Teleconferencing Standards

Network	Narrow-band	Low bit-rate	Guaran-teed QoS LAN	Non-QoS LAN	ATM	High Res ATM
Framework	H.320	H.324	H.322	H.323	H.321	H.310
Video	H.261	H.261 H.263	H.261	H.261 H.263	H.261	MPEG-2 H.261
Audio	G.711 G.728	G.723	G.711 G.722 G.723 G.728	G.711 G.722 G.728	G.711 G.728	MPEG-1 MPEG-2 G.7xx
Data	T.129	T.120 T.434 T.84 Others	T.120	T.120	T.120	T.120
Multiplex	H.221	H.223	H.221	H.22z	H.221	H.222.1 H.221
Signalling	H.230 H.242	H.245	H.230 H.242	H.230 H.245	H.230 H.242	H.245
Multipoint	H.243		H.243		H.243	
Encryption	H.233 H.234	H.233 H.234	H.233 H.234		H.233 H.234	

The H.323 uses RTP as media transport and TCP as control transport over IP-based networks. Below I will introduce the two most widespread products based on the ITU-T recommendations H.32x.

6.3.1. PictureTel

PictureTel has both desktop-based as well as group systems built on H.320 over ISDN and H.323 over TCP/IP. Video frame formats supported are H.261 at 15 to 30 fps CIF and 30 fps QCIF. Some products also supports H.263 CIF/QCIF and PictureTel's proprietary video formats SG3 and SG4 (256 x 240 pixels). Audio is recorded from two different bands: the 300 Hz - 3.4 kHz band for G.711 and G.728, and the 50 Hz - 7.0 kHz band for G.722, PT724, PT716 plus, SG3 and SG4.

The PictureTel 550 desktop product have up to 60db total echo cancellation and also supports the G.723.1 (GSM) audio codec.

6.3.2. Microsoft Netmeeting

MS Netmeeting uses H.323 and comes with G.711, G.723.1, and ADPCM software speech codecs. Some companies (like Lucent) make plug-in audio codecs for Net-Meeting. For video it uses a H.263 software codec getting video from any Video for Windows compliant hardware. It has support for bit-rate limitation to a few common network speeds: Modem (14.4 kbits/s), modem (28.8 kbits/s), ISDN (128 kbits/s) and LAN (unlimited, but often around 400 kbits/s). Supported frame formats are SQCIF, QCIF and CIF.

6.4. The CosmoNet

The CosmoNet [52] was built 1995-1996 as part of an effort to provide joint lectures between two lecture rooms at the department of *Electrical Engineering at Stanford University* in the USA (SU/EE) and the department of *Teleinformatics* at the *Royal Institute of Technology* in Sweden (KTH/IT). As such it was optimized for a video seminar situation and the video delay was not considered crucial in that context. The video format used was MJPEG compressed, 4:2:2 subsampled, Square PAL (768x576 active) from the KTH/IT and full Square NTSC (640x480 active) from the SU/EE. Unfortunately, the analog video feed was of S-VHS quality so it didn't utilize the high resolution very efficiently. The audio format was raw DAT mono audio, and the interleaved audio and video used UDP/IP transport.

The CosmoNet system consisted of two dedicated audio-video compression/decompression machines at each site connected via a 7.5 Mbps dedicated link that proved to be sufficient for a video compression ratio of just over 23:1. This system solution was forced by the low bit-rate of Ethernet at the time (10 Mbps) and the high compression ratio needed. An intercom channel using RAT over the Internet was used since the link technology proved to be quite unstable.

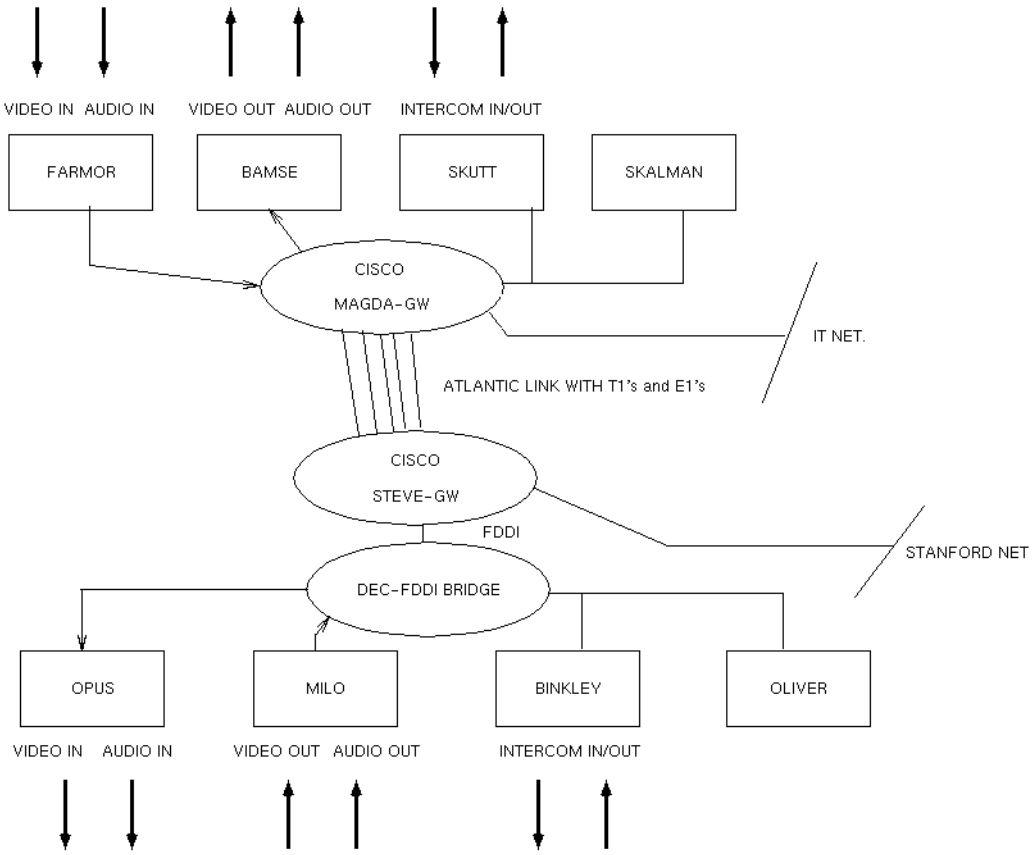

FIGURE 15. The CosmoNet network [52].

6.5. The WIDE DV Transmission System

The *DV Transmission System* (DVTS) [53] was implemented by Akimichi Ogawa in the WIDE project at Tokuda, Murai, Kusumoto & Nakamura Laboratory, Keio University at Shonan Fujisawa Campus, Japan. The prototype runs on a PC with FreeBSD operating system and a kernel patch and driver for a Texas Instruments TSB12LV21 chip-based IEEE 1394 (Firewire) PCI card.

DVTS is divided into a sender and a receiver part. The sender part grabs frames from the IEEE 1394 device and package them into RTP version 2. It currently uses a static maximum packet length of 512 bytes, but mr Ogawa intends to make this more general. On the receiver side it uses two different processes, one for writing to the IEEE 1394 device and one for receiving and processing the RTP v2 packets. The prototype uses UNIX shared memory inter-process signalling for moving the received DV frames between the two processes.

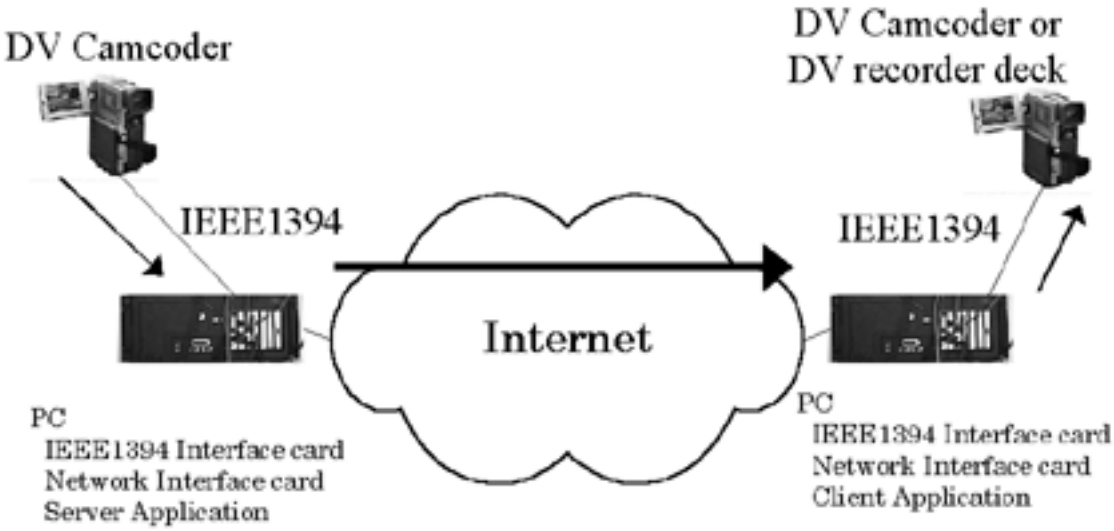

FIGURE 16. The DV Transmission System [54].

7. Research question

In the early nineties the packet-based *Local Area Networks* (LAN) technology was good enough to support videophone-quality video, i.e. a QCIF-sized view of the head and shoulders of the other person. Several computer-based collaboration prototypes and products utilizing video over LAN appeared in the first half of the nineties, but they never became more than funny toys that people stopped using after a while. Some identified reasons is that the quality was too low, the applications too cumbersome to start up and/or the average LAN was not large enough to motivate people to go through the fuss to use the applications instead of walking over and talk in person, or just using the phone instead. A few solutions were proposed:

1. Include application sharing to compete with telephone.
2. Use over *Wide Area Networks* (WAN) to compete with face-to-face meetings.

The first solution adds a function to the service that was not currently available via other media at the time and it actually made the designers of circuit-switched-based systems to start developing support for this in H.32x. However, this function doesn't explicitly need audio-video connections, but could as easily be used together with a telephone or a text-based chat application. This is why the ITU-T people later moved the work into a separate family of standards - the T.120.

The second solution actually promoted the use of room-to-room systems since the only WAN connections of good enough quality was circuit-switched, and the high cost of the links were motivated by pooling of resources. However, as we saw in chapter 6, even the circuit-switched networks-based room-to-room systems have a quite low resolution and is not suitable for casual meetings or short question-answer sessions that people are used to have in a typical office environment. In this paper I tackle the low resolution problem and try to find out how to implement a high quality, IP-based videoconferencing system.

As we saw in section 3.2, success with a new media is often dependent on whether the participants can use their existing norms. Thus, the expectations of the users probably are governed by the presentation media and the environment in which they are used. If a TV screen is used, then new users naturally will expect at least TV quality resolution and frame rate, and if a video projector or backlit screen is used then the users will expect the same quality as in the cinema theater.

Thus, in a room-to-room videoconference, the user already have a strong expectation of what quality to expect, because the environment is so similar to watching TV or a movie.

In a desktop videoconferencing system, the computer screen is the presentation media. The users of an ordinary desktop computer may or may not have some previous experience of what quality to expect from the system. If they do have some experience from using a desktop computer, the expectations might be based on experience of using other applications on the system, such as waiting several minutes for a web page to download. Thus the user expectations on image and sound quality provided by a desktop-based videoconference tool is lower.

Alas, with the currently fast proceeding transition into the information society more and more people will be using computers and with increased usage we will see demands on higher performance and reliability from the consumers. And there will be no special exception for video content. Some currently available DVD and other MPEG-2 decoder boards offer realtime decoding and fullscreen play of up to widescreen TV resolution video. As these applications become more commonplace the users will start to expect similar quality, robustness and ease of use of the desktop videoconferencing tools as well.

In this work I concentrate on the case where a TV set is used as presentation medium. I will show ways to build a videoconference system offering the same audio and video quality as people are used to have from broadcast and cable television media. Apart from the audio and video quality “normally” provided by the presentation medium I also have to consider issues related to the interactive nature of videoconferences and human conversation and communication in general. Anyone who have placed a very long distance phone call over a bad line or a satellite link knows how difficult it is to hold a normal conversation in a medium with a long end-to-end delay. Thus, since a videoconference is intended to support a distributed meeting, where conversation is an important part, delay-related factors have to be considered.

8. Ways to meet the user requirements of room-based videoconferencing

In this section I will investigate ways to implement a system that provides TV quality video and audio and at the same time ensures that the rules of human interaction are not violated by delay-related factors. To be able to know when we have a system that offers a sufficient end-to-end quality we need to translate the fuzzy notion of TV quality into measurable parameters. This is covered in section 8.1.

The rest of this chapter is dedicated to discuss key components in a videoconferencing system and their influence on the parameters from section 8.1.We'll look into the choice of compression scheme, computer system design and network technology as well as the trade-offs in design regarding different system resources, including some common solutions and their bottlenecks.

The concept of an end-system is introduced and is used to show how the components fit together. We'll also see some examples of how a careful end-system design can alleviate bottlenecks in the components.

All in all this chapter should cover both fundamental issues in the design of videoconferencing systems and a fair overview of the state of the art in computer support for audio-video mediation.

8.1. End-to-end quality parameterization

A straightforward way to translate perceptual quality into measurable parameters is to conduct a series of user studies using one of the subjective opinion scores presented in section 5.3.1, but they tend to be lengthy and costly and the results may or may not be useful. Besides, a lot of work has been done in the areas of television and telephony as well as CSCW, so I decided not to do any user studies myself but instead consult the literature in those areas and collect commonly used parameters that I later can translate into a videoconference system requirements specification.

8.1.1. Audio quality

The quality provided by the NICAM format (section 5.1.1.) should be the minimum needed to compete with existing TV broadcasts, but it's not the best quality that can be achieved and it might be overkill for the purpose of videoconferencing.

In a typical meeting, most of the audio communication consists of speech. Therefore it might seem reasonable to use a specialized encoding scheme for speech, such as the ITU-T G.700-series presented in section 5.1.3. I'm sure the remote participants in a videoconference can live without a perfect reproduction of the screeching of a dry pen on a whiteboard or moving chairs, but on the other hand, who knows what new gadgets will be available in the meeting rooms in the future? In the best seminars that I've been attending the speaker used video clips and audio effects other than speech.

Another thing that speaks against using speech coders is that they don't support multi-dimensional sound. The NICAM format uses two channels, and as mentioned in section 3.1.2. three-dimensional sound helps to give situation awareness and to discriminate between participants in a meeting. To achieve the impression of space, one can use multiple audio channels recorded by microphones located along a few known axes and then transform the audio data for playout on speakers located along other axes. Two to four channels are common in such settings.

I decided to go for full CD or DAT quality, i.e. recording a little more than a human can hear in a stereo channel.

8.1.2. Video quality

As shown in section 5.2.1, NTSC uses has about 484 active lines visible on the screen. Experiments on ordinary TV screens have shown that the actual subjective resolution is about 338 lines, that with an aspect ratio of 4:3 would give a horizontal resolution of about 450 lines. Thus to provide TV quality video on an ordinary TV set it would suffice to have a resolution of 450x338 pixels.

However, since all the video compression schemes expect an ITU-R BT.601 sample stream I find it more practical to choose the active part of the sample frame delivered by this format, i.e. 720x484 for NTSC and 720x575 for PAL/SECAM feed. Thus, depending on the quality of the analog feed this can result in a higher resolution than broadcast TV. ITU-R BT.601 also retains the interlaced field sequence of the feed and the same frame rate as for the TV standards, i.e. 29.97 for NTSC and 25 for PAL/SECAM. Note that this framerate is only suitable for an interlaced feed. Tests have shown that 30 complete scans per second will produce a sensation of flicker rather than smooth motion. With interlacing the screen is refreshed at 60 times per second, and flicker is avoided. Thus, for a progressively scanned feed a frame rate of 60 or 50 Hz is preferable.

A subsampling of 4:2:2 is often said to cause no noticeable degradation in the quality of the image, and from my own subjective experiences I have to agree. While a further subsampling to 4:1:1 or 4:2:0 gives a somewhat bleak impression in comparison, I found it acceptable as well. I wouldn't recommend further subsampling, though. As we saw in section 5.3, the number of frames suffering from visible artefacts generally should be lower than 1 in a few thousand frames.

8.1.3. Delay-related quality aspects

From section 3.3.2 we know that human conversation is very sensitive to delay, delay variation (jitter), echo effects and audio-video synchronization.

The end-to-end one-way delay affects turn-taking in a conversation. Adding a fixed round trip delay longer than a normal breathing pause naturally will turn the conversation into a confused series of interruptions and in time will force a very formal and tedious turn-taking procedure. Body language, such as facial expressions, gestures, posture, mutual glances, gaze, et.c. used to invite someone else to take the floor is sensitive to delay in the same way as pauses and other audio cues. Participants in a videoconference suffering a significant round-trip video delay tend to compensate through acting excessively clear, as if the remote participants were small children or complete dimwits.

In the ITU-T recommendation G.114 [41] a concept called end-to-end transmission time is discussed. Since in the world of traditional telecommunication, the terminal equipment (telephones) doesn't introduce any significant delay I will reuse their findings as constraints on *end-to-end delay*, i.e. the time between sampling on one site to display/playout on another site.

In G.114 is stated that a 0 to 150 ms delay is acceptable for most user applications. 150 to 400 ms is acceptable provided that the administrations are aware of the impact on the quality of user applications. International connections with satellite hops is mentioned as one probable cause for such delay. Above 400 ms may be acceptable in some exceptional cases. Probable causes are double satellite hops, videotelephony over satellite or temporary fixes. Thus it seems that delays up to 400 ms should be bearable. Other sources are more restrictive though. Karlsson et al states in [55] that the video and sound may be delayed 150 ms without causing much disturbance. Le Gall states in [30] that applications such as videotelephony need to have a total system delay under 150 ms in order to maintain the conversational "face-to-face" nature of the application. A choice of maximum 150 ms end-to-end delay seems fairly safe.

Delay variation, or delay jitter, affects the intelligibility of speech, rythm of music and causes jerky movement in video. The *Inter-frame Display Time* (IDT) should be constant with very low variance (*jitter*) to maintain smooth media playback. In a study to find the acceptable amount of jitter in a 30 seconds long audio recording reported on in Appendix C showed a relatively strong mean of 35 ms, while some sources in the biliography recommends up to 100 ms.

By using *delay equalization* one can reduce the delay variation of an audio or video traffic stream by buffering incoming data and regenerate the IDT at the receiver. The buffer can be of fixed size, or can adjust it's size to current jitter measurements, acceptable late loss, one-way trip time estimations, acceptable total loss, and the mean packet size so far. Thus one can trade increased delay for reduced delay variation up to the maximum acceptable end-to-end delay if a delay variation of more than 100 ms is encountered.

Echoes of a speakers voice is distracting for both the speaker and the audience. The ITU-T has defined 24 ms as the upper limit of the one-way transit delay beyond which additional echo canceling techniques have to be employed. Typically, the longer the delay the more distracting the echo will be.

As discussed in section 3.4, audio-video synchronization affects the feeling of presence and ease of conversation that is important in a meeting. According to [55] the video may antecede the sound by up to 100 ms or succeed it by less than 10 ms

8.1.4. Summary

To sum up the results of the previous sections, I have found a set of parameters that I believe quantify the end-to-end audio and video quality that we need to provide. Now we have a fair idea about the end-to-end quality that have to be supported by a videoconferencing system. Using the parameters obtained in the previous section it is possible to test if a certain system is capable of delivering a sufficient quality end to end.

Examining the parameters we see that none of the State of the Art videoconferencing tools presented in chapter 6 can deliver the chosen video resolution while several of the video compression schemes in section 5.2 support this resolution. On the audio side, a recent version of RAT should be capable of delivering the chosen audio resolution. As for the delay-related quality aspects I have no data for any of the state-of-the-art videoconferensing tools.

TABLE 8. Parameter values for slightly better than TV perceived quality

Parameter	Value
Audio bandwidth	20 Hz to 20 kHz
Audio channels	2
Video resolution, luminance	720x484 or 720x575 samples
Video resolution, color differences	4:2:2 - 360x484 or 360x575 samples 4:1:1/4:2:0 - 180x480 or 180x575 samples
Video frame rate	29.97 or 25 fps interlaced, 60 or 50 fps progressive
Video sample size	8 bits each for Y, Cb, Cr before subsampling
Video aspect ratio	4:3
End-to-end delay	150 ms maximum
End-to-end delay variation	100 ms maximum
Video antecede sound	100 ms maximum
Video succeed sound	10 ms maximum
Echo cancellation needed	> 24 ms end-to-end delay
Video artefact occurance	less than 1 per every few thousand frames

8.2. The choice of compression scheme

The raw sample streams of the audio-video resolution chosen in the previous chapter are seldom feasible to transmit directly. To reduce the data rate one uses compression schemes such as the ones presented in sections 5.1 and 5.2. The choice of compression scheme significantly influence the traffic characteristics but it also influences other parameters such as delay and delay variation and often introduce some audio or video data loss, causing signal distortion.

