Investigation of Audio Transmission over the MBone for the purposes of Remote Lecture Attendance

Pagonis John

May 18, 1998

Abstract

During the course of this project, I was set to investigate the numerous possibilities of transmission of audio (speech) for the purposes of a wider project which deals (1998) with the conception of a remote lecture attendance system. The team of four, which was supervised by Prof. D. Pearson, was assembled by John Woods which had the tedious task of building the (hybrid)MBC-H.263 module and subjecting it to packet loss, by Dr. S Ramanan who works as a research staff for the multimedia lab. and kept us going by filling the part of our project manager, by D. Papanikolaou who worked on the OHT and still images transmission, and me. The actual system and its applications address such a vast amount of disciplines and problems, not to mention 'solutions' and implementations, as well as satisfactions, that can surely not be possible to be tamed or thoroughly scrutinized analytically. Nevertheless the team did its best to face all the challenges and to supply, most importantly, viable answers, solutions and ideas, within the limited time frame allocated. Most importantly what it is believed, is that this wide project, has an obvious strong commercial and academic future. I personally believe that if deployed at the correct scale and backed up by adequate resources this project can be a fully functional product in less than two years.

Contents

List of Figures

Chapter 1

The Wider Picture....

1.1 What is a Remote Lecture Attendance System ?

For many years now, almost for many generations, people have been exposed to the idea of videoconference and telepresence. Most of these ideas have been conveyed through science fiction writing and movies. Even though, less times, such systems have been tried in research facilities as part of piloting programs, or as prototypes[19]to a larger, though specialised, audience. Now it has come the time, thanks to the rapid development of the Internet and relevant engineering areas, that such systems can be feasible and most importantly necessary.

No classic[1]definitions or even descriptions of such system exist and people seem to have, as expected, numerous views or ideas of what such a system really is or should be. The working definition for the purposes of this project, describes an RLA system as

> the dissemination of information using various media, for the realisation of a decentralised and/or distributed classroom or audience, that can hopefully facilitate learning and information[2] .

This definition of course is so general and true, that can accommodate methods like the Open University's TV broadcasts, Web based teaching, video tape delivery, workgroup collaboration systems, circuit switched or ATM video-on-demand and Internet/intranet delivery , and probably more. Which have all been tried and tested and considered for future use, excluding maybe the latter (Internet/intranet), which is probably the most promising and dynamic.

The focus of this project[19] is on an **Internet based remote lecture/conference system**, which is supported by the multicast capabilities of the Internet and indeed of these of its *Multicast Backbone*, the *MBone*[5, 16].

The beauty and momentum of this project and consequently of a remote lecture attendance system, is its diversity and flexibility, in terms of technology and applications. Its potential drives beyond its cause, because it is also a remote conference attendance system, possibly a telepresence system in a shared virtual environment, news or concert multicasting service or even it can be transformed to an MBone video on demand system.

[1]Meaning, globally accepted and honoured (ie. classical music or law of physics)
[2]the communication or reception of knowledge or intelligence - Webster dictionary

The techniques and technologies themselves, do not change, or maybe a litle. The system itself is the same, only some of its components get redirected or enhanced. What changes is the *content and the human audience* which comprise the 'whys', the 'whos' and the 'wheres', because we have almost come to know most of the 'hows'.

1.1.1 Why is it Needed and by Whom ?

Given sufficient quality[3] at each receiving terminal, the scenario of busy, or otherwise, people completing university courses or attending conferences, through the use of their midrange personal computers escapes the sci-fi model and becomes real. A multimedia window to a remote - decoupled both in time and space- class, conference, virtual press room or indeed any such information source yields benefits like :

- remote training and collaboration
- cost effective, rapid and even live information feed
- affordable attendance of events which may be physically, economically or timewise remote
- energy and environmental friendlyness
- scalability and rapid deployment
- user friendlyness
- low cost

With the current acceptance of computers as home/consumer appliances and the popularity of the Internet, an ever increasing amount of the technical and not technical population[4] can enjoy these potential benefits.

1.2 What are its Components ?

The work done so far, targets on a multiple-medium(aka. multimedia[5]) system, whose existence though, as a working system - that is a group of interacting and co-operating components - is in its prenatal period. Some of its individual components, though, exist in one form or another, either at conception or at the implementation stage. Which will eventually grow up to form the system.

Since information has to be presented via many media, this is a good point to present the most immediate 'high level components' that will facilitate this. These can be described as the Model Based Coding *video* components, the *audio* components, the *overhead transparencies* and imaging components and of course the *networking/communication* components. These unsurprisingly are, by far, not the only components needed and indeed investigated, but can at this point show the media used, namely audio video, text and images.

[3]Which is itself a complex issue and reserch area
[4]At least in countries with an average standard of living.
[5]Multimedia is indeed syntactically incorect, the corect word should be multimedium !!

1.3 Where does Audio Transmission Investigation fit in ?

Although sound is the most important[15] medium, it is at the same time the one whose delivery *should not be guaranteed the most.* Speech data[6], are so important to one's understanding of a lecture[7], that should be there, but are also so *transient* that it is even more important to be there *on-time.*

This fact, makes the decision for the appropriate tradeoffs between quality, latency, jitter, bandwidth and processing, a highly non trivial task. These are in fact problems that most real-time service systems have to consider.

Throughout this report, the facts and data of the investigation, results, findings, suggestions, implementations and ideas on , what can loosely be described as, the 'audio transmission part' of the system (see 1.1), will be presented, explained and hopefully elaborated.

1.4 Which Disciplines and Practices are involved ?

The system, is so much human oriented, meaning that its sole purpose and aim is its human audience, therefore it must be directed towards what the audience needs, at the same time though, because such system is not available, the audience does not know what it likes (and probably needs). The audience's reaction to what such a RLA system can offer is not known, or just partially[2] known within limiting constraints. Nevertheless at the same time in order to correctly target and design the system, the audience's 'likes' and 'dislikes' have to be present, to such an extend as for the system to be usefull and therefore successfull. Hence the 'chicken-and-egg' situation that haunts it.

That is why other disciplines like psychology, psychoacoustics and practices like human-computer interaction study and subjective assessment tests have to step in, along with other more evident engineering disciplines, like software engineering and coding. Although rough and crude directions, from these 'human oriented' disciplines and practices were followed, by no means such study has taken place or claimed to have been investigated here. This project should never be regarded as complete if such studies are excluded because clearly, if the human factor is left out, the result will be failure. And again the problem here is that in order to complete such research, at least a prototype system has to be available.

1.5 So is it Possible then ?

After all the discussion about the complexity and diversity of the project, one might be discouraged and question its feasibility. In opposite though, it was found and believed that it is *possible* and more importantly *necessary.* Although this is to be judged by subjective quality assessment tests, some of the individual components tested (following the black box practice), showed that they can very efficiently cope with 30% loss and sometimes more !

[6]That is digitised speech data.

[7]From now on, 'lectures' will be the main targets and in order to avoid redundancies 'lecture' should be read as 'lecture, conference movies..etc'

Chapter 2

Literature Survey

This chapter deals with the case for Forward Error Correction for the audio, the nature of Internet dynamics and packet loss. It supports the decision for FEC, presents interesting traffic statistics and addresses the problems of using the Internet for something that it wasn't designed for.

2.1 Internet

The evolution[22] of ARPANET, in the late '60s, from a military project capable of surviving a nuclear war, to an '80s 'inter-network' and the Internet of the '90s, established not only the use of TCP/IP[23] and now the Web, but also brought closer millions of people and computers. It is supported by various operating systems and numerous computer and communication architectures. Inevitably this exponential[22] growth could not withstand any control or follow any strict hierarchical structure. The aim of its protocols was to allow for openness and inter-operability and for the network itself to resemble an ecosystem where nothing exists in isolation, though everything evolves at its own pace and with its own characteristics.

This lack of structure is what makes the Internet so powerful. Its beauty lies behind the fact that everyone - every host and sub-net - is created 'equal' and has to cooperate and 'serve' each other.

2.1.1 The nature of the beast

This complexity and 'democracy' is upheld by the 'best-effort' delivery approach, that is offered by the TCP/IP protocol stack. Where no guarantees are given about the loss rate. delivery time, bandwidth allocation, or even for the packet routing.

Therefore if any RLA system, which highly depends upon packet loss, delay and jitter, has to use the Internet, either it has to reverse the Internet's nature and super-impose some structure like done on the telecoms networks which are highly hierarchical and structured, or work its way through the internet's anarchy and exploit to the maximum those best-effort features.

With the current implementation of TCP/IP, two *transport layer*[1] *protocols* are offered on top of the underlying (network layer) Internet Protocol, namely TCP and UDP. TCP offers *reliable full-*

[1] Following the ISO 7-layer OSI naming convention

duplex connection-oriented byte stream[2] communication with *error control* between two Internet hosts, whereas UDP allows for *unreliable, connectionless* communication between hosts that wish *no flow control or sequencing.* A 'newcomer' to this family of protocols, is RTP[8, 20],which is currently implemented at the application layer and uses UDP as its bearer transport. RTP is said to be included in the future, at the transport layer within IPv6, and together with RSVP promise to bring QoS and real-time applications efficiency to the internet. Clearly UDP's *fire and forget* mode is of interest for the audio transmission of this project, if today's technology is to be used.

The end-to-end packet switched connection between two hosts, on the internet can, from a high level view, be seen as a single communication channel (see also chapter 3), whose loss. delay and jitter characteristics though, are not known(before transmission) and even worse, whose characteristics change with time and location following no obvious and simple patterns.