8.2.1. The expected total bit-rate

The main motivation for using compression is to reduce the bit-rate of audio-video material to more manageable levels. As we saw in sections 5.1.4 to 5.1.7, one can use a more or less complex coding scheme that produces CD or DAT quality audio at 1.5 M or fewer bits per second. E.g. the MPEG-1 Layer III is one of the most complex schemes producing a near CD-quality sound at 64 kbits/s per channel and DVI-audio produce stereo broadcasting quality at the same bit rate. If only software codecs are available, then the 1.4 to 1.5 Mbps sample stream has to be transferred through the system at least once, while hardware audio codecs usually is co-located with the audio A/D hardware and thus the system only has to handle the compressed stream.

As for video, the raw ITU-R BT.601 stream has a bit rate of at least 270 Mbps. However, since a lot of the bits is not picture content, I decided on using only the active part of the video content. Still, NTSC video, with a frame size of 720x484 samples, a frame rate of 29.97 frames per second and a 4:2:2 subsampling scheme would produce a bit-rate of ca 167 Mbps while a PAL video with a frame size of 720x575 samples, a frame rate of 25 frames per second and a 4:2:2 subsampling scheme would produce a bit-rate of ca 165 Mbps. Further subsampling to 4:1:1 or 4:2:0 reduces the bit rate to 125 Mbps for NTSC and 124 Mbps for PAL.

As in the audio case, there's a multitude of different compression schemes to further reduce the bit rate. Most of them requires a specific subsampling of ITU-R BT.601 input to the compressor part and the corresponding decompressor returns the same format at the receiving end. A bit-rate reduction of around 40:1 can be achieved by the most complex asymmetric codecs, such as MPEG-2, while still retaining the same subjective quality, as perceived by the human eye, as the original sample stream. For symmetric codecs, such as MJPEG, a bit-rate reduction of around 25:1 is common for the same quality.

As we'll see in section 8.3.2, the most of the video capture hardware also include some hardware compression as well, while on the receiver side ordinary graphics hardware seldom support decompression. Thus, depending on the hardware configuration, zero or more uncompressed video streams will be transferred through the system in addition to the compressed stream.

What kind of uncompressed format will be used also depends on the graphics rendering hardware and presentation device, e.g. RGB is most common for computer screen graphics rendering boards while 4:2:2 subsampled YCbCr format is the most common feed to analog video rendering hardware.

8.2.2. Burstiness

The sample stream generated by the digitizer is more or less *Constant Bit Rate* (CBR). A constant bit rate makes it easier to reserve resources in a network as well as dimension a computer architecture for audio and video data streaming. In the case of a constant quality video codec, also called *Variable Bit Rate* (VBR) codec, we have to use resource reservation based on a worst-case analysis to get an upper bound for how much resources might be needed, or we have to be prepared to employ some techniques to help repairing damaged data [56]. Thus the burstiness of the compressed data is an issue that has to be taken into account when designing a videoconferencing system.

Most audio codecs produce a more or less constant stream of bits, sometimes packaged into frames of a few tens of milliseconds worth of samples by the encoder. Some applications, e.g. RAT and Communique!, take advantage of the fact that, during speech, sequences of phonemes follow silent periods. As we saw in section 3.1.2, typically 60% of the speech is silence, so by only sending incoming audio which is louder than a cer-

tain threshold you can save up to 60% bandwidth. This technique is called *silence suppression.* A byproduct of this "compression scheme" is that the generated bitstream becomes bursty. Also, the receiving side has to generate some kind of audio data to fill in the silence because the audio stream of samples that is sent to the audio card have to be constant with a bounded jitter [49]. I haven't seen any audio hardware that accepts a silence-suppressed (bursty) stream.

The ITU-R BT.601 produces a constant sample stream irrespective of which subsampling scheme is used. There are also some compression schemes that includes *bit-rate regulation* algorithms to keep the produced bit-rate at a certain value, normally at the expense of varying picture quality. Other compression schemes are designed for a constant image quality - then at the expense of a freely varying bit-rate [56]. The burstiness ratio of such a constant-image-quality-compressed video bitstream may reach 10 to 1.

8.2.3. Delay and delay variation

There are two main ways that compression schemes contributes to the end-to-end delay; due to complex computations and due to buffering.

The more complex codecs, using a combination of algorithms, often are not capable of real-time encoding in software. I.e. compression of one frame takes longer time than the frame rate of the feed. For audio, the complexity of codecs and data size rarely is any problem for today's computers to handle in real-time, but for video the situation is different. For video, non-real-time means that more than 33 ms delay is introduced by an encoder given a NTSC feed, or more than 40 ms if the feed is PAL. When this occurs, the codec must skip incoming frames to avoid running out of memory and the frame rate drops.

The H.263 is one example of video compression schemes that is too complex for real-time encoding in software. In [28] is shown that a software H.263v2 encoder running on an Intel Pentium 200MHz machine can at most encode 13 fps of QCIF video at 64 kbits/s. Furthermore, in [57] a group at Lucent Technologies presents their implementation of a software layered H.263v2-compatible encoder that can encode up to three video layers of QCIF format at 7.5 fps on an Intel Pentium 200MHz machine. Although the machines used in these two studies are slow by current standards, note that the QCIF is about one sixteenth of the video resolution that was chosen in section 8.1.

MPEG is another example of a non-real-time encoding scheme. For the development of the MPEG compression algorithms the consideration of the capabilities of "state of the art" (VLSI) technology foreseen for the lifecycle of the standards was most important [30]. Thus to be able to encode and decode MPEG or H.263 streams at anywhere near full frame size and frame rate you have to rely on hardware acceleration.

The same seems to apply also for the codecs used in VIC, as shown in Appendix B, even though the software and hardware codecs used in that test operate on a quarter of the frame size that was chosen in section 8.1.

The second type of delay contribution is caused by differential encoders, where the encoding of new frames depends on the contents of previous frames, e.g. in MPEG interframe coding between up to twelve sequential frames is common, which means that a whole sequence of 12 frames has to be buffered while the frames in the sequence are coded. This gives a minimal delay contribution equal to the sequence length times the frame interval.

Some other encoders, like the PCM audio encoder, does not divide into frames itself. Still, the small size of each sample makes it a good idea to collect a few samples and pack them together into a frame of suitable length. Some PCM codec vendors use 10 ms and others use 20 ms as frame length, while 25 ms was considered too long since the gaps in the audio caused by packet loss was too noticeable [49].

Delay variation often occur in algorithms where the computational demand changes depending on the signal characteristics, e.g. in conditional replenishment only video blocks containing more than a certain amount of motion are coded. However, most of the end-to-end delay variation is introduced by other parts of a videoconferencing system, such as the network and operating systems.

Bit-rate regulation implementations may include a buffer at the end of the encoder to take care of variations that are too small and/or too fast to be handled by other mechanisms, such as varying the quantization threshold et.c. This means an additional end-to-end delay and increased end-to-end delay variation. However, this delay and delay variation is generally insignificant if the bit-rate regulation is done in hardware.

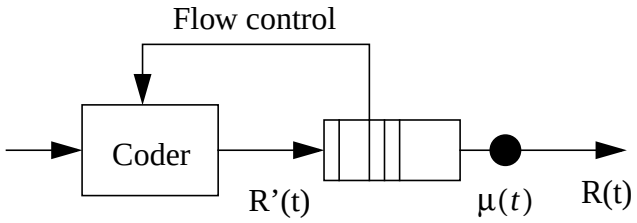

FIGURE 17. Bit-rate control system (from [56]).

8.2.4. Compression and signal distortion

The compression schemes introduced in sections 5.1 and 5.2 are all lossy, in the sense that they discard data that is considered redundant according to some algorithm. As we saw in section 5.3, sometimes they throw away too much data, or distort the data in a way that is noticeable - creating artefacts.

Another, less obvious, side-effect is that by throwing away redundancy a data loss of a certain size during transport affects a larger part of the audio signal or video frame when it is decompressed at the receiving side than if no redundancy had been removed. On the other hand, increased load on the network and hosts may increase the loss rate, so there is a tradeoff to be aware of here.

8.2.5. Fault tolerance

In most complex compression schemes the encoder and decoder are state-machines which must keep in step with the other with respect to the encoded data. If they get out-of-synch, e.g. because of packet loss or bit-errors in the network, the result is unpredictable and often result in audible or visible artefacts. To repair this faulty condition, the state of the encoder are sent by regular intervals, so called resynchronization points, in the code. These are often located on frame boundaries, but sometimes the network conditions demands shorter intervals.

Loss resilient coding, such as FEC, and loss repair strategies, such as reusing parts of previously received video frames, may also be neccessary to be able to recreate a smooth playout at the receiving end in the presence of data loss.

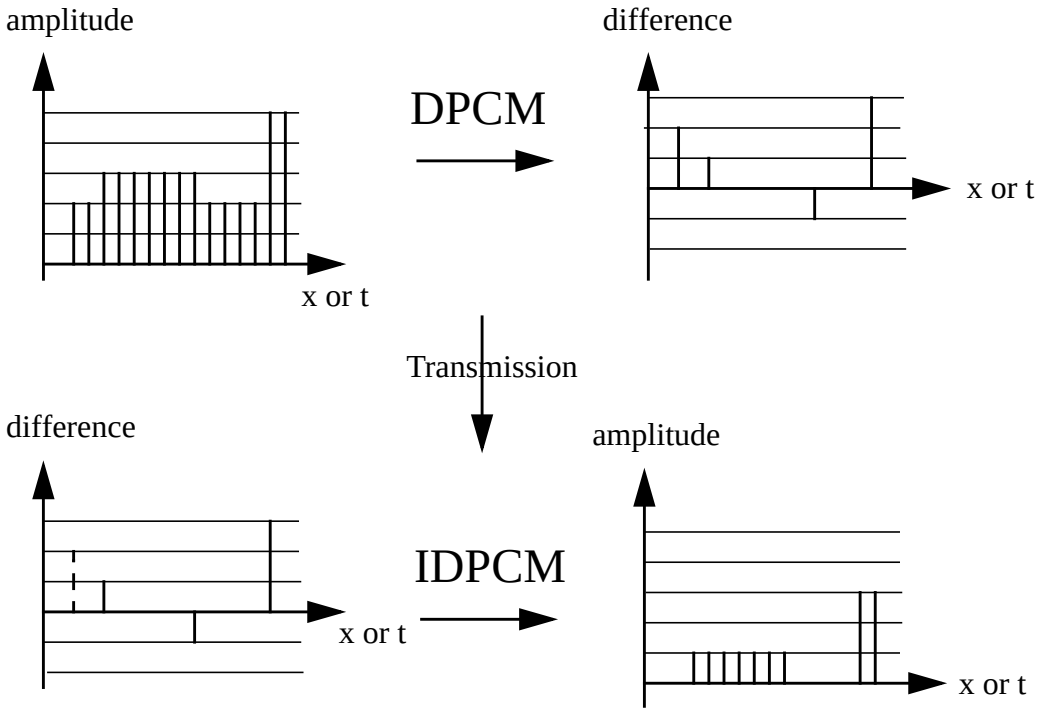

FIGURE 18. Data loss in a DPCM compression scheme.

8.2.6. Summary

Summarily, we have now a fair idea of the average traffic charateristics of audio and video streams with a resolution corresponding to the findings in section 8.1. A single audio stream may have an average bit-rate of 64 kbits/s to 1.5 Mbps and a burstiness ratio of up to 2.5:1 depending on compression scheme and the level of compression. The video stream bit-rate ranges from 167 Mbps for the active part of a 4:2:2 subsampled ITU-R BT.601 NTSC stream to around 4 Mbps for the most advanced video compression schemes. The burstiness ratio ranges from 1:1 for a raw sample or rate-limited stream to 10:1 for a constant quality video codec.

A video compression scheme may introduce up to several video frames length of delay while audio codecs seldom uses frame lengths longer than 20 ms. Delay variation due to compression is usually small compared to the delay variation introduced by the network and operating system.

To achieve the video quality given in section 8.1 using a modern compression scheme it seems necessary to use hardware while the audio can be compressed in software if needed. The compression ratio has to be carefully selected to avoid artefacts. The fault tolerance of a compression scheme is mainly a function of the length of the resynchronization interval.

8.3. The choice of computer platform

The traffic characteristics presented in the preceding section is for a single audio or video data stream. In a videoconference the computers used may have to handle multiple such streams. Also, the data may have to be copied several times between different parts inside the computer. How many streams the platform have to handle internally depends on

- the computer’s hardware architecture,
- the software running on the computer,
- the type and location of audio and video hardware and
- the number of incoming and outgoing streams.

The number of incoming and outgoing streams depends on the end-system architecture which will be discussed in section 8.5 below. Most of the State of the Art, IP-based videoconferencing applications that I have looked into build upon standard computer hardware and software available in 1995-1996. Since then, not much have happened in the further development of the applications. Unfortunately, this applies also for the support for real-time interactive audio-video handling in most commonly available computer configurations today. Fortunately, there are some exceptions as we will see in this section.

8.3.1. Hardware architecture

From section 4.1 we know some of the drawbacks and solutions of two common computer system architectures, and that the placement of key devices, such as video I/O, processing, buffers, display- and network interface, play an important role in the video handling capacity of a system.

The worst case is to have a computer system equipped with non-interoperable peripheral components in a bridge-based architecture that has to shuffle data through the CPU and main memory each step of the way. The best case would be to follow an optimized path through a shared memory architecture in which case only one copy of the data would be kept in main memory and a pointer would be passed to the different devices in turn. The best-case architecture could be implemented as a whole system architecture as in the SGI O2 or as a stand-alone subsystem consisting of one or more peripheral devices communicating directly over a private bus in a PC. Whichever soulution we choose, in the end we'd get a highly specialized hardware configuration that might become expensive and difficult to extend with new features as they become available. .

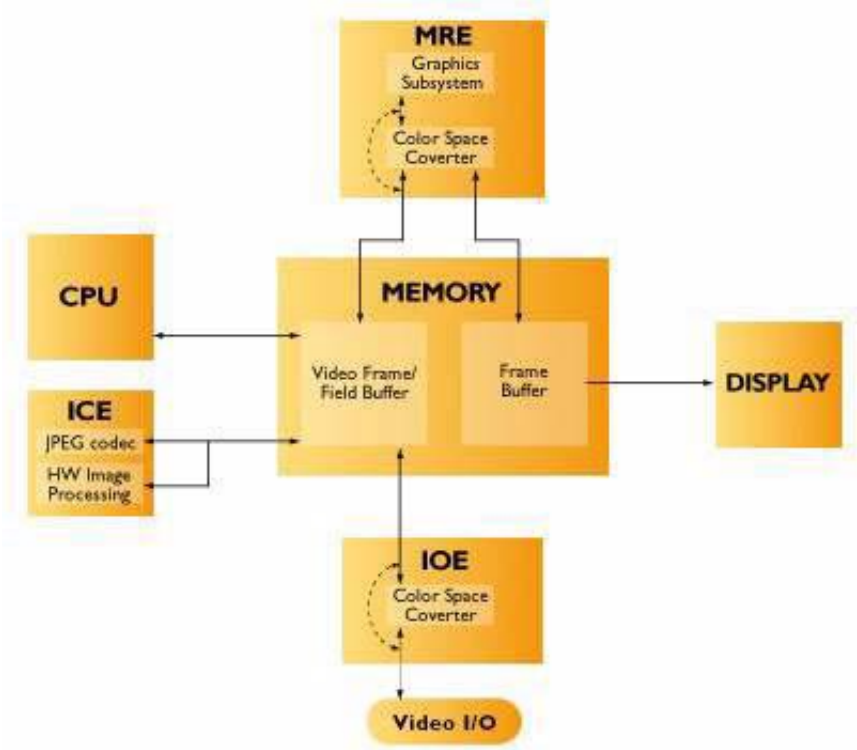

FIGURE 19. Path options for a video frame in the SGI O2 system [3].

The recent developments in digital home video equipment may provide a cheaper solution in the future. In a way, the digital home video equipment can be seen as an external capture/display and compression/decompression subsystem connected with the computer via a private bus.

The bus commonly used for interfacing computers and digital home video equipment today is the *IEEE 1394 Standard For a High Performance Serial Bus*, also called *FireWire*. IEEE 1394 interfaces will probably become a standard component of desktop computers and home PCs, because it has been recommended in SIPC, PC97 and PC98 [4]

8.3.2. Audio and video hardware

One fundamental limitation of desktop videoconferencing is the one about technical requirements. The minimum is to install an audio-video client software module, but even with todays computers a software module is only sufficient for low frame rate and low-resolution quality. To receive higher quality and to be able to send as well, the computer needs to be equipped with specialized hardware.

Most computers today comes with hardware for handling 8 bit or higher resolution linear audio since audio is an important part of the user interaction in a multitude of applications ranging from window systems to computer games. Choosing audio hardware is quite straightforward so I will not discuss that any further. The only thing you have to watch out for is hardware support for full-duplex audio, since half-duplex audio cards are still quite common. Half-duplex audio prevents interruptions and worsen the effect of delay jitter.

Most vendors today provide some analog-video capture hardware as an option. The products often also support hardware acceleration for some compression schemes, typically JPEG. Co-locating capture and compression is a good idea since you avoid having to move the uncompressed sample stream through the system to the compressor. I have looked at the following add-on products:

- miroVIDEO DC20/DC30 for PC,
- SunVideo 2 and SunVideo Plus for Sun,
- SGI A/V option.

The digitizing hardware in analog capture cards normally can deliver a part of the frame and some have automatic active video sensing. The SunVideo 2 for example, returns a window, 768 samples wide and 576 samples high which gives some margin around the active part of the PAL frame. For NTSC this window has the dimensions 640x480. The window can also be moved vertically to capture the remaining active lines [58]. The SGI A/V module has built-in circuits that can deliver the full active video part in either interlaced or progressive mode, using many different subsampling schemes and different pixel packings of both YCbCr and RGB color maps.

Unfortunately, most of the hardware support for analog video capture that I have seen for different platforms have been optimized for less frame size than I want to provide. The Quarter-PAL and SIF frame formats are the most commonly supported in hardware. To take SunVideo 2 as an example again, it can only deliver full frame rate for QPAL and QNTSC frame sizes of both uncompressed and compressed video.

There are also different digital video solutions available where the camera digitizes the video and use a digital interface directly connected to the computer. Examples are:

- SGI cameras (O2 camera, Indycam) using some proprietary interface.
- QuickCam using the parallell port of a PC or Macintosh.
- *Universal Serial Bus* (USB) cameras.

The CPU load of the receiving desktop computer can be alleviated by the use of an outboard decompression card. Then the CPU no longer acts as the limiting factor to the displayed frame rate. The drawback with this solution is that uncompressed video has to be transferred somehow from the decompression card to the graphics card. Some outboard decompression cards also supports compression as well, such as the SGI Cosmo compress, Indyvideo and ICE.

Something very similar to the digital video cameras is the recent development in consumer electronics for home video equipment where compressed digital video can be transferred to or from a video camera or VCR to a computer via a IEEE 1394 serial bus. In this case both capture and compression is taken care of by the camera at recording time. Products that I have looked at using this technology are:

- miroVIDEO DV300 for PC,
- SGI DVLink.

Most computers comes with graphics hardware capable of displaying at least thumbnail-sized video at full frame rate or bigger at reduced frame rate. As mentioned above, most analog video capture cards have on-board compression hardware. The same reasoning should apply on the receiver side as well. You should strive to keep the compressed format as far as possible, preferably all the way to the graphics rendering board to avoid having to shuffle uncompressed video through the system. However, most ordinary graphics rendering boards used today doesn't include decompression hardware. A change is on the horizon, though, with the relatively cheap MPEG and DVD playback boards, equipped with an on-board display memory and a VGA port. Another solution is to use an IEEE 1394 interface and leave the the decompression and display operations to a digital camera or VCR.

8.3.3. Operating system

As we saw in section 4.2, it is the operating systems that handle real-time data such as audio and video. Therefore I have looked into the real-time capabilities of a few common operating systems.

UNIX is an old time-sharing OS trying to provide a "fair" use of the system resources. Thus, preemption or advance allocation of resources is not possible. However, most of the current OS have some real-time support in addition to the basic UNIX functionality. As an example, SUN OS does not yet provide a RTE although Hagsand and Sjödin proposed one in paper C in [59]. Both IBM AIX and SGI IRIX includes a RTE.