2.1.2 Internet Packet Dynamics, so far...in brief

Although this subject is of great interest, not to mention complexity, not many studies[18] exist that try to give an analytic model for Internet end-to-end transmission based on large scale measurements. The major issues regarding such analysis, involve measurements and investigation of:

- Packet Loss
- Packet Delay
- Delay Jitter
- Packet Reordering
- Packet Loss Bursts and Loss Correlation
- Packet Duplication and Corruption

All these investigations , should span on different topologies, time scales and geographic locations. From the literature it seems that two major approaches have been used, one dealing with large number of transfers of bulk data using TCP[18] and other, more common, measurements using fixed length UDP [11, 1, 7, 24, 10] and ICMP "echo"[6] packets.

The ultimate reason for performing these experiments is to derive a *working model, with which, if possible*, to *describe an 'Internet channel' from source to destination, as accurately as possible*[3]. Such models are then used for checking against real transmissions, and consequently used in analysis within simulation[3] environments, in order to study realistic scenarios, and/or for 'training' and tuning existing or developing systems. *The latter is the reason for following*(see chapter 3) *and proposing*(see section5.1) *such an approach with this project.*

The problem, with these experiments, is the, sometimes, oversimplification of an inherently complex problem, which is nevertheless necessary, in order to produce an analytic view of manageable complexity. Thus researchers are against yet another 'chicken and egg' situation, since

[2]Streams have their packets ordered automaticaly from the transport protocol.

[3]Which is by construction impossible

the point of the experiments is to understand the complexity and characteristics of Internet 'pathways'. This is the reason why, when using such results or advocating these methods, one should be very weary and cautious, since some times the bits that are left out are the most important.

The results so far have disclosed two distinct behaviors, which surprisingly are mutually exclusive and expectedly bound to the design of the experiments and transport protocols used.

The first behavior based on UDP transmissions describes

- *most packet losses as isolated,*
- *with most packet loss periods having short lengths*
- and the *probability of a loss event involving N packets decreases geometrically fast with N, for small enough values of N,* as Bolot[11] so elegantly states.
- *the packet arrivals, and hence the delays, follow a Poisson process*
- *out-of-order delivery is prevalent* - although MBone routes are more static
- *geography does matter allot*

Behavior based on TCP transmission describes that

- *loss events are highly correlated and cannot be modeled as independent*
- *traffic and loss are both bursty*
- *data packets adapt*[4] *to the network's conditions*
- *ACK packets do not adapt to the network's conditions*
- *queued data packets have more changes of being dropped, ACK packets less chances and transient*[5] *packets even less*[18].
- *packet timing compression is possible, though rare for both ACK and data packets*
- *out-of-order delivery is prevalent*
- *geography does matter allot*

These findings about the two basic kind of traffics, allow not only for a better understanding of the packet dynamics, but also for deciding on a transmission protocol (UDP for our purposes) and hence error recovery policy. *The patterns observed by the studies, named previously, on UDP traffic, were also observed during the course of the project*(see section 2.3 and chapter 3).

[4] Due to the sliding window strategy
[5] Packets that arrive at a node and undergo little or no buffering

Figure 2.1: UDP Packets Delay Distribution Example

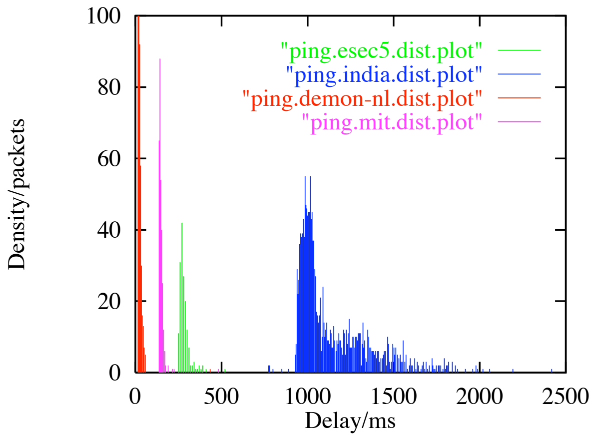
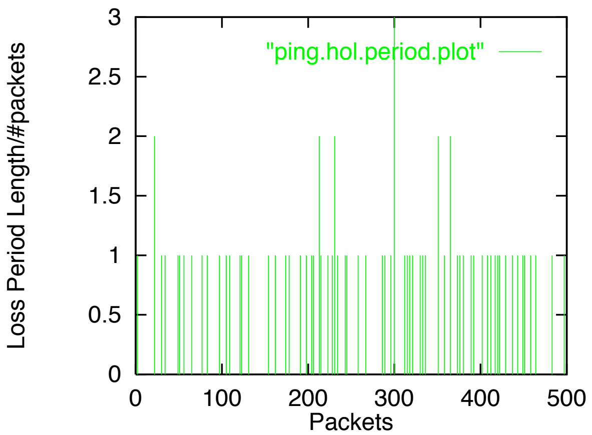


Figure 2.2: Loss Period Representation Example



2.2 The Case for FEC

Taking the knowledge on the UDP-based transmission characteristics onboard, one can see that really the only way to go with packet audio transmission on the Internet is by using Forward Error Correction as the best error recovery strategy.

The patterns show that loss events can be modeled as independent, with short[11] loss periods most of the times, like the example in figure-2.2, thus the sender can protect the data by adding redundancy to the information sent. This can be achieved by the process of *piggybacking*, where an audio sample (in this case) is sent across the network, together with redundant information about previous (or even future) sample(s), in the same packet. Usually the redundant information undergoes higher compression. Other ways of FEC like XOR and Reed-Solomon codes parity checking may be used, but are more effective on image transmissions, where the real time constraints may be more relaxed. At the receiver (see also section 5.2), the samples are unpackaged and stored according to their priority, based on their coding scheme. The best quality stream is assembled and if samples are missing the receiver falls back to error recovery mode, where it compensates for the loss using the lower quality samples, or in the worst case by adding noise or repeating previous samples[11].

It is very difficult to decide on an effective FEC policy, and indeed to choose one of many. It is also very much dependent on the channel. therefore *no single policy is sufficient for the entire of the Internet channels.* There are so many codecs of which to choose and there are also so many different combinations possible, that any decisions should be really driven by well conducted subjective quality assessment tests, channel loss characteristics investigations and bandwidth budgets. It could be possible to decide for a global 'worst case scenario', but this tends to be really inefficient (see chapters 4 and 5) and bulky.

It is only fair to suggest that *each channel should be carefully studied and each transmission should be carefully targeted and tailored to be 'custom made'*(see section 5.1). *By 'channel' it is really implied "an end-to-end path, at a particular time and day", the same path at different times should be regarded as a different 'channel'.*

Therefore it was decided that FEC is the way to go, but no single FEC policy is sufficient. Forward Error Correction is in practice redundancy, either in the form of piggybacking or layering (see chapter 4), but the tradeoff between bandwidth and quality is necessary.

2.2.1 What are the alternatives ?

In order to support further this decision, the alternatives to the FEC strategy should be presented.

The first immediate alternative would be to use TCP. This would eliminate the need for redundancy (from the FEC data) since the automatic retransmission (due to the ARQ) will compensate for loss. The downside to that is the added delay and increase in jitter due to the retransmissions and acknowledgement mechanism. Also even if the network could provide minimal loss and low latency like a LAN, where even interactive audio communication using TCP is possible[14], TCP cannot support multicasting, hence multiple connections would have to be set-up each time, therefore leading to more expensive bandwidth requirements.

Another new approach that is currently being used with success by UCL[10, 24] and INRIA[17], is to make the audio tools (namely RAT and Freephone) adapt to the network's conditions. This is done by utilising a feedback path from the receiver, which periodically sends back QoS reports. Hence the transmitter adjusts the send rate and the redundant information. Although this approach is most effective for Internet telephony and will eventually bring high quality voice communication over the Internet, it does not scale for MBone transmissions, because of the complexity of handling all the return paths and adapting to all the receivers' requirements.

The Real Time streaming Protocol, jointly developed by RealNetworks, Netscape Communications and Columbia University, is built upon RTP and is used as a control protocol for initiating and directing delivery of streaming multimedia from media servers, the "Internet VCR remote control protocol". RTSP does not deliver data, though the RTSP connection may be used to tunnel RTP traffic for ease of use with firewalls and other network devices. RTSP uses RTP for the actual delivery of the streaming data and it relies heavily on the buffering of data at intermediate servers. Although it facilitates unicast and multicast delivery, it has yet to be used extensively on the MBone.

2.2.2 Experts' results

Some interesting results have been obtained from various studies and these could be used as pointers and 'rules of thumb' towards what is needed 'quality-wise', here some characteristic ones are presented.

Vicky Hardman at UCL(during the MICE project) found, when working for their audio tool RAT, after conducting experiments on subjective quality, that people gave positive response to a [PCM,LPC(1)] FEC scheme used for a 'connection' to the US. Where people would rate the system under 15% loss with 4/5(on a 1-5 MOS scale) and 3.6/5 and 2.5/5 when subjected to 20% an 30% loss respectively. Therefore using that model, with single copy of redundancy is good enough to provide packet loss protection in the region of 20-30%[24][9]

The audio tool that Bolot's team built at INRIA uses a clever way of compensating from a burst of 8 consecutive packet losses. A brief description of this scheme is given here, as found in one of Freephones web pages (at www.INRIA.fr).