The most commonly OS run on a Microsoft/Intel BIOS architecture is the *MS Windows* that doesn't have any real-time properties, but it does provide an enhancement to the Windows programming environment to handle audio and video files, devices and playback.

8.3.4. Summary

In this section I have discussed three possible platform configurations that should be able to handle the bit rate and real-time constraints of the audio and video part of a videoconference. In the bridge-based architecture a suite of interoperable peripheral devices is the preferred solution. Unfortunately, at writing time, I am not aware of any such product offering the audio and video quality outlined in section 8.1, although recent announcements from C-Cube Microsystems Inc seems promising. In the shared-memory architecture, I have one system that I have tried, i.e. the SGI O2, and it seems that it is equally good for any of the video capture and output modes. The third solution is independent of the computer system architecture in that codec, capture and display functions are "outsourced" to digital home video equipment connected via IEEE 1394.

8.4. The choice of network technology

One possible conclusion drawn from the demands of a real-time service like videoconferencing may be that some kind of *synchronous data* transport is needed where channels offering a certain *Quality of Service* (QoS) can be reserved. On the other hand, audio and video is a *soft* real-time service, i.e. a small amount of data can be late or lost without catastrophic consequences. Thus, a non-guaranteed QoS transport can be used if one takes care of regenerating the ordering and timing of data at the receiver, e.g. by using a protocol such as RTP.

The Internet Protocol (IP) is a non-guaranteed QoS network protocol, although there are a lot of different efforts to provide QoS guarantees over IP as seen in [60]. Using IP gives a high freedom of choice between different underlying technologies, QoS support, multipoint distribution techniques and techniques for supporting real-time data transfer.

8.4.1. Delay and delay variation over the Internet

The dedicated link between KTH/IT and SU/EE had a mean *Round-Trip-Time* (RTT) of min 170 ms, mean 178 ms, max 332 ms in September 1999. Over Internet between KTH/IT and *www.stanford.edu* in September 1999 showed a RTT of min 164 ms, mean 166 ms, max 278 ms. In these measurements I used ping with the default payload 64 bytes/second run over a period of 3 hours. Thus, it is far from representative for a typical video feed, but should be enough to show the order of magnitude of the end-to-end delay over WAN even for such data.

We see that the end-to-end delay is fairly stable around 85 ms, thus leaving 65 ms for delay in end-systems. This supports the average audio frame size of 20-30 ms in applications sending audio over the Internet, leaving some time for delay equalization at the receiver side while still not breaking the delay budget of 150 ms chosen in section 8.1.

Delay variation is more common in networks utilizing statistical multiplexing without quality guarantees, such as the best-effort Internet, than in networks using deterministic multiplexing with fixed quality guarantees, such as TDM [56]. Physical jitter over long-distance circuits is typically in the order of microseconds. Over high-speed optical-based technologies a value of about 6 nanoseconds is targeted. Thus the heavy delay variations that can be observed on the Internet is mainly protocol induced, e.g. back-off timers, and queueing in buffers in the hosts and network nodes.

I have heard figures of more than 200 ms delay variation although the measurement above only showed on average 5 ms delay variation with some high spikes. The tests on the dedicated link measured a 16 ms mean delay variation, also with some isolated spikes. Comparing these figures with the ideal parameter values of section 8.1 we see that the delay variation may not be a problem if we can accept some spikes of late loss.

8.4.2. The art of packetization

Real-time data, such as audio and video suffer from two common types of network-related loss; late loss caused by delay, and loss due to router and network downtime. For video there is also another potential source of loss - fragmentation.

A video frame can be quite big. The active part of an ITU-R BT.601 sampled NTSC frame is more than 2.5 MB, while the maximum UDP packet size is 8 KB. Using the packet size of UDP is not recommended, however, because it will trigger IPv4 fragmentation on most link technologies. In IPv4 fragmentation a lost fragment makes the receiving host discard the other fragments even if they could have been useful to the application. Besides, the video frames have to be fragmented anyway since video with the resolution that was chosen in section 8.1 cannot be compressed to fit in a single UDP packet with today's compression schemes. Thus, each video frame should be split into fragments that fit into link-layer units, if possible.

Also, by taking into account the syntax and semantics of the video stream one can fragment the data in an intelligent way and minimize the damage caused by a lost packet. The concept of *Application Level Framing* (ALF) lets the application deal with data loss according to its needs and capabilities. In RTP, the data is transported in *Application Data Units* (ADU) that contains data that can be processed out-of-order with respect to other ADUs [19]. Thus, using ALF we don't have to throw away a whole video frame just because one fragment got lost on the way.

As an example, the RTP payload for MPEG [61] utilizes the resynchronization points in MPEG [30] to form ADUs that can be decoded independently:

- To handle errors the predictors are frequently reset and each intra and predicted picture is segmented into *slices* where state is reset.
- To support “tuning in” in the middle of a bit stream, frequent repetitions of the coding context (Video Sequence Layer) is supported.

This demands that the MPEG encoder and RTP packetizer have to work closely together to insert resynchronization points on packet boundaries. Such a combined encoder and packetizer is not always possible, e.g. when receiving MPEG data over IEEE 1394 from a hardware encoder. In this case the packetizer has to parse the MPEG stream and insert the appropriate resynchronization codes at the appropriate places, introducing extra computational overhead in the process. The same problem applies for a few other compression schemes as well.

Another important issue to consider is the protocol overhead. The network protocol overhead for RTP/UDP/IP is the same as for TCP/IP, i.e. 40 bytes. Thus, it’s quite a lot of transmission capacity that is used for overhead. Especially in the case of audio where the bit-rate is often very low. In section 8.2.1 we saw that the audio bitrate is expected to be between 64 kbits/s to 1.5 Mbps. Using the recommended framing of 20 ms for audio in [17] we se that an audio frame will be between 160 to 3750 bytes giving a maximal protocol overhead of 20%.

For IP telephony, using the ITU-T G.723.1 speech codec at a bit rate of 5.3 kbits/s and audio frame size of 30 ms, gives a payload size of 20 bytes including padding bits gives a stunning 200% protocol overhead.

8.4.3. Summary

It seems that IP-based networks should be able to keep the delay and delay variation budget of section 8.1 although sporadic late loss is to be expected, due to spikes in the delay. In addition to this late loss, the data will suffer from loss introduced in the network. Careful packetization can ensure that this loss will not multiply in higher-layer protocols and applications at the receiver, and it can also facilitate delay equalization and loss repair mechanisms to further reduce the effects of data loss.

8.5. End system architectures

With *end-system* I mean the computers, the network connection, and audio-video equipment used at one of the participating sites in a videoconference to mediate audio and video to and from the other sites the videoconference. Although, the room design is important, it’s not part of the end-system. To distinguish between different system solutions I look at the path of the different data flows through the system.

The main *data flows* in an end-system can be roughly divided into four categories; incoming audio, outgoing audio, incoming video and outgoing video. The computers, network connections and the audio-video equipment in the end-system can be seen as possible multiplexing points for the different data flows. Using these definitions, different end-system architectures can be ordered by the degree of separation between the different data flows and where the multiplexing points are located.

In the state-of-the-art videoconferencing applications of chapter 6, there is a wide variety of hardware architectures ranging from monolithic H.320 roll-about systems to the cluster of workstations used in the CosmoNet. On the software side we see the same variety ranging from interleaved audio-video codecs to separate applications for each of audio and video.

8.5.1. Hardware setup

The typical monolithic system, consisting of a single computer connected to the network by a single interface in one end and interfaces to presentation devices in the other, is the lowest degree of hardware distribution. In this configuration the data flows may have to share resources all the way from the capture to the presentation interfaces. To make this architecture able to handle a large number of remote sites we need an end-system dimensioned for the worst case scenario. This is the most common end-system architecture, implemented for both desktop and room-based videoconferencing.

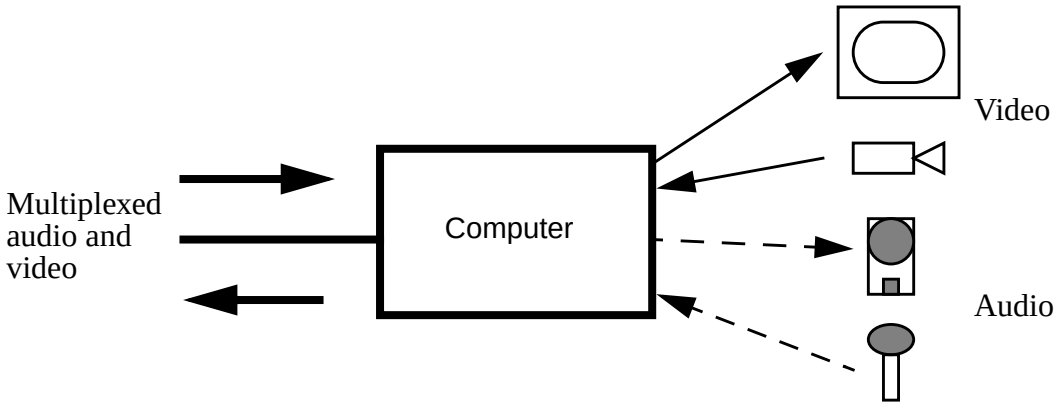

FIGURE 20. Monolithic system with separate audio-video I/O.

If we split the network connection into separate interfaces for incoming and outgoing data, only the computer need to be dimensioned to handle a worst case scenario. This architecture is useful if the local network is the bottleneck in the end-system. E.g. if only 10 Mbps Ethernets are available and the data streams needs 6 Mbps each, you have to use separate interfaces for incoming and outgoing data. It may also be necessary to use different network interfaces for audio and video, e.g. if using analog composite transmission over fiber.

With today's networks, lack of transmission bit rate is seldom the bottleneck. Instead, the bottleneck usually is located in the computers. By using separate computers for incoming and outgoing data, connected to the network by separate interfaces, you can relieve the computers from the burden of multiplexing incoming and outgoing flows. Also, by distributing the incoming data over a set of receiving computers, the videoconferencing system can scale to larger numbers of participating sites. As long as incoming and outgoing flows have no need for correlation, this architecture has no side effects. The network or audio-video equipment used may need a multiplexed flow, though, forcing multiplexing points in those parts of the end-system. Specialized equipment for this is commonly available, e.g. routers, switches and analog mixers.

If the computers are still too heavily loaded, there is one more dimension in which to split the data flows. By using specialized computers for running audio and video encoders and decoders we can reduce the complexity of the codecs and avoid the overhead of multiplexing/demultiplexing the audio and video data flows. In this case handling audio and video priorities must be taken care of by the network and/or audio-video equipment. Resynchronization of audio and video may become problematic, but with NTP it is possible to keep the computers in an end-system synchronized to a few milliseconds accuracy.

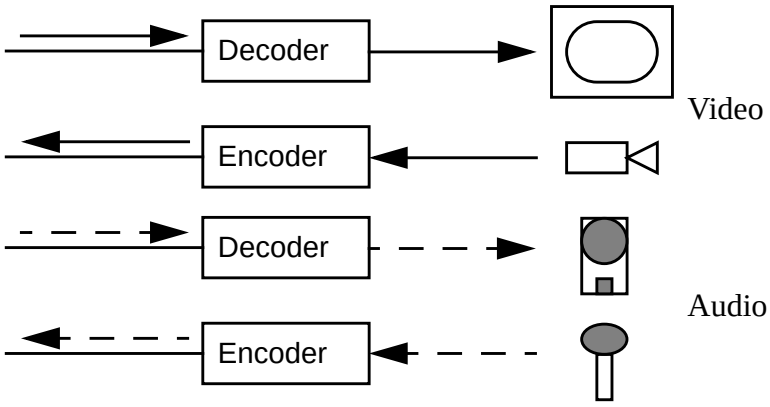

FIGURE 21. Data flows fully separated.

8.5.2. Audio-video software organization

Even though each of the computers, networks and audio-video equipment has it's own software architecture, the software handling the audio and video data mediation through the computers is the glue that make the system work as a unit. Typical functions include conversion between different protocols, compression/decompression, resynchronization and loss concealment.

The architectures for the audio-video software, in theory, can have the same grades of separation as the hardware - there are monolithic applications handling both incoming and outgoing interleaved audio-video data and there are specialized audio tools and

video tools - but there are a few combinations of hardware and software that are not feasible. E.g. a monolithic application transfering an interleaved audio-video format is not the best choice for a distributed hardware setup. Thus the audio-video software should be designed specifically for a certain hardware setup and vice versa.

Another constraint is a need for information sharing and filtering. The reason for not separating the sender and receiver part of an audio tool is that the decoder part needs to know what the encoder part has sent to be able to avoid local echoes when using IP multicast.

8.5.3. Summary

The preferable end-system design depends on many things; the capacity of the different components of the end-system and location of bottlenecks is one important factor, as is constraints given by the room design or budget. There is no single configuration that works best, instead the end-system design can be used to cancel bottlenecks in its components or it can be forced into a certain design in order to “glue” normally incompatible components together.

9. The prototype

The prototype still has no fansy name, which is mainly because it's development has been a series of additions to test different aspects discussed in chapter 8. The outline of this chapter will follow the steps in sections 8.2 to 8.5 discussing the design tradeoffs and choices made in the implementation. A RTP Payload Format for DV [62] is under development in the IETF and the prototype is one of three known implementations used in the development of this standard. See also [63] for more up-to-date information about the prototype development.

9.1. Choice of audio and video coding

I chose the HD Digital VCR Coference codec, commonly called DV, because it's SD-DVCR format provides sufficient resolution and because of the new DV camcorders that you can connect to a computer via IEEE 1394 becoming available on the consumer electronics market. Actually, this solution has been mainly targeted for *Non-Linear-Editing* (NLE) and not for real-time applications like videoconferencing, but the hardware codec and the isochronous transfer over IEEE 1394 allows real-time applications as well. The DV bit-stream is rate-limited to a constant bit-rate, which makes it easy to compile a flow specification for the traffic in case resource reservation is available, and if no resource reservation is available, the compressed format is addressable which enables error concealment algorithms operating in the compressed domain. The interleaved audio-video format of DV allows audio-video synchronization, but also means that you cannot prioritize the audio over the video, e.g. by sending the audio separately over a guaranteed QoS transport, without having to parse the DV stream to first filter out the audio at the sender and then put it back in the right place at the receiver. Another drawback is the high bit-rate, a SD-DVCR encoded bitstream carried over the IEEE 1394 bus by the *Digital InterFace* (DIF) takes about 28 Mbps.

9.2. Choice of hardware

DV, like most other modern video compression schemes, uses a *Discrete Cosine Transform* (DCT) step and thus needs hardware acceleration to achieve real-time performance. The easiest way to get your hands on a DV hardware codec is to buy a camcorder equipped with a IEEE 1394 socket. Then you equip your computer with a IEEE 1394 interface card if it's not already provided. All the computer system have to do then is to translate between IP and firewire, and handle resynchronization of the data at the receiver side.

There is also a DIF format for transferring MPEG over IEEE 1394 if the bit-rate of DV is considered too high. Other nice properties of IEEE 1394 is that it supports 100 to 400 Mbps isochronous transfer between up to 63 nodes attached in a non-cyclic tree topology by cables of up to 4.5 m in length. Specifications for much longer cables (30-50m) is also in development [64].

9.3. Software organization

The prototype is split into a sender part and a receiver part. The sender part takes input data from either a camera or from file, packetize it into RTP/UDP/IP and use either multicast or unicast to reach the receiver. On the receiver side, the data is processed to recreate the isochronous data stream that then can be output to a camera or VCR via IEEE 1394, or to a file, or to a local display. Splitting the prototype in this way makes it independent of the end-system architecture since it can be used in both end-systems and end-hosts. Local display of DV video on the computer screen demands hardware accelerated DV decompression in the host. At writing time, his part is still under development. See [65] for more information.

The RTP part is quite rudimentary, but slowly improving. It uses the standard BSD sockets API for the IPv4 and IGMP transmission and the source code is written in C. There are some UNIX-specific code since the receiver program is split into two parts, one that receives packets asynchronously from the IP network and one that handles resynchronization of data for isochronous transmission over the IEEE 1394.

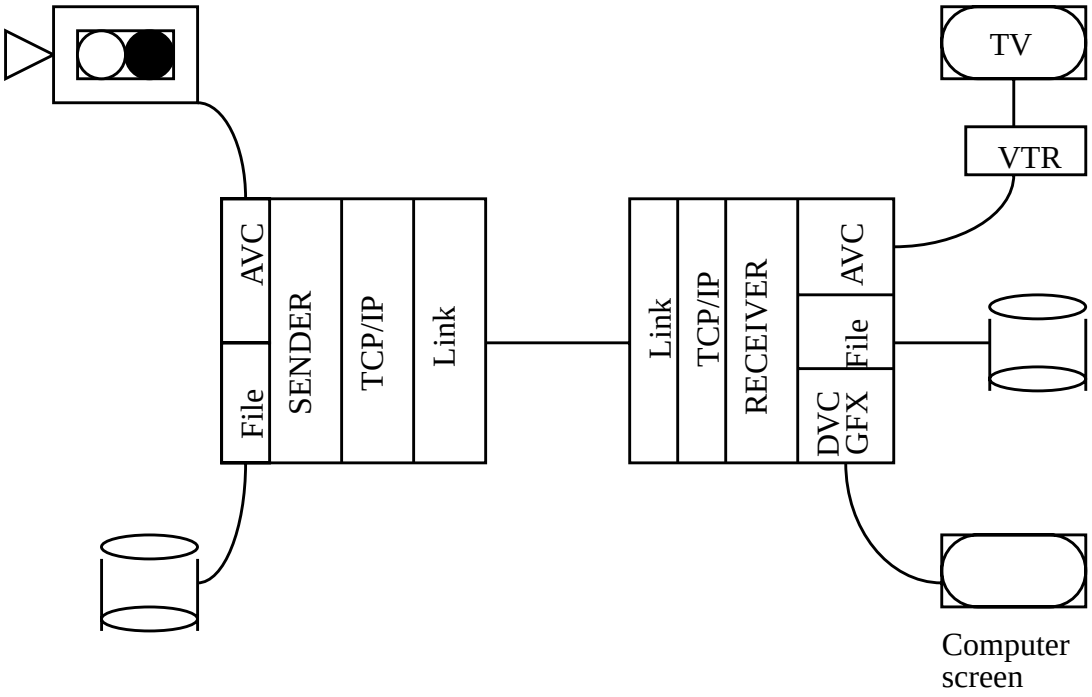

FIGURE 22. The prototype

9.4. A proposed end-system architecture

I think that a distributed end-system is a better choice than an end-host because it minimizes the multiprogramming overhead and the influence of multipoint conferencing on scalability. The high bit-rate of DV saturates most LAN technologies used today if the videoconference is multiparty. A distributed design allows for an arbitrary number of receiver machines to cope with scaling problems related to the number of additional end-systems sending traffic.

Since UNIX doesn’t have a RTE, I have to rely on overprovisioning in order to avoid data loss in the kernel and applications. I do this by using separate machines for incoming and outgoing traffic as well as by minimizing competitive load in the machines. Also, in many cases the LANs of the sender and the receiver don’t support guaranteed QoS. E.g. in the case of Ethernet the main delay contribution is due to collisions [66], therefore having only one sender and one receiver on each segment should minimize this delay contribution. However, if none of these above constraints apply, i.e. if the link technology and hosts involved all supports QoS guarantees and can handle the maximum amount of data, then any end-system architecture can be used.