Lost packets are reconstructed in three different ways:

- Using redundant information to reconstruct (at least part of the) missing information
- Using the previously received packet (i.e. using packet duplication)
- Using silence

An example of packet reconstruction is shown in the figure 2.3. The figure shows a sequence of 8 lost packets starting at packet i+1 and finishing at packet i+8. Packet i is duplicated as packet i+1. Packets i+2 and i+3 are replaced with silence. Packet i+4 is a duplication of packet i+5. Finally, packets i+5 trough i+8 are reconstructed using the redundant information contained in packets i+9 and i+10

FreePhone attempts to reconstruct lost packets first using redundancy, then using duplication if no redundant information is available, then using silence if the maximum number of duplications has been reached. Note that using these reconstruction techniques does increase the end-to-end delay.

Figure 2.3: Packet reconstruction in FreePhone

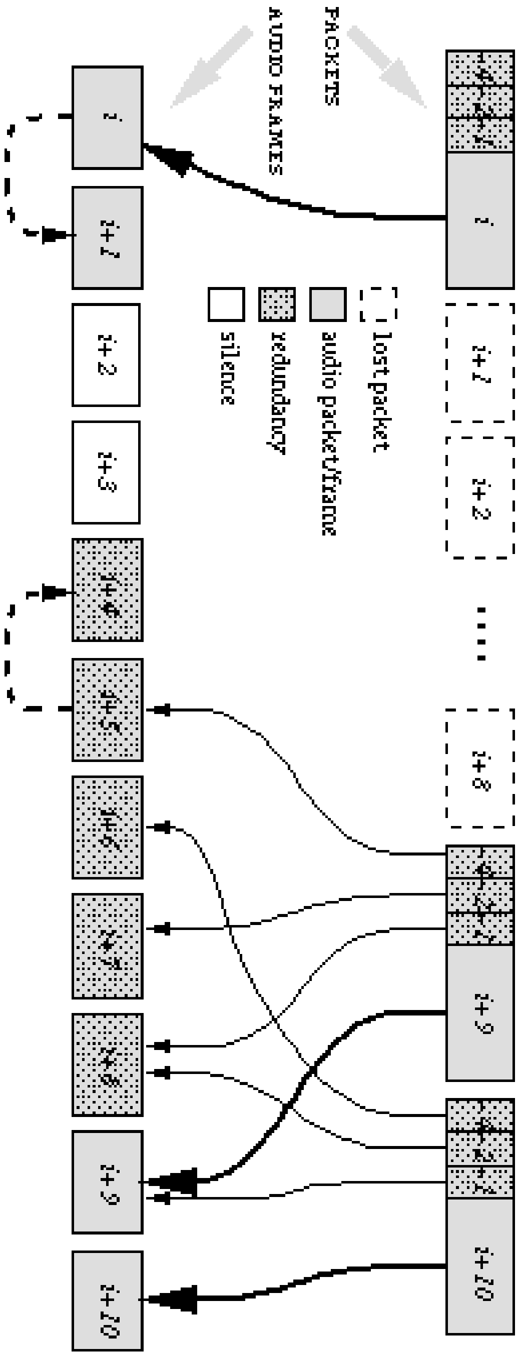
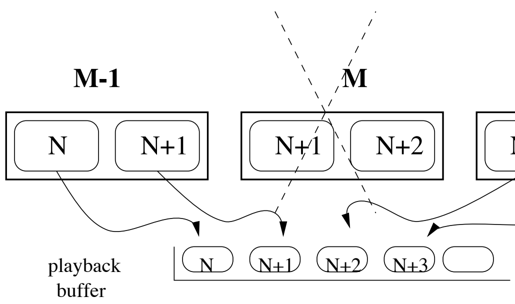


Figure 2.4: Piggybacking coping with 1 loss



2.3 Some Experimental Work....

2.3.1 A Loss Recovery Experiment Using FEC

Although Bolot showed a superb way of protecting delivery of voice samples from a burst of 8 lost packets, he didn't give any MOS results for this technique or described a "user's" response to this compensation. It is believed that his results are indeed good but also it was believed that a more personal opinion should be obtained, since this is a subjective matter. Also his scheme was redundant enough, so that it cannot be experienced from a 'thin' connection to the Internet.

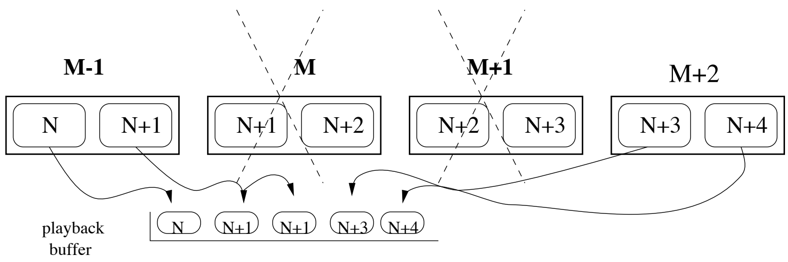
Thus another experiment (top level code can be found in `'SoundCheck.cc'`) was conducted that would give a hands-on experience with this mater. It was assumed that the connection to the Internet would have been as 'thin' as possible. Note that at this point only audio communication is considered and this limitation is not unlike the one that would had been imposed if a 128Kbits/sec connection to the MBone was used. Therefore the GSM 6.10 codec was selected, whose source code was freely available by the University of Berlin and can also be compiled in many platforms. This implementation of the codec encodes an array of 160 13bit samples into a frame of 33bytes and vice versa. In order to work it needs 16bit(signed) linear PCM samples as input(coder), of which it uses the 13 most significant bits. This produces 20ms worth of audio fragments. Assuming now, that 2 such fragments can be packaged together in a packet and sent over the network, the result is a [GSM,GSM(1)] piggybacking.

This means that for each Nth 33byte frame sent, the (N+1)th is added to that packet. Therefore if the receiver does not get packet M(containing frame N+1) it can recover frame N+1 from packet M-1(figure 2.4), if sequence integrity[6] is assumed(in order to avoid sorting the packets since this was just a simulation). If now 2 consecutive packets are lost(M+1 does not arrive) then frame N+1 is recovered as before and in place of frame N+2, N+1 is repeated instead(figure 2.5). If more than 2 consecutive losses are experienced, then the 'gaps' get substituted with 'silence'[7]. This same experiment was also conducted using frame duplication even after the third loss, but it came to be very annoying for loss rates(loss period>4).

The minimal subjective test that roughly evaluated this scheme's quality showed that people did not mind even when having 20% loss and also responded positively when the loss rate was 30%. The conclusion was made that people would have no problem using a 2-way communication system

[6] Which is not the case in a real transmission
[7] Although some form of noise would had been better

Figure 2.5: Coping with 2 lost packets



with this scheme, under 30% loss, if it was for long distance (expensive otherwise) communication and that they would certainly listen to a football match(if it was for their favorite team) with 50%loss if that was the only medium available !

Though these results are not representative, since the subjects were only 7 (5 men and 2 women) and the running time was only 1 minute (for each different loss ratio), nevertheless they gave some pointers towards what to be expected. It is believed though that if the exposure had been longer (around a half hour or so), then the response would had been less enthusiastic. And still it is very difficult to say what kind of loss would be tolerable[2] for lecture attendance.

While this experiment might give optimistic results, it still remains a rough simulation. This is because although theoretically the bandwidth needed by such transmission is 2x13.2Kbits/sec, in reality due to the IP and UDP encapsulation[8] (and therefore header data), this transmission would actually need 37.6Kbits/sec (though in reality it could be eased by the use of silence compression). It is therefore very inefficient to send UDP packets worth of 94bytes(70% worth of data of which half are redundant). Packet length at the region of 160-320bytes would be more appropriate (and still information like sequence number and time stamp are not considered), also the inter-packet frame multiplexing would have to be more clever.

2.4 Finally...

The literature survey shows that TCP and UDP traffic not only is used in different families of applications but are also foundamentally different in behavior and characteristics so that need to be investigated and modeled using different approaches. Hence, as showed, UDP traffic is of interest for this project and should be studied under various circumstances in order to yield the best transmission policies for a viable multimedia system.

In this chapter not only the theories were stated, but they were also supported by both the researchers' results and our own findings. This mixture of theory and experiments was found to be appropriate in order to convey a more complete picture of the happenning in this area. Forward Error Correction was shown to be adequate and indeed fits well with the characteristics of the UDP traffic, where under 'normal' conditions losses are uncorrelated and loss periods are short. Also it was found that the experiment conducted here (section 2.3.1) somehow agrees with the experiment conducted in UCL[24, 9], although the coding used wes different, and though the

[8] Ethernet or IEEE 802.3 frame encapsulation is not considered here since such information stays on the LAN and does not travel to the Internet.

technique was the same, giving this way an idea towards what might be the correct approach. These examples above in section 2.2.2 together with the experiment in section 2.3.1 agree with the suggestion in section 2.2 that " FEC is the way to go, but no single FEC policy is sufficient".

What remains unresolved is to find a way of modeling the transmission channels on the Internet/MBone in order to obtain the 'a priori' knowledge for offering the best loss recovery scheme. Later it is going to be shown how to offer different grades of service on the MBone and what can be done in the future to add some intelligence to the system and how it could possibly be implemented.

Chapter 3

Probing the Net....