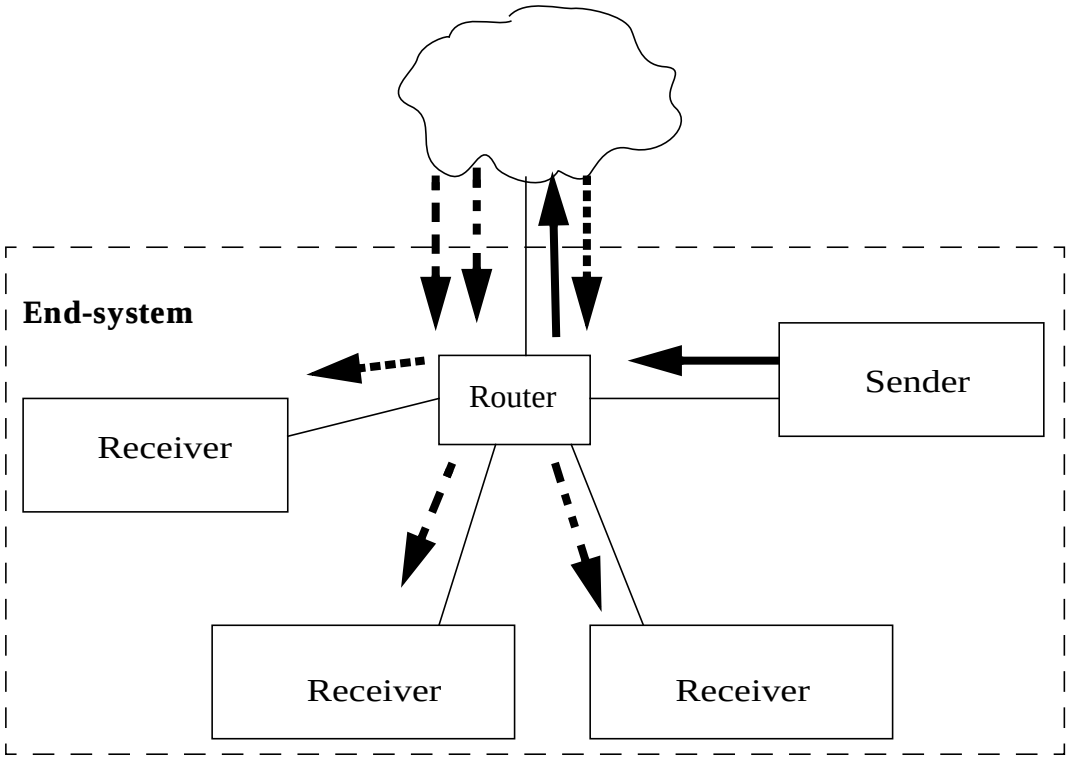

FIGURE 23. Proposed end-system design.

In a distributed end-system design the multiplexing bottleneck is located in the router, which in general is optimized for this task. Enabling technologies are:

- IP multicast allows sending to a multicast address without having to join the distribution tree. This minimizes the amount of traffic on the link between the sender machine and the router.
- IGMP v.3 allows for joining a subset of the sources in a multicast group. This allows for load balancing between the receiver machines and their links to the router.

9.5. Packetization

The RTP payload format for DV is quite straightforward and doesn't require any changes to the RTP header other than those defined in the A/V profile. One important note though is that packet boundaries must reside on DIF block boundaries, which means that the packet payload will always be a multiple of 80 bytes long.

The RTP ADU size was chosen so that RTP/UDP packets wouldn't get fragmented over an Ethernet since over 60% of all LANs in the world consists of Ethernet. However, if some other link technology is used the ADU size is easily configurable at compile

time. It would also be easy to extend the application to make use of a MTU discovery service to dynamically change the ADU size, if such a service becomes available in the future. As we saw in section 8.4.2, the RTP, UDP and IP headers together occupy 40 bytes, which leaves 1460 bytes payload. The DV payload format doesn't specify any header extensions, but since 1460 is not a multiple of 80 bytes, we can use at most 1440 bytes payload in each packet.

The protocol overhead for DV transmission intuitively should be very small due to the high bit-rate of the DV data. A PAL DV frame is 144 kbytes large, thus requiring 100 packets per DV frame. With a frame rate of 25 DV frames per second times 100 packet per DV frame times 40 bytes ov headers in each packets gives a protocol overhead of 800 kbits/s, or 2.7 %. The NTSC DV frame is a little bit smaller, only 120 kbytes, requiring 83 full packets plus one leftover packet with 480 bytes payload. A frame rate of 29.97 DV frames per second times 84 packets per DV frame times 40 bytes of headers in each packet gives a protocol overhead of 805 kbits/s, which also is about 2.7%.

10. Evaluation of the prototype

As we saw in the preceding chapter, the DV format was chosen because it fulfills the requirements on audio and video resolutions in section 8.1. In this section I will present some tests performed to see if the prototype also complies with the rest of the parameters from section 8.1.

10.1. General description of the evaluation model

By assuming that we have a distributed end-system, such as the one proposed in section 9.4, a simple model of the data path can be used since each media flows independent of the other. The model I will use is common in textbooks and papers on real-time multimedia systems and consists of more or less independent blocks, all contributing to the end-to-end behavior. At the sending site, the signal is sampled and digitized, then the digital signal is compressed, and finally the compressed signal is packetized in a suitable way for transmission over the network to the receiving site. At the receiving site, techniques for delay equalization and loss recovery are performed to reconstruct the compressed digital signal. The compressed digital signal is then decompressed and the *intramedia synchronization* is reconstructed before the digital signal is converted to an analog format suitable for the presentation device.

In the test setup, the sampling and signal reconstruction as well as compression and decompression, is performed in the camcorder, while the prototype handles the packetization and depacketization of data, delay equalization and loss recovery. Thus, the contribution to end-to-end delay, delay variation and distortion of the boxes located in the camcorder and the DV-to-analog converter box could not be measured separately.

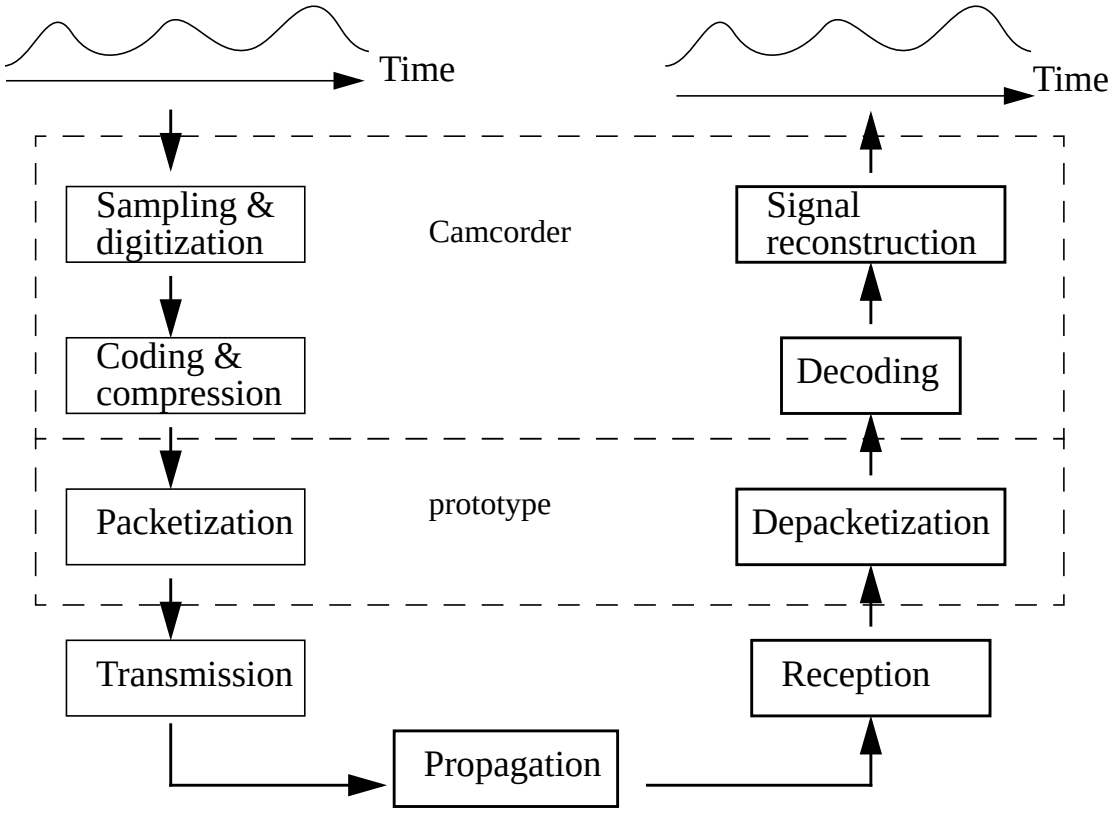

FIGURE 24. Signal path through the videoconferencing system.

10.1.1. Limitations of the model

We cannot use this model to predict the effects of differences in auxiliary factors such as setup-, appearance- and quality of analog equipment and the influence of non-audio, non-video functionality. These factors are important to the perceived end-to-end quality as well, but are not treated in this report.

The model would also perform poorly if used, as is, to predict the effects of some multipoint communication issues. For example, to model multipoint communication without any transcoding we would need multiple independent copies of the stacks and the propagation boxes. However, this problem can be turned into a pure networking issue. The base for this reasoning is that, if I cancel out transportation factors, a signal from one sender is conceptually independent from signals from other senders. In section 10.5 below I do this by using a single Ethernet between the machines and doing measurements to detemine the contribution to end-to-end delay, delay variation and distortion by this minimal network.

10.2. Tools

The measurement tools that I used in the experiments were all software programs running on the test machines. Except for DCTune, that had to run in a Sun Solaris environment.

10.2.1. MGEN

The *Multi-Generator* (MGEN) toolset from the U.S. *Naval Research Laboratory* (NRL) [67] consists of a traffic generator tool, *mgen*, a recording tool, *drec*, and a tool for analyzing the recorded data, *mcalc*. The *mgen* tool can send one or more traffic flows over IP unicast or multicast. The generated packets contains a 20 bytes long header followed by padding bytes to give the requested packet size. Apart from the packet size, one can also specify the packet rate, ttl, start-time, duration, interface, the receiving port number, the base address and number of sequential multicast groups to send to, and the packet generation distribution. The version of *mgen* that I used, version 3.1a2, only supported periodic or poisson distributed packet generation and couldn't send multiple flows onto the same multicast address. At exit time, *mgen* reports number of packets sent, transmission time and average packet rate as well as link layer statistics.

The *drec* tool records the incoming packets to a logfile that later can be analyzed by hand or a summary can be computed with *mcalc*. The summary contains the number of packets received, packet rate, bit-rate, packet loss and delay related parameters such as max, min and average delay and delay variation. To measure delay-related parameters, synchronized machines is needed. *mgen* and *drec* can be run in a graphical mode or in command line mode.

10.2.2. C-libraries for time measurements

For some timing measurements I inserted timing code into the source code of the prototype. The function used for timing was `gettimeofday()` defined in `sys/time.h`.

The timing code overhead was measured to be around 30 microseconds.

TABLE 9. Timing code overhead on an SGI O2

Max	0.000043 seconds
Min	0.000019 seconds
Average	0.00002981 seconds
Standard deviation	0.000008334 seconds

10.2.3. DCTune

DCTune 2.0 was used to measure perceptual distortion introduced by the DCT-part of a video compression scheme. The metric used in this tool, called the *Digital Video Quality* (DVQ) metric, is presented in [44]. The DVQ metric computes the visibility of artifacts expressed in the DCT domain using a model of human spatial, temporal and chromatic contrast sensitivity, light adaption and contrast masking. It is described by its inventor as being “reasonably accurate, but computationally efficient”. It is still far from real-time and the input formats are limited to the *Portable PixMap* (PPM), *Portable BitMap* (PBM) and the *Portable GrayMap* (PGM) formats. PPM/PBM/PGM is never used for motion video so each frame in both the original and the tested video clips has to be converted from the normal input/output format of the video compression scheme to PPM/PBM/PGM. The result of DCTune includes *just-noticeable differences* (jnds) and perceptual errors.

10.2.4. top

top displays the processes on the system, ranked by percentage of raw CPU usage. The actual display varies depending on the specific variant of UNIX that the machine is running, but usually includes among other things; total size of the process, resident process memory, and the raw CPU percentage. *top* also includes a summary of system load containing among other things; the number of existing processes in the system and a summary of the overall CPU, physical and virtual memory utilization.

10.2.5. xntp

The *Network Time Protocol* (NTP) is used by Internet time servers and their peers to synchronize clocks, as well as automatically organize and maintain the time synchronization overlay network itself. It is specifically designed for high accuracy, stability and reliability, even when used over typical Internet paths involving multiple gateways and unreliable networks [68]. In the experiments below I used xntp3-5.39e running NTP version 3 [69] to keep the machines synchronized. In 1991 University of Delaware measured less than 50 ms discrepancy between all time servers, all but 1% were below 30 ms difference and some servers had a 1-millisecond accuracy.

Timing accuracies to a few milliseconds and frequency stabilities to a few milliseconds per day were regularly achieved [70]. With the short and fairly constant RTT (2 ms) to the time server shared by the two test machines, I judge the synchronization between the test machines to be within a few milliseconds accuracy.

10.3. The testbed

The system I used for these tests consisted of two SGI O2s connected by a single 100BaseT Ethernet and a Sony TRV900 camcorder attached to one of the computers via IEEE 1394. For the end-to-end delay measurements I used a Sony DVMC-DA1 DV-to-analog converter. I have also tried using a Panasonic AG-DV2700 VTR attached to the receiving computer although it had some problems initializing the reception over the IEEE 1394 interface from the computer. The VCR-part of the camcorder could handle both NTSC and PAL formats so both formats could be used in the tests involving the camcorder. The AG-DV2700 could only handle PAL feed, and the DVMC-DA1 could only handle NTSC feed.

I chose the SGI O2 since it is built to handle video and hopefully wouldn't introduce any hardware bottlenecks. Also, the operating system has a programming environment for digital media that offer a limited real-time support. However, the prototype development has partly taken place in Sun Solaris environment as well, so none of the SGI IRIX-specific features has been used so far. The two SGI O2s that I used were equipped with 180 MHz MIPS R5000 processors, 384 and 128 Mbytes of main memory, 100 Base Tx network interface cards, Iris Audio Processors, and MVP video. The operating system was IRIX version 6.5.4. For interfacing to the IEEE 1394 I used SGI DVLink 1.1 that is based on a Texas Instruments TSB12LV22, which is a later version of the chip used in the DVTS in section 6.5. The DVLink package also includes a library, called AVC, for controlling the card and for sending data over the IEEE 1394.

10.4. Platform performance measurements

I did a few experiments to test the platform performance characteristics and their influence on the prototype. First I measured the competitive load, i.e. the number of processes and the CPU time and memory usage of those processes. Using the UNIX command *ps* I counted up to 42 processes out of which 24 could be removed. Using *top* I saw that the remaining 18 processes used up to 2% CPU time and 22 to 83 Mbytes of memory.

Of the competitive load processes, the X server was the most computationally expensive. When there is a lot of activity in the window manager, the X server consumes most of the CPU time, but when no user interaction takes place, the X server consumes very little CPU time.

During the experiments I could not see any significant difference in performance due to the choice of sending and receiving machine. Since the only difference between the two machines was the amount of memory, this was no surprise.

10.4.1. CPU and memory usage of the prototype

The load generated by the sender when taking input from the camcorder via AVC was 32% CPU time, 3000 Kbytes total process memory and 732 Kbytes resident memory. When taking input from file, the CPU time was 88-90% because the prototype use a busy loop for timing the sending of frames. The memory used was 2976 Kbytes total and 744 Kbytes resident.

On the receiver side, the application use two processes when output to AVC. Together, the two processes used around 35-40% CPU time, 7780 Kbytes total and 2376 Kbytes resident memory. When output to file it used the same amount of CPU time, but only half as much memory.

10.5. Network performance measurements

The minimal network connecting the two test machines should not influence the tests too much, but to be on the safe side I decided to measure delay, delay variation and packet loss using the MGEN toolset simulating the data stream as generated by the prototype. I simulated PAL by sending 1440 bytes payload packets at 2500 pkts/sec and NTSC by sending 2487.51 full 1440 bytes payload packets per second on one multicast address and 9.99 leftover packets with 480 bytes payload on a second multicast address. The two multicasted NTSC flows' start times were separated by 15 ms to place the simulated leftover packet at the end of the simulated NTSC DV frame. The total bit-rate for PAL was 28.8 Mbps and for NTSC it was 28.7 Mbps.

I also tested using both of the two packet generation distributions available in *mgen*. Five test runs of 60 seconds for each of the four permutations gave the data shown in table 10 and 11. For these runs, a packet loss between 0 and 0.3 % (0.05 % on average) was measured for PAL and 0 to 0.02% (0.003 % average) for NTSC.

TABLE 10. MGEN end-to-end delay per run

Flows	Max	Min	Average
PAL periodic	0.012-0.172 seconds	0.002-0.004 seconds	0.003-0.005 seconds
PAL poisson	0.013-0.094 seconds	0.002 seconds for all	0.003 seconds for all
NTSC periodic	0.013-0.021 seconds	0.007-0.008 seconds	0.008 seconds for all
NTSC poisson	0.012-0.060 seconds	0.003-0.008 seconds	0.003-0.008 seconds

As you can see, the max delay can become more than 4 times the IDT. However, the very low average delay, combined with the high max values indicates that spikes of delay occurs. PAL poisson showed fewer delay spikes. Intuitively it should have the opposite behavior, but I would need more tests to draw any serious conclusions about the impact of the choice of packet generation distribution on delay and loss.

TABLE 11. MGEN end-to-end delay variation per run

Flows	Delay variation
PAL periodic	0.009-0.169 seconds
PAL poisson	0.011-0.092 seconds
NTSC periodic	0.014 seconds
NTSC poisson	0.008-0.057 seconds

10.5.1. Transmision and propagation

The delay and delay variation for transmission and propagation should be insignificant in this minimal network. However, send collisions on an Ethernet have been showed to cause a delay of about 10 ms at around 40-50% throughput [66]. This is a far way from 170 ms (the maximum delay for PAL periodic) but the study was for audio which generated much smaller packets. Therefore I checked the collisions reported by *mgen* but saw no correlation between send collisions on the Ethernet and the delay spikes, so collisions on the Ethernet is not the single cause even if it contributes to the delay.

10.5.2. Packetization and depacketization

Packetization and depacketization delay, on the other hand, depends on CPU load and the amount of context switching. Measurements with *top* during the PAL periodic transmission showed that *mgen* used all available CPU resources, around 88.5 %, and that the size of the process was 5260 Kbytes of which 932 Kbytes were resident in memory. *drec* didn't use up all of the CPU resources, only around 52 % but the competitive load left a tiny 2 to 7% CPU idle time.

The size of the *drec* process was 5172 Kbytes of which 936 Kbytes were resident in memory. Considering the minimal marginals in CPU time in the receiver machine, a probable cause for the delay spikes is that competitive load blocks the CPU and prevents *drec* from retrieving data from the network. Another observation that supports this theory is that the delay peaks coincided with a higher packet loss, which in combination with the high data rate leads me to suspect that it was some buffer in the TCP/IP stack that overflows causing the corresponding loss.

10.5.3. Delay equalization buffer size of the prototype

From the delay variation measurements we can compute the size of delay equalization buffer needed in the prototype on this end-system configuration. The IDT for PAL is 0.040 seconds and for NTSC the IDT is 0.0333 seconds so we need a buffer capable of holding at least 5 PAL frames or 2 NTSC frames to eliminate the packet transmission delay variation of the minimal system used in these experiments.

In the prototype measurements below I used a six frames long delay equalization buffer with a 66 ms preloading time, and no buffer underrun conditions was encountered. Some tests with smaller buffer sizes resulted in buffer underrun, though.

10.6. Delay measurements

There are three different types of delay that one have to consider in a videoconferencing system; the initialization delay and response time of the system, and the end-to-end audio-video transmission delay. There currently is no interaction with the system after initialization, so response time is not a valid factor. The end-to-end audio-video transmission delay can in turn be divided into delay contributed by different boxes in the evaluation model. I did some measurements of the time needed to read a DV frame from file in the sender and the time needed to write a DV frame to file in the receiver.

10.6.1. Initialization delay

As mentioned in section 3.3, the general rules of user interface design applies also for videoconferencing. I have tested the initialization delay of the camcorder, DV-to-analog converter and the sender and receiver applications through subjective observations and timing with my wrist watch. This time resolution should be enough given that the initialization delay can be allowed up to seconds. Using this primitive method I found that the camcorder, DV-to-analog converter and the applications each needs less than one second to start up.

10.6.2. Read from file and write to file

The sender part of the prototype can get input data both from AVC and from a file. Likewise, the receiver part can output data to AVC or writing it to a file. At the sender, DV frames are read from file at the same rate as if captured from AVC and the frame is not sent until completely read from file. The machines were equipped with video-certified hard disks with an average seek time of 7.8 ms, but I still wanted to see how this part affected the overall delay budget. For the measurements I used timing code inserted into the source code of the prototype. The overhead of this timing code was about 30 ns (section 10.2.2). 18 reads and 43 writes were timed for NTSC, while for PAL 15 reads and 46 writes. The measured maximum, minimum and average delays are shown in table 12. Although the graphs show both NTSC and PAL, the measurements were not simultaneous.