It was said before that the reason for performing these experiments is to derive a working model, with which, if possible, to *describe an 'Internet channel' from source to destination, as accurately as possible.* Hence the purpose of the *'ping analysis'* described below, is to discover the packet characteristics and decide on the best error recovery and FEC policy. Though as discussed previously no single FEC policy or model can work on the whole of the Internet and each path has to be targeted appropriately. Nevertheless this experiment and analysis described here can be used for that exact reason, to probe the network where necessary and thus to define as many models as needed.

3.1 In search of a Model..

3.1.1 Why...

Following a recommendation from a Fujitsu SciTech paper[3] which points to a study[6] from the National Laboratory for Network Research, an end-to-end Internet performance assessment, suggested that a 'hacked' version of the 'ping' (Unix network utility) could be used to collect information about loss traffic and delay between two hosts.

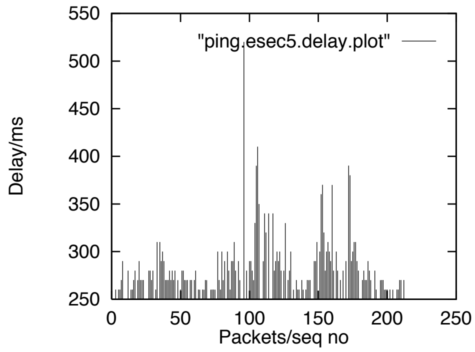
Since this was in accordance with what was discussed in section 2.1.2 it was decided that this utility could help in building 'models' for any channels of interest, in order to later use these models for testing the systems components and customise the FEC policies. The *ping analysis* results could be fed into a synthetic loss model and test how each audio transmission would have coped with that channel and, hence, plan a policy for dissemination of data on that channel.

3.1.2 How...

The *ping analysis experiments* consist of three stages, the first stage is the *probing* of the Internet using the *ping* utility, the second is the *extraction* of the usefull information from the gathered statistics and the third stage is the *analysis* itself, for which the **invaluable help of Dr. Sean Monaghan was used.**

A problem with the experiments is that they cannot span greatly in time, because the granularity becomes coarse and the characteristics of the probed channel tend to average and smoothen in time.

Figure 3.1: Pinging esec5 from Demon – Delay graph 4:00 pm weekday



This is because the behaviour of the Net changes from hour to hour, even from minute to minute, making the task of observation tediou. During the project the 'sample granularity' was chosen to be in the order of hour and half-hour, since these time intervals are representative of lecture transmissions.

3.1.3 Pinging...

Since for the loss simulation experiments (see section 2.3) the GSM 06.10 codec was used(from the university of Berlin) and realistically UDP packets cannot be smaller than 160bytes, the echo ping packets used were of size between 160 and 500bytes. This numbers are very important since they are 'typical'[11, 6, 14, 24, 4, 9, 12] for such interactive and/or real-time transmissions. The interval between each successive ping packet sent, was between 1sec and 1min depending on the time of the day, in order not to load the network. *Although actually no difference was observed in the properties of the channels tested, between the 1sec interval pings and the 1 minute interval pings.* This is somehow interesting and it is believed that more investigation is needed anyway.

By no means any systematic pinging experiment was conducted, this indeed would need to be exercised throughout the year at different times of the day and for different locations with different loads. Which is non trivial and requires formal planning and management, since the amount of the gathered statistics would have been large. It could clearly be a project of its own, though here some pointers are given towards what can be done and what really exists.

Nevertheless, interesting observations were made like the one shown (figure 3.1), were a host (namely esec5) from this departments ESL lab was 'pinged' from a host connected to Demon's[1] UK network on a Monday afternoon around 4:00pm. The data traveled through eight hops, they went off the UK then went through the Netherlands and then entered JANET and finally the ESE department's. network, traveling half the Europe and back !

The pinging intervals used for this (figure 3.1) were 30secs and lasted for about one and a half hours. The *roundtrip* delays shown were mostly distributed between values of 260ms and 280ms as the graph shows (figure 3.2) and the overall loss was 22%.

The loss period lengths were small, in accordance with Bolot's findings[11] and characteristics describe in section 2.1.2. Also notably the delay distribution (figure 3.2) uncovers a Poisson

[1]Demon Internet plc. is a UK Internet service provider with 180.000 residential customers and has a fairly extensive network.