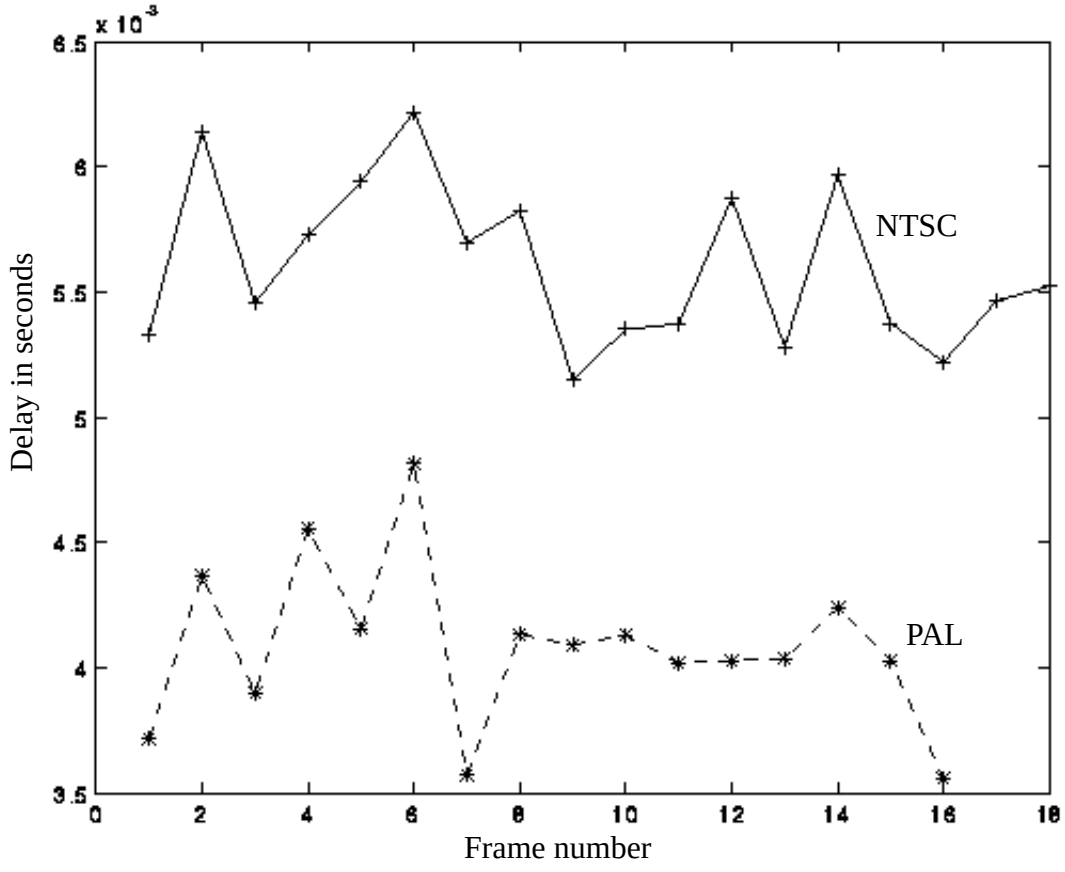

FIGURE 25. Read DV frame from file delay.

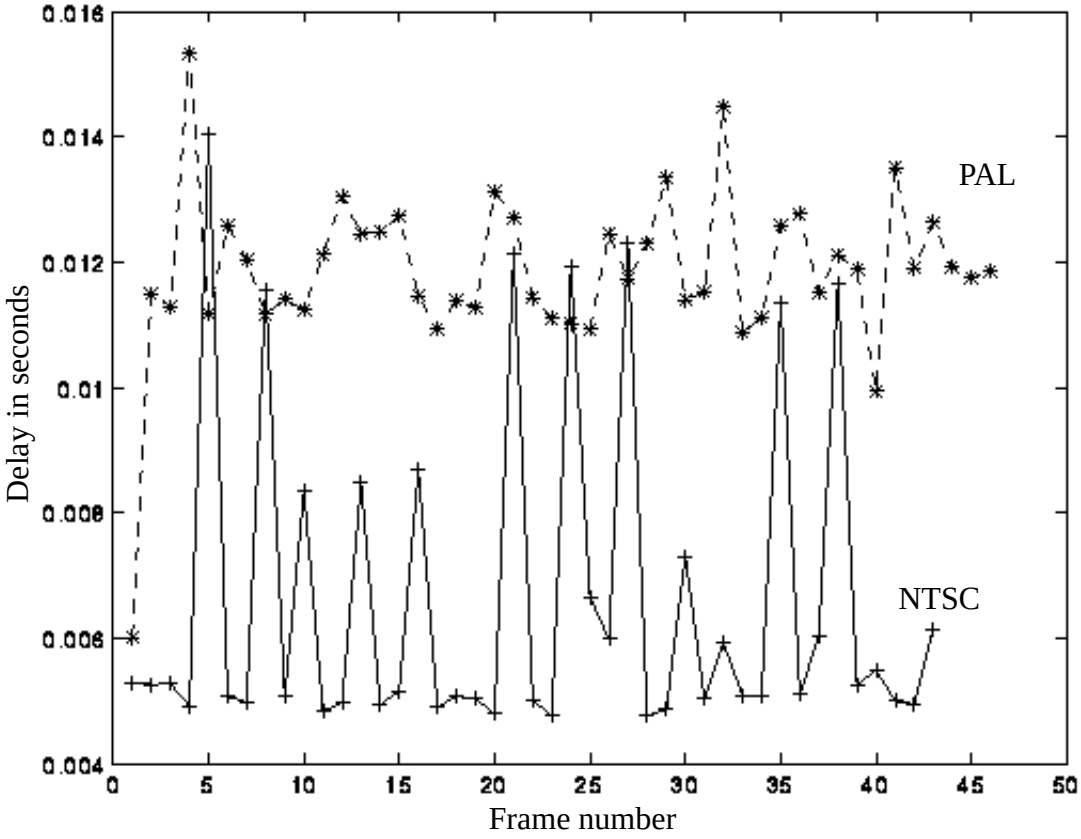

FIGURE 26. Write DV frame to file delay.

TABLE 12. Read and write delays

File operation	Max	Min	Mean
Read PAL	0.0048 seconds	0.0036 seconds	0.0041 seconds
Read NTSC	0.0062 seconds	0.0052 seconds	0.0056 seconds
Write PAL	0.0153 seconds	0.0060 seconds	0.0119 seconds
Write NTSC	0.0141 seconds	0.0048 seconds	0.0066 seconds

The difference between maximum and minimum delay for reads were very low, while for writes the gap is much larger. The standard deviation for PAL writes was 1.3 ms and for NTSC writes it was 2.6 ms. While most of the data are localized around the mean, especially the NTSC writes suffered from frequent delay spikes. The maximum write delays of close to 1/2 IDT for NTSC could pose a problem since the application cannot receive any packets while writing to disk.

10.6.3. Packetization and depacketization

The implementation on which these measurements were conducted generates bursts of packets for each frame. This behavior was not possible to simulate using MGEN since it only supported periodic or poisson distributed packet generation. Therefore I did complementary tests using timing code inserted into the prototype. Only the File-to-File scenario was tested. No tests of the influence of File v.s. AVC input and output were done here. I did 21 test runs of sending NTSC, 15 when sending PAL. On the receiving side I did two tests, one for NTSC(257 frames) and one for PAL(387 frames) of which I only show the timings of the first 100 frames in the graphs below.

In the sender I measured the time needed to packetize and send a DV frame. This includes building RTP headers, filling the packets with DIF blocks and writing the packets to a BSD socket. In the receiver I measured the time between RTP timestamp incrementation and the reception of a packet with the RTP marker bit set to find out how long time it takes to read the packets from a BSD socket, do some minimal parsing of the RTP headers, detecting lost and reordered packets, and inserting the packets in the right place in the right frame in the playout buffer. Some of the measurements also include write to file of DV frames causing a number of spikes.

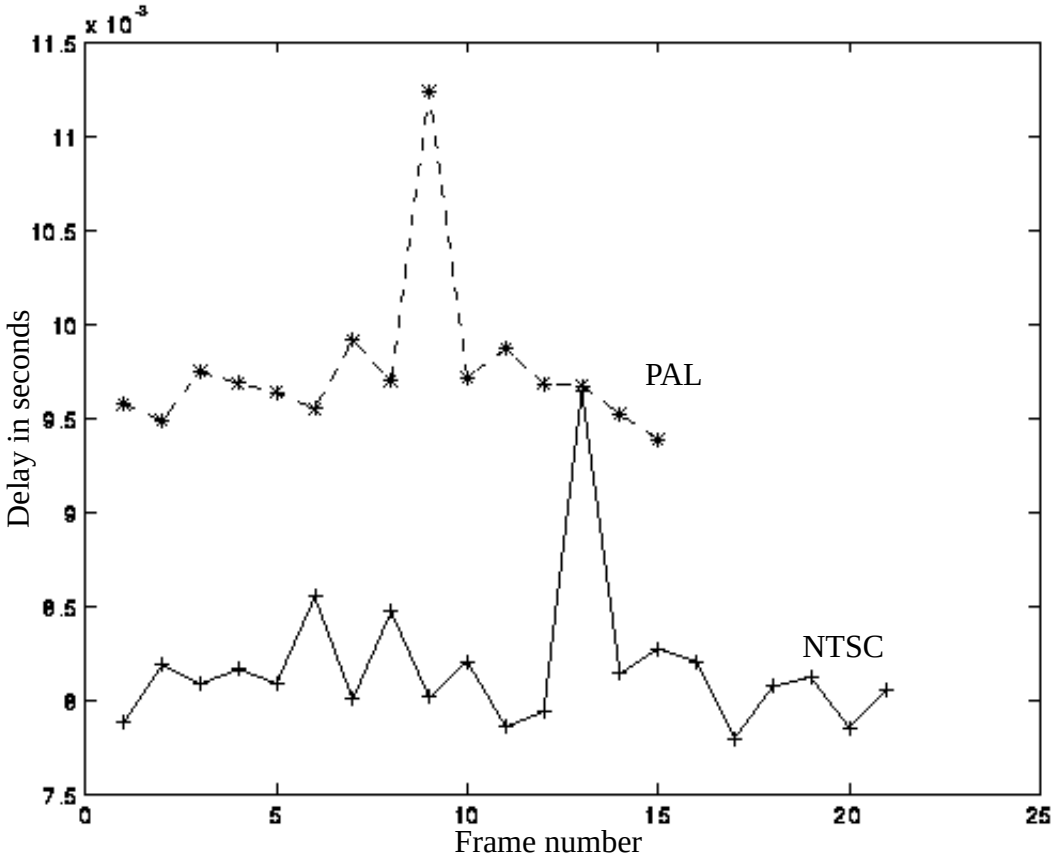

FIGURE 27. Time to send all fragments of a DV frame.

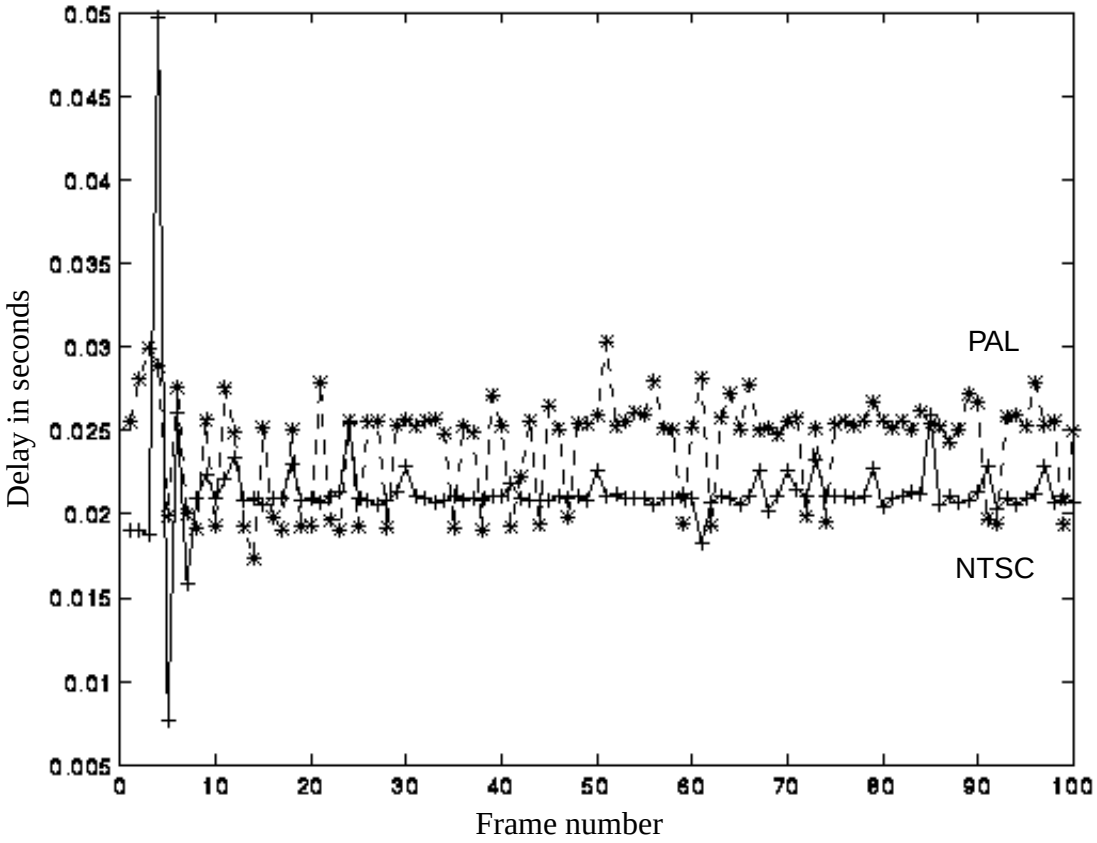

FIGURE 28. DV frame reception time.

TABLE 13. Packetization and depacketization delay.

Task	Max	Min	Mean	Standard deviation
Send PAL	0.0112 seconds	0.0094 seconds	0.0098 seconds	0.0003 seconds
Send NTSC	0.0096 seconds	0.0078 seconds	0.0082 seconds	0.0007 seconds
Receive PAL	0.0332 seconds	0.0143 seconds	0.0255 seconds	0.0036 seconds
Receive NTSC	0.0497 seconds	0.0077 seconds	0.0214 seconds	0.0023 seconds

Packetizing a NTSC DV frame using the prototype took around 8 ms, while for a PAL DV frame around 10 ms was needed. Receiving a NTSC DV frame from a BSD socket took around 21 ms, while receiving a PAL DV frame took around 25 ms.

Most measurements are located around the mean values while the maximum and minimum values consists of occasional outliers. This is to be expected since the measurements only cover frames where the packet containing the RTP marker bit was not lost. Thus, the frame timings long enough to trigger a buffer overflow will also experience packet loss at the end of the frame.

10.6.4. End-to-end

From the measurements presented in the previous sections, we can do a fair guess about the amount of end-to-end delay and delay variation to expect from the testbed. Here I sum up the maximum, minimum and average values of the read and write to file measurements in section 10.6.2 and packetization and depacketization measurements in section 10.6.3.

TABLE 14. Delay in the prototype applications.

Prototype part	Max	Min	Average
Sender PAL	0.0160 seconds	0.0130 seconds	0.0139 seconds
Sender NTSC	0.0159 seconds	0.0129 seconds	0.0138 seconds
Receiver PAL	0.0389 seconds	0.0232 seconds	0.0321 seconds
Receiver NTSC	0.0578 seconds	0.0112 seconds	0.0284 seconds

As we saw from the graphs above, the maximum reception delay results from spikes at the beginning of a transmission and includes a write-to-file delay. Thus the maximum delay values includes a write-to-file delay too much. Since the average delay variation was quite small I think the computed average delay is quite trustworthy.

To these delays we should add the delay introduced in the network and delay equalization buffers. The MGEN measurements in section 10.5 also includes packetization and depacketization delays, but may give a hint about the expected network delay in the minimal testbed.

TABLE 15. End-to-end delay in the file-to-file case.

Feed	Max	Min	Average
PAL	0.2365 seconds	0.0353 seconds	0.0543 seconds
NTSC	0.1396 seconds	0.0285 seconds	0.0498 seconds

The AVC library only delivers a DV frame when it has received the whole frame over the IEEE 1394. This forces a constant audio and video buffering delay of one IDT (33, or 40 ms) at the sender when using AVC instead of reading from file. The prototype used for these measurements also had a six frames long delay equalization buffer with a 66 ms preloading time. Due to lack of measurements of the fill rate of the delay equalization buffer during the above measurements, only a rough estimation of the extreme values is possible.

TABLE 16. End-to-end delay in the file-to-file case including delay equalization.

Feed	Max	Min
PAL	0.475 seconds	0.075 seconds
NTSC	0.340 seconds	0.060 seconds

Note that the above computations are for file-to-file only. I also measured the end-to-end delay and delay variation by recording both a rapid change and the display of the receiving end on tape in the camcorder. This would give a maximum resolution of 33 ms, which is one fifth of the delay budget and one third of the delay variation budget in section 8.1. It might be too coarse to be useful but gives something to compare the calculations with.

Taking the video directly from the S-Video port of the camcorder to display on a monitor via an analog NTSC/PAL converter requires 4 frames delay, i.e. 133.5 ms delay is introduced in the camera. Adding the DVMC-DA1 DV-to-analog converter to the loop increased the delay to 7 frames, or 233.6 ms. Thus the DV equipment alone by far used up the end-to-end delay budget. Adding the prototype in the loop introduces another 6 to 8 frames worth of delay, due to the delay equalization buffering and the one frame delay in the sender application. Thus the end-to-end delay when using the prototype system, a Sony TRV900 camcorder and a Sony DVMC-DA1 DV-to-analog converter over the minimal testbed is between 434 to 501 ms. Around three times the delay budget of section 8.1.

Summarizing, the prototype system could not comply to the end-to-end delay budget from section 8.1. Although the fixed delay introduced in the camcorder and DV-to-analog converter alone is higher than the delay budget, most of the delay is introduced in the prototype. This motivates further work on how to minimize the length on the delay equalization buffer and overall optimization of the data paths in the prototype.

10.7. Distortion measurements

In the version of the prototype on which these measurements were conducted, packet loss results in loss of DV frames. The DV frame loss, in turn, may cause a visible “freeze” or “jump” in the video image that is distracting for the user and it will certainly cause an audible “rapping” in the audio. Thus the current loss handling in the prototype is far from optimal. An evaluation of different loss recovery functions is suggested as a possible future work item.

Another form of distortion is introduced by the DV compressor in the form of quantization noise. There have been a few evaluations of the video quality of DV v.s. other formats [71, Annex C of 72] and they are thorough enough, but I have also done some measurements of perceptual errors using DCTune on a typical videoconference scene.

Since DCTune needs both a reference frame and the frame to test, I did not use it to measure the distortion caused by lost frames due to packet losses. However, it might be suitable for measuring the relative distortion of different loss repair schemes in the future.

10.7.1. Frame loss measurements

I did some measurements of DV frame loss for different choices of input and output media to see what amount of distortion to expect being caused by data losses in the applications over the minimal network. To have something to compare with I computed the expected average DV frame loss for the MGEN measurements, which should be $1 - (1 - P_{packet})^{n}$ where *n* stands for number of fragments used to transport a DV frame and for P_{packet} we choose the average packet loss from section 10.5. For NTSC we get an expected DV frame loss of 0.24 %, and for PAL we expect to lose 4.6 % of the DV frames.

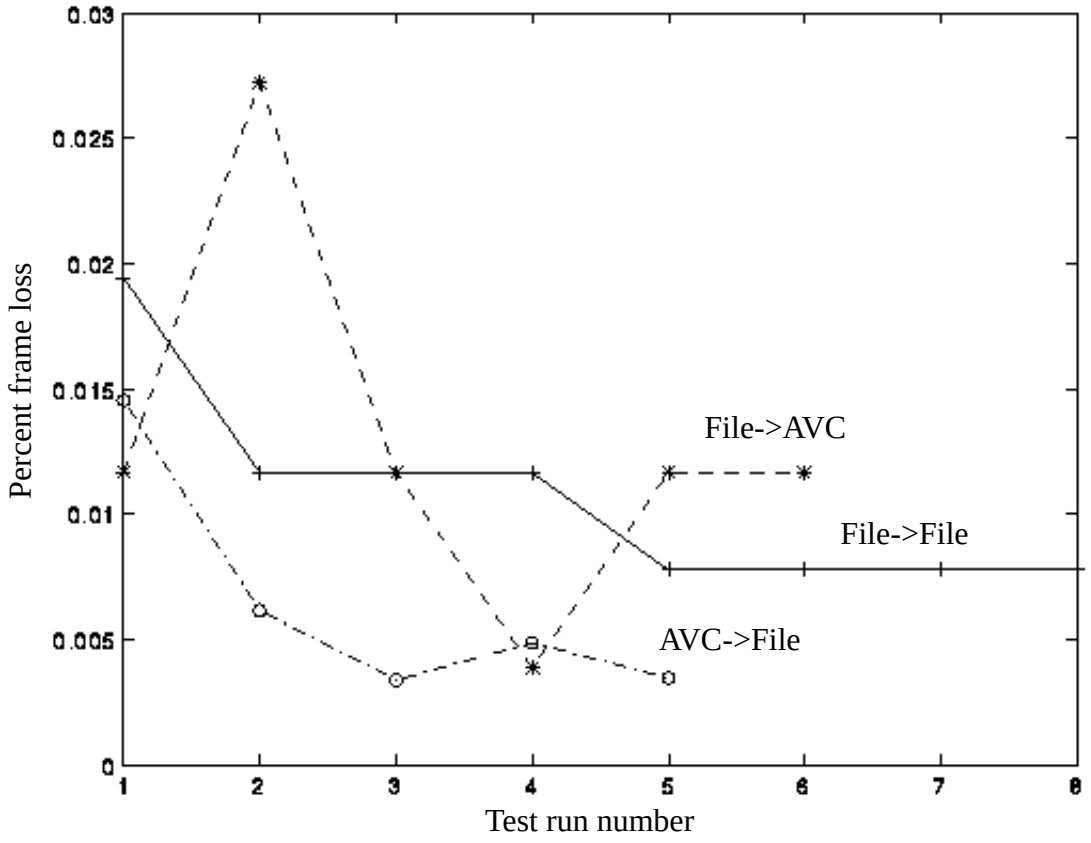

FIGURE 29. Frame loss of NTSC for different I/O combinations.

TABLE 17. Frame loss of NTSC for different I/O combinations.

NTSC	File output	AVC output
File input	0.78-1.95%, 1.07 % avg, 8 runs	0.4-2.7%, 1.3 % avg, 6 runs
AVC input	0.34-1.46%, 0.65 % avg, 5 runs	NA

The average DV frame loss of NTSC with the prototype is somewhat higher than the expected loss for MGEN. A part of this loss increase might be caused by extra context switching between the two processes in the AVC case.

TABLE 18. Frame loss of PAL for different I/O combinations.

PAL	File output	AVC output
File input	6.7-53.5%, 46 % avg, 22 runs	58-99%, 83 % avg, 24 runs
AVC input	6-35%, 16 % avg, 5 runs	NA

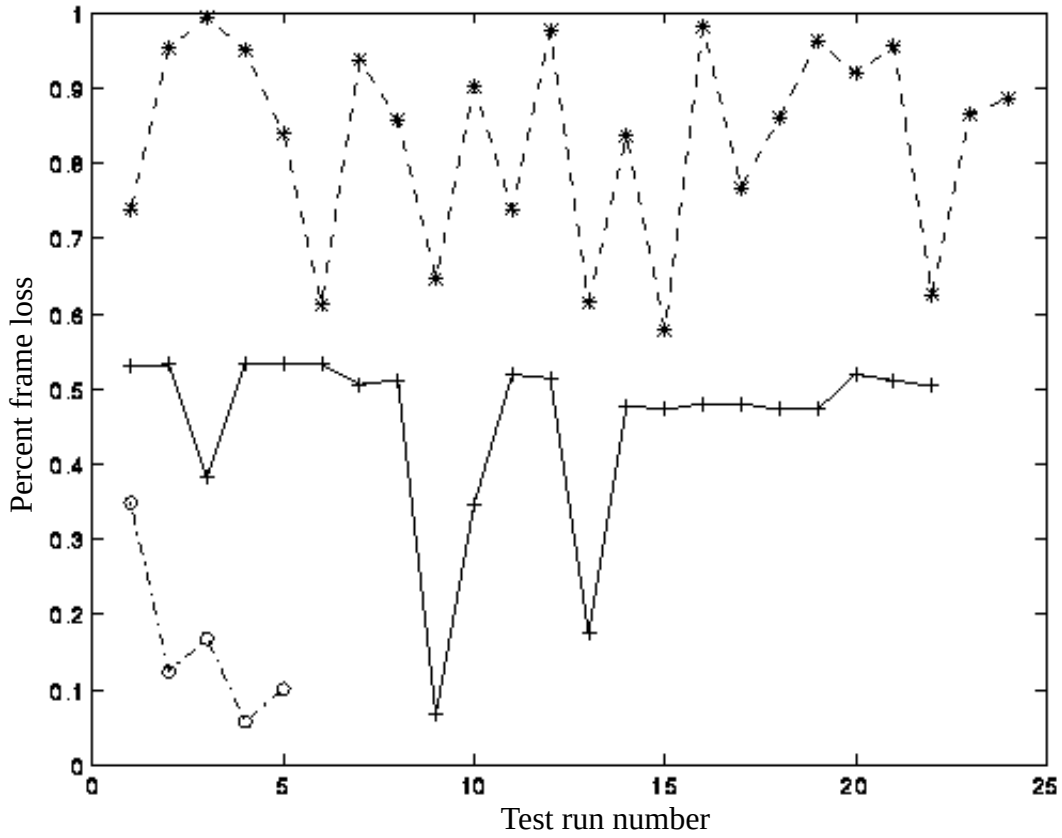

FIGURE 30. Frame loss of PAL for different I/O combinations.

For PAL, the average DV frame loss measured was far higher than the expected 4.6 % from the MGEN measurements. Reaching a staggering 83.4 % for the file-to-AVC case, the difference was so significant that it called for further investigation as presented in the next section.

10.7.2. Measuring packet loss in prototype

In the prototype, packet losses are detected using RTP sequence numbers. Therefore it was quite straightforward to measure the number of lost packets. The measurements presented here are for the file-to-file case, using the same video sequences as in the frame loss measurements above. During six test runs I got a quite constant 5% to 6% (5.7% average) packet loss for a sequence of 387 PAL DV frames. For an equal number of test runs on a sequence of 257 NTSC DV frames I measured 0.07 to 0.6% (0.3 % average) packet loss.

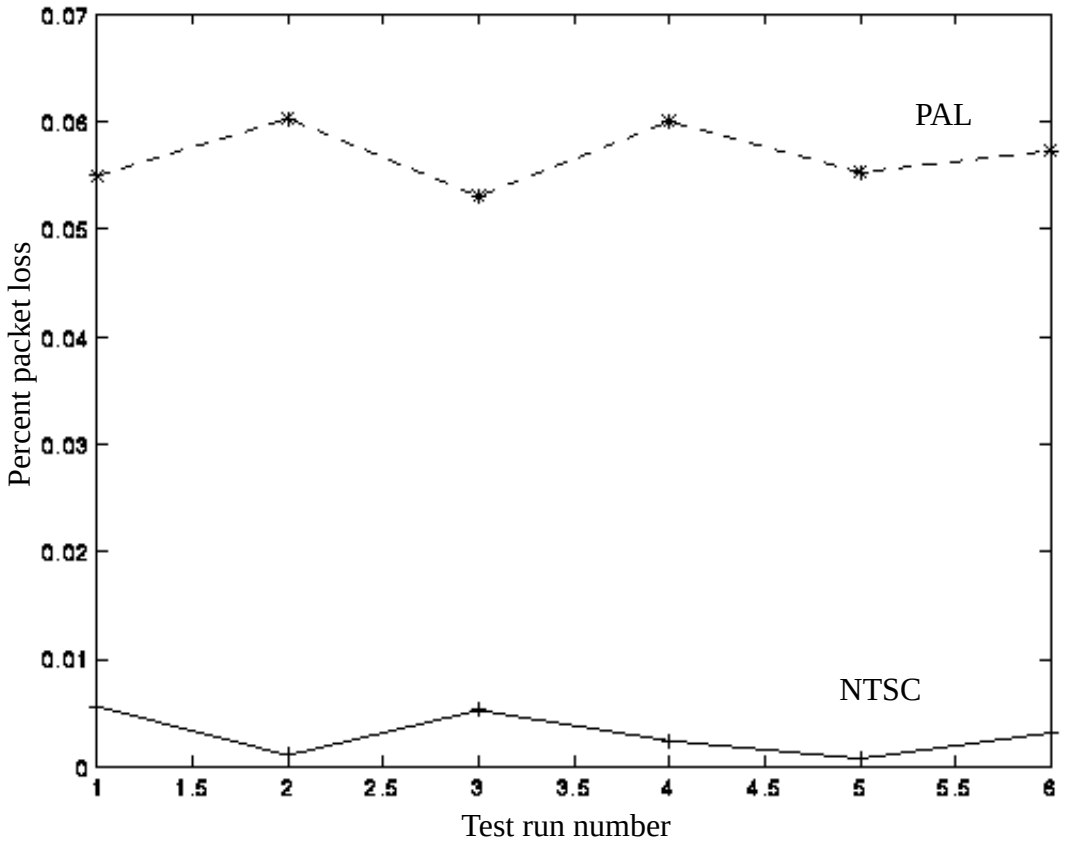

FIGURE 31. Packet loss in the File-to-File case.

Thus the packet loss for NTSC was a little higher than for the MGEN tests, while the losses for PAL was around 120 times higher than those measured in section 10.5. This, explains the much higher DV frame loss reported above. Now, what causes this huge increase in packet loss?

The applications in the prototype use less CPU time than the MGEN tools, so this cannot be the cause of the packet loss increase. I also checked if the difference in clip length was causing the extra loss by sending fewer than 257 PAL DV frames, but still got about 5% loss.

Left is the packet distribution in combination with the difference in frame size between PAL and NTSC. In the prototype, DV frame fragments are sent in bursts as fast as possible, and we saw in section 10.6.3 that it is twice as fast to packetize and send a DV frame than to receive and depacketize one. I.e. at least half of the DV frame data has to be buffered somewhere in the receiver machine. Also, in the implementation of the prototype that was used in these measurements, no packets can be fetched by the application as long as it is writing a DV frame to file, and as we could see in section 10.6.4, the depacketization time plus writing time is close to a full IDT.

Network adapters typically have 128 - 256 buffers [73] so the marginals aren’t that great considering that one PAL DV frame is fragmented in 100 packets. The NTSC DV frame, on the other hand, needs only 84 packets. My guess is that the network adapter runs out of buffers while the application is busy writing to a file or to AVC, or while the CPU is blocked by some other process. With even faster processors this problem may go away, but on the other hand we know that link speed increase proportionally faster than processing capacity. The best solution seems to be to smooth out the bursts by sending each frame over a longer interval than the one supported by the link technology. A proposed remedy would be to split the PAL DV frames in two or more parts and send the parts separated by a suitable interval. I left this optimization for Future Work.

10.7.3. Compression-induced distortion

A typical videoconference scene (VC-room, 383 frames long) was compressed and decompressed using the software DV codec included in SGI DVLink 1.1. The original uncompressed sequence and the decompressed sequence was compared by subjective tests on myself as well a measuring perceptual error using DCTune 2.0. I used the *Portable aNyMap* (PNM) tools for converting from the SGI RGB format to PPM.

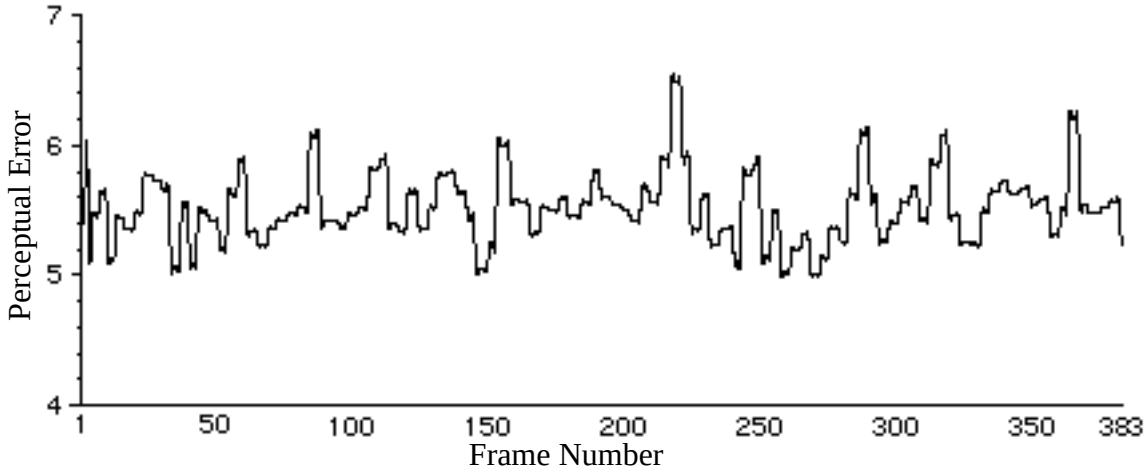

FIGURE 32. DCTune 2.0 perceptual error counts in the clip VC-room.

The error seems fairly constant and follows the amount of motion in the clip. Studying the clips carefully I saw that the peaks in the perceptual error reported by DCTune corresponds to frames where contours of people gets blurry when they move. The minimal error is due to the existence of sharp edges that gets smeared out by the block-DCT. I doubt that I would have noticed the errors at all if I was viewing it at full speed and without access to the original clip to compare it with. The loss of color information was insignificant in the scene.

11. Conclusion

Room-to-room videoconferencing is one of the most demanding future services that a future network infrastructure may have to support. This makes it an interesting service to study. The end users of a videoconferencing system are humans and the system should be designed to help the users to conduct a meeting where the participants are distributed in two or more sites.

Transfer effects and association play an important role in the users decision to adopt a new medium so the audio and video quality of a videoconferencing system should be comparable to that of other services offered in similar environments.

Following this reasoning, ways to provide at least TV resolution audio and video at a reasonably low delay were discussed. Key components in a videoconferencing system, such as compression scheme, computer system design and network issues are treated and the concept of an end-system is introduced to show how the components fit together. A careful end-system design can alleviate bottlenecks in the components.

A room-based videoconferencing system that offer better than broadcast TV resolution and still maintain a good interactivity has been implemented by using standard computers interconnected by an IP-based network. Multipoint scaling issues and multiprogramming overhead is dealt with by using a distributed end-system as opposed to an end-host and by using consumer grade DV equipment and IEEE 1394 firewire busses commonly available in modern computers it is possible to push compression and decompression as far out to the end points as possible and thus lowering the demands on the computer platform itself. The DV compression scheme also proved to be suitable for typical videoconferencing video material.

Tests with the prototype videoconferencing system seems to support earlier observations that even if network bit-rates increases rapidly, the computer design of today has problems to catch up. The designer of systems providing high-bandwidth, networked, real-time, interactive multimedia services like videoconferencing has to be aware of this trend.

11.1. Summary of contributions

- A set of requirements for videoconferencing systems that need to be fulfilled in order to offer better than TV resolution and still maintain a good interactivity.
- Showed a few ways to implement a videoconference system offering high resolution audio and video while still retaining low delay and jitter.
- Highlights bottlenecks that arise when sending high bit-rate, real-time, audio-video data over an IP-based network, and propose a few solutions.

11.2. Future Work

A straightforward continuation of the work presented in this thesis is to examine the relative distortion contribution of packet loss v.s. scaling operations and reduced video frame rates. This includes studying the effectiveness of different loss repair mechanisms, scaling algorithms and frame dropping algorithms when it comes to reduce the perceptual distortion.

The combination of high bit-rate link technologies, bursty transmission of heavily fragmented video frames and a relatively low number of buffers in the network adapters of today can seriously distort video transmission. As was suggested in section 10.7.2, other transmission strategies than burst transmission should be evaluated to avoid unnecessary distortion. As part of this work a study of the transmission pattern of different video streaming applications could be helpful and the result could also be used for better traffic models than the *ping* I used in section 8.4.1.

Most of the end-to-end delay is introduced in the prototype, which calls for further work on how to minimize the length on the delay equalization buffer and overall optimization of the data paths in the prototype.

A 300 MHz CPU upgrade for the SGI O2 has recently been announced. According to the announcement it will increase the computational performance by ca 69%. This upgrade might eliminate the CPU-bound delay and loss contributions for PAL DV format in the prototype.

A similar solution as in the prototype, using MPEG-2 instead of DV, should be possible to implement. Trials of rate-limiting MPEG have been done and it is already in widespread use in many different applications. There is also a DIF specification for sending MPEG-2 over firewire. Using MPEG-2 instead of DV should reduce the bit-rate about 10 times. Unfortunately, MPEG is not as loss tolerant as DV, and as mentioned in section 8.4.2, the MPEG stream has to be parsed to be able to packetize it into RTP.

RTP includes some mechanisms that assumes that a RTP end-system, i.e. an application that generates the content to be sent in RTP packets and/or consumes the content of received RTP packets in a particular RTP session, resides in a single host. How to do, e.g. RTCP signalling and SSRC collision detection when using a distributed end-system?

There are several interesting issues related to multipoint communications and video-audio applications that I do not handle in this work. For example scaling issues, modeling and the interdependancies and grouping of functionality of systems for multipoint communication as well as the effects of different network solutions on human communication.

High Definition Television (HDTV) seems a likely feed in the near future. High-resolution HDTV has a frame size of 1920x1080 and a frame rate of 60 frames per second. Products are under development and will probably be available as part of the Digital TV service. This study doesn't cover HDTV, but a corresponding investigation could easily be conducted based on the methodology outlined in this work.

12. References

3. E. A. Isaacs, T. Morris, T. K. Rodriguez, “A Forum for Supporting Interactive Presentations to Distributed Audiences”, Proceedings of Computer-Supported Cooperative Work ‘94, Chapel Hill, NC, 1994.

4. S. A. Bly, S. R. Harrison, S. Irwin, “Media Spaces: Video, Audio and Computing”, Communications of the ACM, vol 36,No 1, 1993.

5. “Unified Memory Architecture Whitepaper”, Silicon Graphics Inc., 1997.

6. PC 98 System Design Guide Version 1.0, Intel Corporation and Microsoft Corporation, ISBN 1-57231-716-7, 1998.

7. STP2220ABGA Datasheet, Sun Microsystems, 1997, at URL http://www.sun.com/microelectronics/datasheets/stp2220/

8. STP2223BGA Datasheet, Sun Microsystems, 1997, at URL http://www.sun.com/microelectronics/datasheets/stp2223bga/

9. T. P. De Miguel, S. Pavon, J. Salvachua, J. Q. Vives, P. L. C. Alonso, J. Fernandez-Amigo, C. Acuna, L. Rodriguez Yamamoto, V. Lagarto, J. Vastos, “ISABEL Experimental Distributed Cooperative Work Application over Broadband Networks”, DIT/UPM Espana, 1993.

10. “MDL Corporation: Communique! Product Information” at URL http://www.mdlcorp.com/Insoft/Products/C/C.html

11. T. Dorcey, “CU-SeeMee desktop videoconferencing software”, ConneXions, 9(3), March, 1995

12. B. Cain, S. Deering, A. Thyagarajan, “Internet Group Management Protocol, version 3”, Internet Engineering Task Force Request For Comments XXXX, 1997. Work in progress.

13. S. Deering, “Host Extensions for IP Multicasting”, Internet Engineering Task Force Request For Comments 1112, 1989.

14. T. Maufer, C. Semeria, “Introduction to IP Multicast Routing”, Internet Engineering Task Force Request For Comments XXXX, 1997. Work in progress.

15. S. Deering, “Multicast Routing in a Datagram Internetwork”, PhD thesis, Stanford University, 1991.

16. V. Jacobson, “Multimedia Conferencing on the Internet”, SIGCOMM ‘94 Tutorial, University College London, 1994.

17. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications”, Internet Engineering Task Force Request For Comments 1889, 1996.

18. H. Schulzrinne, “Internet Services: from Electronic Mail to Real-Time Multimedia”, Kommunikation in Verteilten Systemen (KIVS) ‘95, Informatik aktuell Series, Springer Verlag, 1995.

19. H. Schulzrinne, “RTP Profile for Audio and Video Conferences with Minimal Control”, Internet Engineering Task Force Request For Comments 1890, 1996.

20. H. Schulzrinne, “RTP: About RTP and the Audio-Video Transport Working”, at URL http://www.cs.columbia.edu/~hgs/rtp

21. M. Handley, “Guidelines for Writers of RTP Payload Format Specifications”, Internet Engineering Task Force Request For Comments XXXX, Work in progress, 1999.

22. S. Hosgood, “All You Ever Wanted to Know About NICAM but were Afraid to Ask”, at URL http://iiit.swan.ac.uk/~iisteve/nicam.html, 1997.

23. “MPEG Home Page”, at URL http://www..cselt.stet.it/mpeg/

24. G.D. Ripley, “DVI - A Digital Multimedia Technology”, Communications of the ACM, July 1989.

25. “Specifications of Consumer-Use Digital VCRs using 6.3 mm magnetic tape”, Blue Book of dissolved HD Digital VCR Conference, December 1994.

26. P. Creek, D. Moccia, “Digital Media Programming Guide”, Document Number 007-1799-060, Silicon Graphics Inc., 1996.

27. M. Liou, “Overview of the px64 kbit/s Video Coding Standard”, Communications of the ACM, Vol. 34, No 4, April 1991.

28. S. Forsberg, “Marknadsgenomgång av ISDN”, Nätverksguiden, December 1996

29. International Telecommunication Union Telecommunication Standardization Sector (ITU-T) Recommendation H.263, “Video Coding for Low Bitrate Communication”, ITU-T, March 1996.

30. G. Cote, B. Erol, M Gallant, F. Kossentini, “H.263+: Video Coding at Low Bit Rates”, IEEE Transactions on Circuits and Systems for Video Technology, November 1998

31. International Telecommunication Union Telecommunication Standardization Sector (ITU-T) Recommendation H.263 Version 2, “Video Coding for Low Bitrate Communication”, ITU-T, September 1997.

32. D. L. Le Gall, “MPEG: A Video Compression Standard for Multimedia Applications”, Communications of the ACM, April 1991.

33. T. Sikora, “MPEG Digital Video Coding Standards”, Digital Electronics Consumer Handbook, MacGraw Hill Company, 1997.

34. A. Puri and A. Eleftheriadis, “MPEG-4: A Multimedia Coding Standard Supporting Mobile Applications”, ACM Mobile Networks and Applications Journal, Special Issue on Mobile Multimedia Communications, 1998.

35. William B. Pennebaker, Joan L. Mitchell, “JPEG: Still Image Data Compression Standard”, Van Nostrand Reinhold, 1993.

36. Gregory K. Wallace, “The JPEG Sill Picture Compression Standard”, Communications of the ACM, Vol 34, No. 1, pp. 31-44, April 1991.

37. L. Berc, W. Fenner, R. Frederick, S. McCanne, “RTP Payload Format for JPEG-compressed Video”, Internet Engineering Task Force Request For Comments 2035, 1996.

38. “The JPEG File Interchange Format”. Maintained by C-Cube Microsystems, Inc., and available in ftp://ftp.uu.net/graphics/jpeg/jfif.ps.gz.

39. R. Jennings, “Consumer and Professional Digital Video Recording and Data Formats”, Special Edition on Using Desktop Video, Que Books, Macmillan Computer Publishing, 1997.

40. K. Jeffay, D.L. Stone, F. Donelson Smith, “Kernel Support for Live Digital Audio and Video”, Computer Communication, July/August 1992.

41. “MPEG-Audio FAQ”, at URL http://www.tnt.uni-hannover.de/project/mpeg/audio/faq/

42. F. Bock, H. Walter, M. Wilde , “A new distortion measure for the assessment of decoded images adapted to human perception”, IWISP’96, Manchester, 1996.

43. International Telecommunication Union Telecommunication Standardization Sector (ITU-T) Recommendation G.114, “Transmission Systems and Media, General Characteristics of International Telephone Connections and International Telephone Circuits, One-Way Transmission Time”, ITU-T, February 1996.

44. A. Watson, M. A. Sasse, “Measuring Perceived Quality of Speech and Video in Multimedia Conferencing Applications, Proceedings of ACM Multimedia ‘98, 1998.

45. C. J. van den Branden Lambrecht, “Perceptual Models and Architectures for Video Coding Applications”, PhD Thesis, Ecole polytechnique fédérale de Lausanne, 1996.

46. A. B. Watson, “Toward a perceptual video quality metric”, IS&T/SPIE Conference on Human Vision and Electronic Imaging III, San Jose, California, January 1998.

47. S. R. McCanne, “Scalable Compression and Transmission of Internet Multicast Video”, PhD Thesis, Report No. UCB/CSD-96-928, Computer Science Division University of California Berkeley, 1996.

48. “The MASH Project Home Page”, at URL http://mash.cs.berkeley.edu/mash/

49. S. McCanne, V. Jacobson, “vic: A Flexible Framework for Packet Video”, Communications of the ACM, November 1995.

50. “Mbone Conferencing Applications”, at URL http://www-mice.cs.ucl.ac.uk/multimedia/software/

51. V. Hardman, M. A. Sasse, M. Handley, A. Watson, “Reliable Audio for use over the Internet”, In Proceedings INET’95, Hawaii, 1995.

52. W. Wiegler, P. Donham, “CU-SeeMe version 3.1.2 User Guide”, White Pine Software Inc., 1998.

53. J. Quemada, T. de Miguel, A. Azcorra, S. Pavon, J. Salvachua, M. Petit, D. Larrabeiti, T. Robles, G. Huecas, “ISABEL: A CSCW Application for the Distribution of Events”, Departamento de Ingenieria de Sistemas Telematicos (DIT), Technical University of Madrid (UPM), 1996.

54. “CosmoNet”, at URL http://www.it.kth.se/labs/ts/bb/cosmonet/CosmoNet.html

55. Akimichi Ogawa et al. “Design and implementation of DV Stream over Internet”, Proceedings of Internet Work Shop 99, 1999.

56. "DV Stream on IEEE1394 Encapsulated into IP", at URL http://www.sfc.wide.ad.jp/DVTS/

57. G. Karlsson, C. Katzeff, "The Role of Video Quality in Computer Mediated Communication", http://www.sisu.se/projects/video/index.html, 1997.

58. G. Karlsson, "On the Characteristics of Variable Bit-Rate Video", Proceedings of the Thirteenth Nordic Teletraffic Seminar, Trondheim, Norway, pp. 293-304, August 20-22, 1996.

59. J. Hartung, A. Jacquin, J. Pawlyk, K. Shipley, "A real-time scalable sofware video codec for collaborative application over packet networks", Proceedings ACM Multimedia, 1998.

60. SunVideo User's Guide, Sun Microsystems Computer Corporation, 1994.

61. O. Hagsand, "Computer and Communication Support for Interactive Distributed Applications", PhD thesis, Department of Teleinformatics, Royal Institute of Technology, TRITA-IT R 94:31, ISSN 1103-534X, ISRN KTH/IT/R--94/31--SE, 1995.

62. U. Schwantag, "An Analysis of the Applicability of RSVP", Diploma Thesis at the Institute of Telematics, Universität Karlsruhe, 1997.

63. D. Hoffman, G. Fernando, V Goyal, M. Civanlar, "RTP Payload Format for MPEG1/MPEG2 Video", Internet Engineering Task Force Request For Comments 2038, 1998.

64. K. Kobayashi, A. Ogawa, S. Casner, C. Bormann, "RTP Payload Format for DV Format Video", Internet Engineering Task Force Request For Comments XXXX, Work in progress, 1999.

65. T. Öbrink, "Prototype intro", at URL http://www.it.kth.se/~nv91-tob/Report/Prototype

66. G. A. Hoffman, "IEEE 1394, the A/V Digital Interface of Choice", at URL http://www.1394ta.org/Technology/About/digital_av.htm.

67. E. Pelletta, "Digital Video (DV) Technology in SSVLNet", at URL http://www.it.kth.se/~enrico

68. T. Poles, K. Elezaj, "Voice over IP/Ethernet", MSc Thesis, IT98/34, Department of Teleinformatics, Royal Institute of Technology, 1998.

69. The Naval Research Laboratory (NRL) "Multi-Generator" (MGEN) Toolset. URL: http://manimac.itd.nrl.navy.mil/MGEN/index.html

70. D. L. Mills, "On the Chronometry and Metrology of Computer Network Timescales and their Application to the Network Time Protocol", ACM Computer Communications Review 21, 5, 1991.

71. D. L. Mills, "Network Time Protocol (Version 3), Specification, Implementation and Analysis", Internet Engineering Task Force Request For Comments 1305, 1992.

72. D. L. Mills, "Internet Time Synchronization: the Network Time Protocol", IEEE Transactions on Communications, vol. 39, no. 10, 1991.

73. Roger Jennings, “DV vs. Betacam SP: 4:1:1 vs. 4:2:2, Artifacts and Other Controversies”, Que Books, Macmillan Computer Publishing, 1997.

74. “The Joint EBU SMPTE Task Force on Harmonised Standards for the Exchange of Television Programme Material as Bit Streams: Final Report”, SMPTE Journal, vol 107, no 9, September 1998.

75. K. M. Zuberi, K. G. Shin, “An Efficient End-Host Protocol Processing Architecture for Real-Time Audio and Video Traffic” NOSSDAV’98, 1998.

76. P. Bahl, P. S. Gauthier, R. A. Ulichney, “Software-only Compression, Rendering and Playback of Digital Video”, Digital Technical Journal, Vol. 7, No. 4, 1995.

77. K. Fall, J. Pasquale, and S. McCanne, “Workstation Video Playback Performance with Competitive Process Load”, Proceedings of the Fifth International Workshop on Network and OS Support for Digital Audio and Video. April, 1995. Durham, NH.

78. “vic: Research Projects”, at URL http://www-nrg.ee.lbl.gov/vic/research.html

13. Bibliography

K. E. Finn, A. J. Sellen, S. B. Wilbur, “Video-Mediated Communication”, Lawrence Erlbaum Associates, Mahwah New Jersey, 1997.

F. Fluckiger, “Understanding Networked Multimedia”, Prentice-Hall, 1995.

C. Katzeff, K. Skantz, “Talande huvuden och dubbningssjuka”, publikation 97:02, Svenska Institutet för Systemutveckling, 1997.

A. J. Dix, G. D. Abowd, R. Beale, J. E. Finley, “Human-Computer Interaction”, 1st Ed., Prentice Hall, 1993.

W. Stallings, “Data and Computer Communications“, Prentice-Hall Inc., 1997.

R. Steinmetz, K. Nahrstedt, “Multimedia: Computing, Communications & Applications”, Prentice Hall Inc., 1995.

J. F. Koegel Buford, “Multimedia Systems”, Addison-Wesley, 1994.

P. H. Young, “Electronic Communication Techniques”, Macmillan Publishing Company, 1991.

W. Kou, “Digital Image Compression Algorithms and Standards”, Kluwer Academic Publishers, 1995.

A. Terminology and Abbreviations

This paper contains a lot of terminology from different fields and it's not easy to keep track of all of them. Due to the interdisciplinary nature of this piece of work, there naturally is no fixed terminology and thus every definition in this appendix may be subject to debate in some way or another. Many of the terms have quite different meaning in different disciplines, and I don't claim to have complete knowledge of all those. Instead, I hope this list of definitions will help in avoiding unnecessary misunderstanding when reading this paper.

Active lines. The lines in an analog video signal that is visible on the screen.

ADU. *Application Data Unit.* A chunk of data that can be processed out-of-order with respect to other ADUs.

ALF. *Application Level Framing.* The concept of *Application Level Framing* (ALF) lets the application deal with data loss according to its needs and capabilities by forming application specific ADUs for transmission.

Analog signals. Physical measures which varies continuously with time and/or space. They can be described by mathematical functions of the type s=f(t), s=f(x, y, z) or s=f(x, y, z, t). A sensor detects a physical phenomenon and transforms it into a measure, usually an electrical current or -tension. The measured values are expressed with an accuracy which is characteristic for the sensor. In signal processing the value is called amplitude.

Artifacts. Visible erors which appear unnatural.

Aspect ratio. The ratio of the width to the height of a frame. Expressed as X : Y where X and Y are the lowest natural numbers such that X/Y = x/y where x is the width and y is the height of the frame.

Audio-video conferencing. Usually abbreviated to videoconferencing. The objective is to support a meeting between more than two remote participants. If biparty, the conference connects groups of people; if multiparty, it may connect a mixture of groups and individuals. Participants may gather either in an office using desktop or rollabout systems, or in a meeting room. To support a meeting situation, documents need to be exchanged, either on paper or in projected or in electronic form.

Awareness. Using different ways to provide information on which other people are around and information about them.

Bandwidth. A range of frequencies of a transmission media.

Binocular convergence. Using the angle between the line of sight of each eye to determine the distance to the point of focus.

Binocular parallax. Using the differences between the two images due to the space between the eyes to determine the distance to the point of focus.

Bitmap. A spatial two-dimensional matrix made up of individual pixels.

Bit-rate guarantee. The type of guaranteed transmission capacity a network can give to an end-system. Can be either None, Reserved, Allocated or Dedicated.

Bit-rate regulation. Regulates the bit-rate generated to be within a certain (small) interval.

Burstiness. The degree of bit rate variation of a data stream. Common metrics are peak bit rate (PBR), the peak duration, the mean bit rate (MBR) and the ratio between MBR and PBR.

CBR. *Constant Bit Rate.*

CD-DA. *Compact Disc-Digital Audio.*

Chrominance. In analog broadcast television, chrominance signals are constructed by linear combinations of color difference signals.

CIF. *Common Intermediate Format.* A video format defined for internal use in the H.261 codec. Gives a maximum 352x288 resolution of the luminance signal.

CNAME. *Canonical Endpoint identifier* or canonical name. A text string that uniquely identifies a participant within all sessions. The CNAME identifier should have the format *user@host*, or just host. Two examples given in [15] are *doe@sleepy.megacorp.com* and *doe@192.0.2.89*.

Codec. A system that bundles both the functions of coding and decoding is called a coder-decoder, abbreviated codec.

Coding. Associate each quantized value with a group of binary digits, called a codeword.

Color difference signal. A color difference signal is formed by substracting the luminance signal from each of the primary color signals.

Color space. The space of possible colors that can be generated by a certain combination of image components. A few examples are monochrome, YCrCb and RGB.

Competitive load. Other processes/traffic than the ones under study competing for the

same shared resources.

Compression. Compression refers to the algorithms used to reduce the bit rate of a digital signal.

Computer. By computer we mean any technology ranging from general desktop computer, to a large scale computer system, a process control system or an embedded system.

Computer-assisted circuit telephony. A computer is used for part or all of the functionality of a terminal connected to a circuit telephony system. It can also provide a multitude of complementary services, e.g. directory functions, message recording, call distribution etc..

Computer display scan rate. The frequency at which the screen is refreshed by the electronics of the monitor. Usually in the order of 60 to 70 Hz.

Continuous media. Data are generated at a given, not necessarily fixed, rate independent of the network load and impose a timing relationship between sender and receiver, that is, the data should be delivered to the user with the same rate as it was generated at the receiver.

CPS. *Constrained Parameter Set.* Minimal requirement of MPEG-1 decoders.

CSCW. *Computer-Supported Cooperative Work.* A research field encompassing both the development of groupware and studies of the effects of computers on cooperative working in general.

DAT. *Digital Audio Tape.*

Data alteration. The most frequent alteration concerns inversion of bits, or loss of trailing or heading parts in data blocks or packets. In modern networks, alteration is the least frequent form of error.

Data duplication. The same data are received unexpectedly more than once by the receiver. This is a rather rare incident in practice.

Data loss. Data may be discarded by network components because of detected data alteration or most frequently due to internal network congestion affecting nodes or transmission lines.

DCT. *Discrete Cosine Transform.*

Dedicated rooms mode. When the videoconferencing service is delivered only to a dedicated room.

Degradation mean opinion score (DMOS). Subjective measures including ratings of perceived quality degradation compared to an original. DMOS uses a five-grade scale ranging from 1, Very annoying, to 5, Inaudible. The DMOS value is extracted from the results of an Degradation Category Rated (DCR) test performed on 20 to 60 untrained persons.

Delay equalization. Also called delay compensation. The real-time transmission of continuous media over networks is very sensitive to delay variation. To overcome delay variations, an additional offset delay is inserted at the sink end to achieve a smooth playout. This technique may add a substantial component to the overall latency between the source and the final playout of the sound. In theory the delay offset should match the upper bound of the delay variation, but in practice interactive applications will require an upper bound on the end-to-end delay.

Desktop mode. When the videoconferencing service is delivered on the end-user's desktop.

Desktop videoconferencing system. A regular desktop computer system equipped with software and hardware for real-time, interactive tranfer of audio, video and often also data sharing.

DIF. Stands for *Digital Interface.* A communication protocol for carrying isochronous data over the IEEE 1394 high performance serial bus.

Digital signal. A time-dependent or space-dependent sequence of values coded in binary format.

Digitization. The transformation of analog signals into digital signals. Consists of Sampling, followed by Quantizing, followed by Coding. Also called encoding.

Dithering. Algorithms used to minimize visual artifacts caused by compression and image transformations.

DSP. *Digital Signal Processing.*

DVQ. *Digital Video Quality.* A perceptual metric specialized for measuring the perceptual distortion introduced by the DCT-part of a video compression scheme.

DVTS. *DV Transmission System.*

Echo. The hearing mechanism normally filters out the echo of one's own voice when speaking. Unfortunately, this filter doesn't work if the echo is delayed long enough.

EDF. *Earliest Deadline first.* A priority-based scheduling algorithm giving the process with closest deadline the highest priority.

End-system. I define an *end-system* as; the computers, the network connection, and audio-video equipment used at one of the participating sites in a videoconference to mediate audio and video to and from the other sites the videoconference.

End-to-end delay. I use this as the time between capture/sampling and display/playout of a media.

Error rate. The error rate is a measure of the behaviour of the network with respect to alteration, loss, duplication, or out-of-order delivery of data. Metrics used are the bit error rate (BER), the packet error rate (PER), the cell error rate (CER), the packet loss rate (PLR) and the cell loss rate (CLR).

FEC. *Forward Error Correction.* Enable a limited error correction at the receiver by adding a parity code that can be used to reconstruct damaged data. Recent work has extended this technique to be applicable also for repair of limited packet loss.

Flow control waiting time. The time the source has to wait for the network to be ready before being authorized to transmit.

Frame. A complete and individual view, and part of a succession of displayed views.

Frame rate. The rate at which the frames are displayed in frames per second (fps). Also called temporal resolution.

Frame size. The number of pixels per frame. Denoted X * Y where X is the number of pixels per line and Y is the number of lines. Also called the spatial resolution and frame format.

Full connection. Every end-system is directly connected with a physical cable to all the others, requiring n^2 - n cables to fully interconnect n systems.

GUI. *Graphical User Interface.*

HCI. *Human-Computer Interaction.* A research field studying the interaction between people and computers.

IDT. *Inter-frame Display Time.* The time interval between the start of video frames.

IEEE 1394. A high-performance serial bus technology also called *FireWire.*

IGMP. *Internet Group Management Protocol.* An internet protocol for handling IP multicast group membership on a network segment.

Image components. A pixel is encoded using either luminance and chrominance signals (YIQ or YUV), luminance and color difference signals (YC_rC_b) or RGB signals. These building blocks are called image components by a common name.

Initialization delay. The delay from “switchin on” something to the time when it is ready for use.

Interaction. By interaction we mean any communication betweeen a user and computer, be it direct or indirect. Direct interaction involves a dialogue with feedback and control throughout performance of the task. Indirect interaction may involve background or batch processing.

Interactive threshold. The time that an average human can wait for a response triggered by an action before drawing the conclusion that something is broken. About one second is mentioned in HCI.

Interlacing. Every frame is divided in two fields, the even field consists of the even-numbered lines and the odd field is composed of the odd-numbered lines of the frame. The resolution loss is in the order of one-third compared to progressive scan, but it saves bandwidth in analog broadcast.

Intermedia synchronization. Timing relationships between different streams is restored. A typical case of intermedia synchronization is synchronization between audio and motion video.

Intramedia synchronization. Timing relationships between elements in a media stream is restored within the individual streams at playout. Also called streaming.

Intra-stream dependencies. All compression mechanisms imply that blocks carry some form of updates, so that the data of a block generated at time t carries information affecting blocks generated within an interval $\{t - \Delta t_1, t + \Delta t_2\}$.

Isochronism. An end-to-end network connection is said to be isochronous if the bit rate over the connection is guaranteed and if the value of the delay variation is guaranteed and small.

Jitter. Variance of the IDT. Often used as a synonym for variance of the network delay as well.

jnds. *just-noticeable differences.*

KTH/IT. Department of Teleinformatics at the Royal Institute of Technology.

Lip-synchronization. Intermedia synchronization between audio and motion video.

Luminance. The cumulative response of the eye to all the wavelengths contained in a given source of light. Luminance is represented by the function $Y = \int C(\lambda) V(\lambda) d\lambda$, where $C(\lambda)$ is the spectral distribution and $V(\lambda)$ is the spectral response. Luminance is usually denoted by Y.

MBone. *Multicast Backbone.* A virtual network backbone built on top of the unicast Internet using IP-in-IP tunnels bridging together multicast-enabled subnets.

MCU. *Multiparty Conferencing Unit.* A VBX that switch digital signals directly.

Mean opinion score (MOS). Subjective measures including ratings of perceived quality on a five-grade scale ranging from 1, Bad, to 5, Excellent. The MOS value is extracted from the results of an Absolute Category Rated (ACR) test performed on 20 to 60 untrained persons.

Mesh. A set of interconnected stars with redundant interconnecting links, so that alternative routes exists between two end-systems.

MGEN. The *Multi-Generator* (MGEN) toolset from the U.S. *Naval Research Laboratory* (NRL).

Mirror effect. Most users find it disturbing if their captured image is displayed directly without left and right being inverted. People, when viewing their own faces, are accustomed to the mirror effect.

Motion parallaxes. Using the relative motion of objects and parts of objects to determine the distance to a point.

MSE. *Mean Squared Error.*

Multimedia conferencing. When a conference integrates text, graphics, or images-based conversation with audio or video-based dialog, it is called a multimedia conference.

Network connection set-up delay. The time it takes to set up an end-to-end connection between two end-systems. This only applies to those networks which are aware of end-to-end-system connections, such as ATM, ST-II, or satellite-based communications.

NICAM. *Near-Instantaneous Companded Audio Multiplex.* A system for broadcasting digital stereo audio multiplexed with the analog video and audio in the TV signal.

NLE. *Non-Linear-Editing.* As opposed to linear video editing, NLE allows for editing anywhere in the image data at any time.

Nyquist Theorem. The Nyquist classical theory requires that, to faithfully represent an analog signal, the sampling frequency should be equal to or greater than twice the highest frequency contained in the sampled signal. Studies have, however, shown that under certain circumstances, lower sampling frequencies can in practice be used.

Out-of-order delivery of data. Long-haul packet networks, in particular, may have alternate routes between two end-systems. When failures or congestion occur, alternate routes may be involved, and route oscillations may happen. As not all routes have the same transit delay, packets may be delivered in a different order than they were emitted.

Packet voice conversation. Same as computer-assisted telephony, but the underlying network is packet switched.

Perceived resolution. Determined by the frame size, the pixel depth, the frame rate and the subsampling scheme used.

Perceptual metrics. A metric system used by some measurement tools that mimic human senses. Perceptual metric values, in turn, can be mapped to subjective rating scores.

Personal space. The closest distance around the body that humans can accept other humans to be without feeling uncomfortable.

Physical jitter. The variation of the delay generated by the transmission equipment, such as faulty behaviour of the repeater's reshape signals, crosstalk between cables may create interference, electronic oscillators may have phase noise and changes in propagation delay in metallic conductors due to temperature changes.

Pixel. Stands for picture element and is the smallest element of resolution of the image. Each pixel is represented by a numerical value, the amplitude. The number of bits available to code an amplitude is called the amplitude depth, pixel depth or color - or chroma resolution. The numerical value may represent black/white in bitonal images, level of gray in grayscale images or color attributes in a color image.

Playout. The process of transforming an digital representation of a signal into analog form.

Progressive scan. The screen is refreshed progressively line by line, each line being scanned from left to right.

PSNR. *Peak-to-peak Signal to Noise Ratio.*

Quantization. Converting the sampled values into a signal which can take only a limited number of values.

Real-time data. Real-time data imposes an upper bound on the delay between sender and receiver, that is, a message should be received by a particular deadline. Packets that miss their deadline are considered lost (late loss), just as if they had been dropped at a switch or router.

Rendering. The technique used for the display of digital still or moving images. Rendering refers to the process of generating device-dependent pixel data from device-independent sampled image data, including dithering.

Resolution. One parameter of resolution is the frame size, another parameter is the pixel depth.

Red-Green-Blue (RGB). The Commission Internationale de l'Eclaire (CIE) has defined a Red-Green-Blue system by reference to three monochromatic colors. The respective wavelengths are Red = 700 nm, Green = 546 nm, and Blue = 436 nm. The television standards have adopted triplets which are generally slightly different from that of the CIE.

Roll-about. A circuit-switched-based videoconference system that can be moved between meeting rooms.

RTE. *Real-Time Environment.* Priority-based scheduling of data according to it's inherent timing requirements.

Sampling. Retaining a discrete set of values from an analog signal. Also called capture.

Sampling rate. The periodicity of sampling is in general constant and called the sampling frequency or sampling rate.

SDES. *Source Description.* RTCP message for linking a SSRC with a CNAME.

Sender-based copying. Each sender has to send identical copies to all other participants.

SIF. *Source Input Format.* The most common format for MPEG-1 Video material. The luminance signal has a resolution of a quarter of Television, i.e. 352x240 (NTSC), 352x288 (PAL).

Silence suppression. Only audio that is louder than a certain threshold is transmitted.

SNR. *Signal to Noise Ratio.*

Soft real-time. A small amount of data can be late without catastrophic consequences.

Spectral distribution. Most light sources are composed of a range of wavelengths, each having its own intensity. This is called the spectral distribution of the light source and is represented by the function $C(\lambda)$.

Spectral response. How sensitive the human eye is to light of a certain wavelength. The response of the human vision to a wavelength λ is represented by the function $V(\lambda)$.

Speech synthesis coder. A speech encoder that use a model of the vocal tract to analyse and produce a code that describes the incoming speech.

SSRC. *Synchronisation Source identifier.* Part of RTP header that uniquely identifies packets from a certain source within a certain RTP session.

Star. All end-systems connect to a star point. At the star point, there is a system called a switch which can route the information from one cable to another.

Store-and-forward switching delays. Delays caused by internal node congestion.

SU/EE. Department of Electrical Engineering at Stanford University.

Subsampling. Fewer samples per line is taken, and sometimes fewer lines per frame. The ratio between the sampling frequency of the luminance and the sampling frequency of each of the color difference signals have to be an integer, resulting in all components are sampled at locations extracted from a single grid. The notation is of the type; <Y sampling frequency>:<C_d sampling frequency> for a single color difference scheme, and <Y sampling frequency>:<C_{d1} sampling frequency>:<C_{d2} sampling frequency> for a luminance-chrominance scheme.

Symmetrical meeting. A meeting with balanced contributions from all participating sites.

Synchronous data. Periodically generated bits, bytes or packets that have to be regenerated with exactly the same period at the receiver. Synchronous data has a constant bit rate.

Teleconferencing. Computer-based conferencing at a distance. A generic name for any application which supports real-time bidirectional conversation between two groups or several groups of people. Videoconferencing and shared whiteboards are examples of specific teleconferencing applications.

Throughput. The rate at which two ideal end-systems can exchange binary information. Also called bit rate, data rate, transfer rate and bandwidth. The unit is bits per second (bps). In the cases where networks only handle fixed-sized blocks, other units may be used too, e.g. cell rate in ATM networks.

Transcoder MCU. A MCU capable of translating between different audio and video encoding and compression schemes.

Transit delay. The time elapsing between the emission of the first bit of a data block by the transmitting end-system and its reception by the receiving end-system. Also called latency. If the end-systems are connected by a single link, then this is the same as the propagation delay of the medium.

Transit delay variation. The variation over time of the network transit delay. Usually measured as the difference between experienced delay and some target delay for the data flow. Other definitions are based on the difference between the longest and the shortest transit delays observed over a period of time. Also called jitter and delay jitter.

Transmission delay. The time necessary to transmit all the bits of a block. For a given block size this only depends on the acess delay.

Tree. A set of interconnected stars, so that only one route exists between two end-systems.

Turn-taking. Subconscious negotiations on who should talk at a single moment.

UMA. *Unified Memory Architecture.* A shared memory computer system architecture used in the SGI O2.

USB. *Universal Serial Bus.*

User. By user we mean an individual user, a group of users working together, or a sequence of users in an organization, each dealing with some part of the task or process. The user is whoever is trying to get the job done using the technology.

VBX. *Video Branch eXchange.* Also called videoconferencing hub or video-hub for short. A VBX is used to provide a star point which acts as a switch between all participating video-codecs.

VIC. The UCB/LBNL *VIdeoConference* tool.

Video-codec. A packaged, stand-alone videoconference system.

Videoconference. An abbreviation of audio-video conferencing. The objective of a videoconference is to support a meeting between more than two remote participants.

Videoconference studio. Dedicated meeting rooms, equipped with analog audio-visual devices, digitizers and compressor/decompressor systems as well as a connection to a network.

Video distribution. Traditional broadcast-like, one-way video.

Video format. Consists of resolution, frame rate, aspect ratio and subsampling scheme.

Videophony. Telephony with motion video. Videophones may be video-extended telephone sets, so called video dialtones, or a computer equipped with necessary hardware and software.

Video-switch. Also called video mixer. A VBX that convert digital signals into analog before switching.

Voice-activation. Only forward video from the party generating sound at a given moment.

YCbCr. The image components of ITU-R BT.601 consisting of a luminance signal and two color-difference signals.

YIQ. The transform used in NTSC to convert RGB into a luminance and two chrominance signals.

YUV. The transform used in PAL to convert RGB into a luminance and two chrominance signals.

B. The effect of local rendering in VIC

In most desktop videoconferencing solutions the video that is captured and sent can be shown on the local screen. According to [47] image rendering sometimes accounts for 50% or more of the execution time. In [74] were found that between 27 - 60 % of the CPU time on a Digital 266 MHz Alphastation with PCI bus where needed for software-only decompression and playback of MPEG-1, MJPEG and Indeo video and that rendering accounts for around one third of the decompression and playback time. Since these papers were written in 1994 -1995 I had to check if this still holds with today's platforms.

In this test I used a Sun Ultra2 Creator 3D with a SunVideo 2 capture card and a Sun camera giving a PAL feed to the capture card. The software used were the UCB/LBL VIdeoConferencing tool (VIC), which is widely used on the Multicast Backbone (MBone). When capturing video and transmitting it in compressed form it is possible to display the captured video. The SunVideo and XIL grabbers in VIC delivers SIF format video, so it has to be converted to CIF before being displayed. How this operation is implemented in VIC is not specified in [47], [75] or [76].

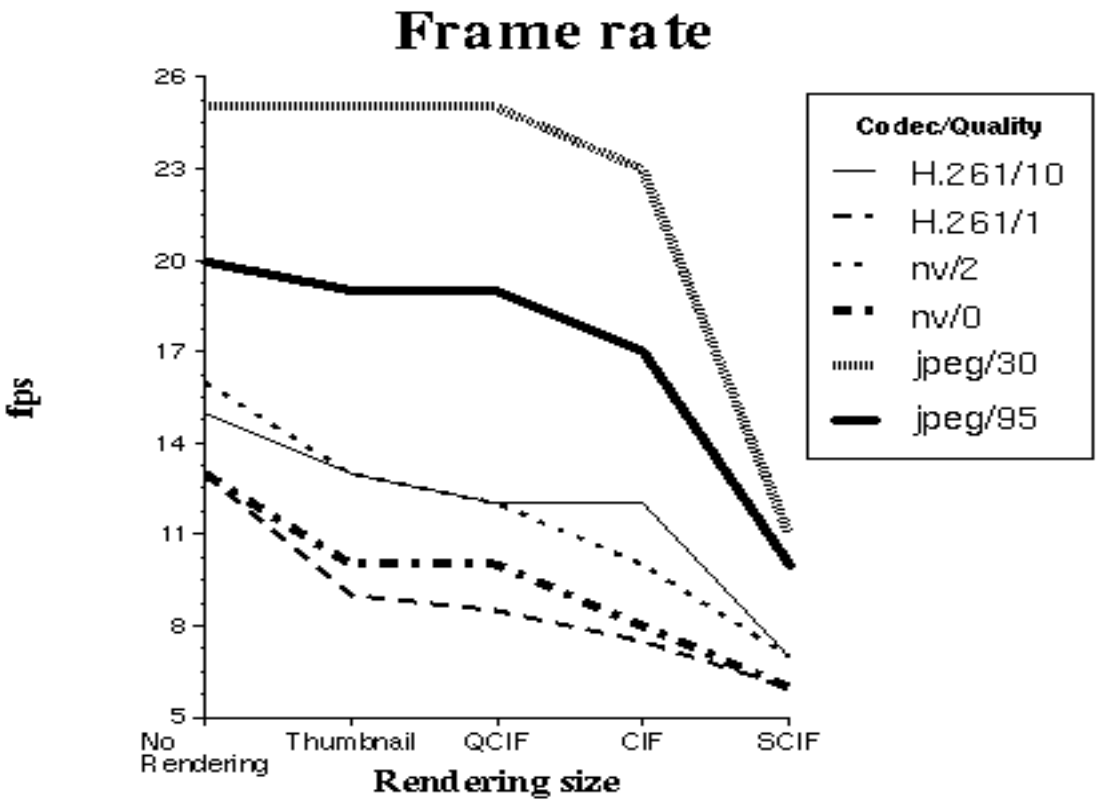

FIGURE 33. Framerate degradation due to local rendering.

After doing some worst case tests I found that the performance of the tool degrades up to 56% on the given platform depending on the size of the rendered picture and the codec used. To check how the performance degrades I incrementally increased the size of the local image. The results from these tests are shown in Figures 33 and 34 below.

The codecs tested were H.261 with default(10) - and maximum(1) quality, nv with default(2) - and maximum(0) quality and jpeg with default(30) - and maximum(95) quality. Other codecs supported by the VIC tool was nvdct and cellb, but these were found to give too low subjective picture quality compared to the other codecs to be considered in the test.

I also found that the frame rate degradation varied somewhat depending on which coding standard was used. The deviation between the most degraded - and the least degraded coding scheme was 6% giving a range between 50 % - 56 % maximum framerate degradation. The bit rate degradation varied between 0 % - 61% maximum degradation. The constant bitrate for MJPEG can be explained as due to the hardware support for JPEG compression in the SunVideo 2 card. That the frame rate is falling also for MJPEG is harder to explain, but is probably a consequence of the overload handling of the rendering part of VIC as reported in [75]. For nv and H.261 the maximum bit rate degradation varied between 56 % - 61 %.

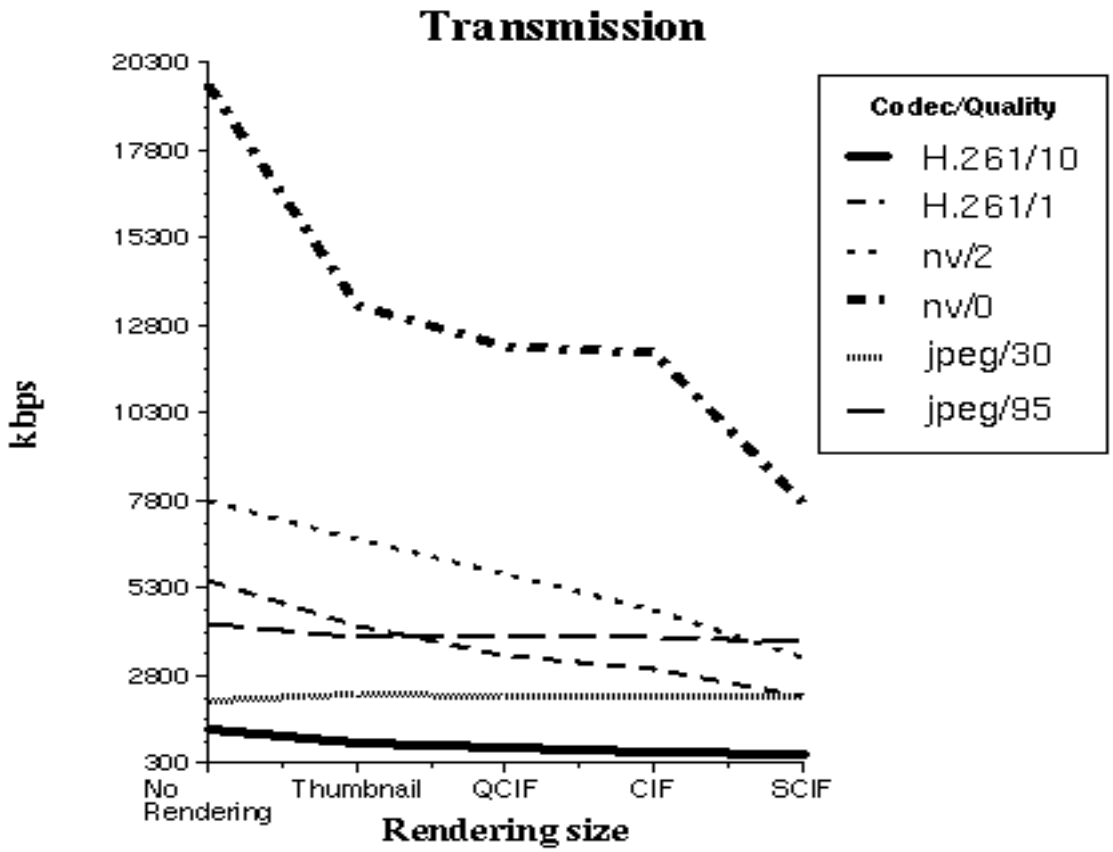

FIGURE 34. Transmission rate degradation due to local rendering

C. Acceptable amount of jitter

In a laboration in the Telesystems basic course in 1996 and 1997 one of the exercises included determining the maximum acceptable amount of jitter in a 30 s long 8 bit PCM recording. 23 groups of 2 to 4 students used a program that segments the audio samples into packets of 160 bytes which were then delayed by different amounts and the resulting signal is played.

Packet delay was generated from an one-sided normal distribution with mean 0, and the standard deviation, in ms, of the distribution was given as a parameter to the program. The groups tried different values for the standard deviation to find the point where the resulting signal became unacceptable. The values reported were:

25 30 40 20 3 80 10 30 30 10 100 2 80 20 30 30 15 20 20 25 5 150 35

Arithmetic mean is computed to 35 ms while the median is 25 ms.

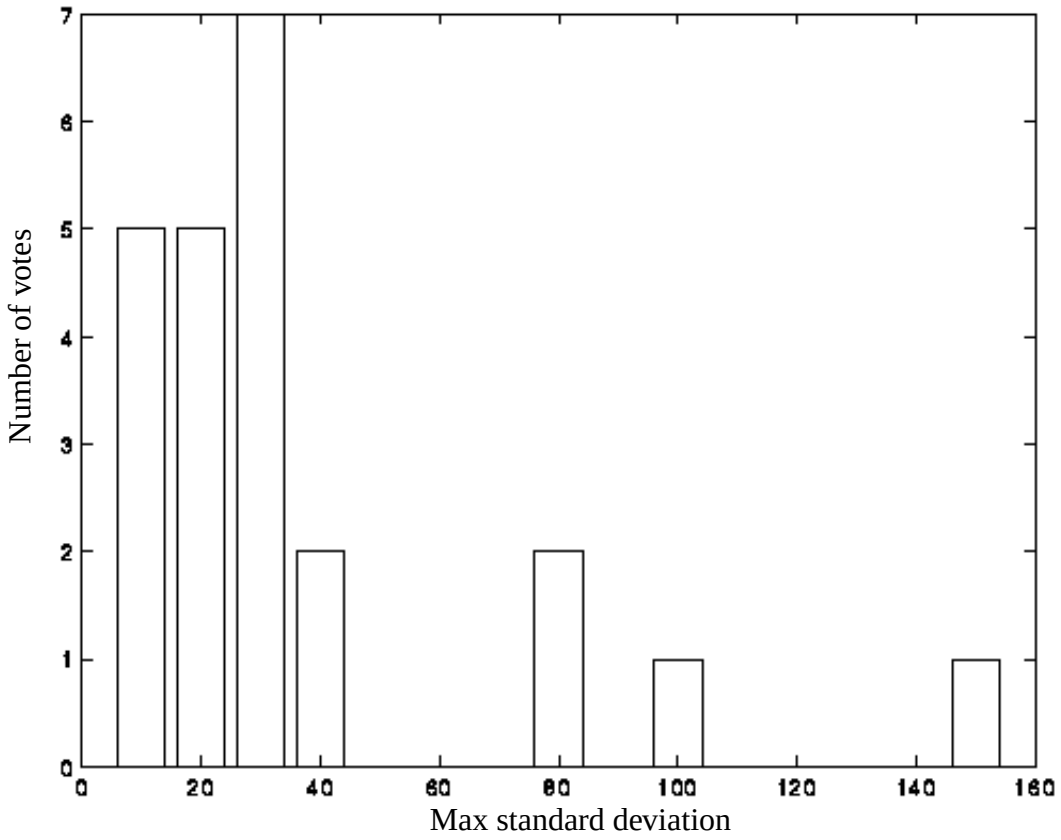

FIGURE 35. Distribution of votes over maximunim acceptable standard deviation.

TRITA-IT R 99:05
ISSN 1103-534X
ISRN KTH/IT/R--
99/05--SE

KUNGL TEKNISKA HÖGSKOLAN
Royal Institute of Technology

Telecommunication Systems
Laboratory
Department of Teleinformatics
Royal Institute of Technology
Electrum 204
S-164 40 Kista, Sweden