Figure 3.2: Pinging esec5 from Demon – Delay distribution graph

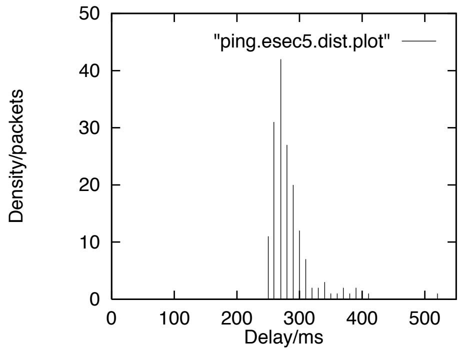
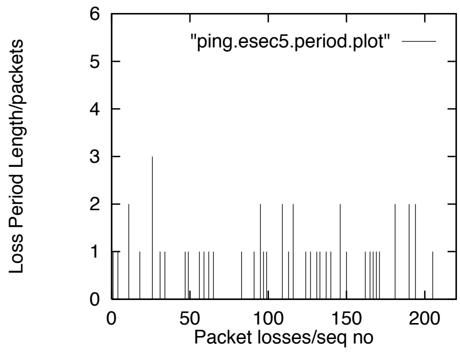


Figure 3.3: Pinging esec5 from Demon – Loss periods graph



distribution nature.

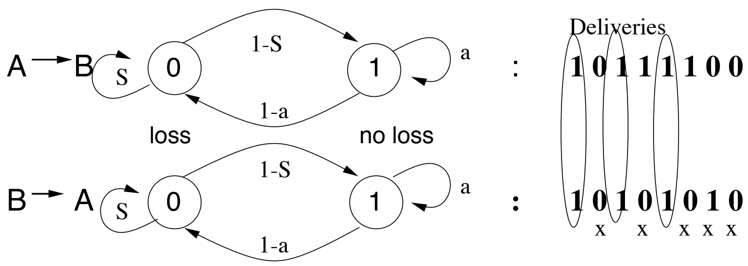
Interestingly the same experiment was conducted few more times and for a Sunday afternoon it gave a loss of 12% and similar variations in delay, while on a Tuesday night (11:30 pm) there was no loss at all for ping packets of 64bytes, though, but the delay variation was between 170ms and 56000ms with most of the delays distributed around the 3000ms mark !!

"What can we say about these graphs and results ?", is the obvious question, but there is no obvious answer. This is because these are results based on *packets that traveled roundtrip* from host A to host B and back. Therefore, this means that are only good for crudely describing path A->B->A and nothing else. For example, the esec5 statistics say that because the roundtrip delay was between 260-300ms this channel can be used for interactive real time voice communication. And that since the loss periods for the roundtrip were small then statistically the one way loss periods will be small. Therefore we can protect our data with a fairly light FEC scheme and we have to do so because the roundtrip loss was 22%. But how much is the one way loss and how can this path be described?

3.2 Analysis

In order to find a model that could describe channel A->B, first a model that describes A->B->A has to be given since all the collected data, source from that path. Then this roundtrip path should be broken to downstream and upstream paths, of which the downstream is of interest here. Then

Figure 3.4: 'Roundtrip' channel representation using Marchov



hopefully by utilising the measurements and knowledge from the experiments and the analysis, a viable model can be created.

3.2.1 The general model

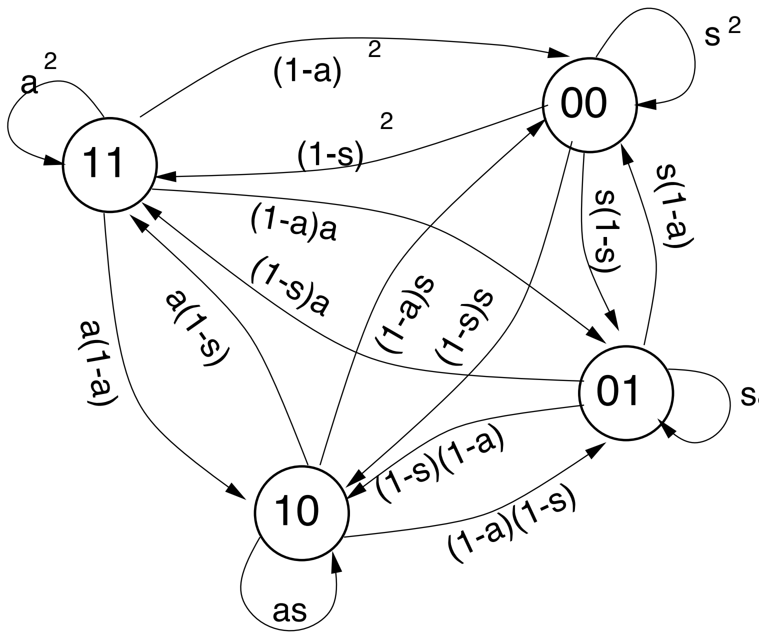
The general form of the problem can potentially be described by two two-state Marchov models (see figure 3.4), which yield a four-state system. By no means the system can be simplified to the trivial two-state Marchov chain as some may suggest. This happens because the system describing the roundtrip of the ICMP echo packets, falls in at least four states.

As shown in figure 3.4 the whole path can be described by two separate Marchov chains which use two general parameters for probabilities s and a. As shown a packet will arrive back only if the system falls in the 11 state, which is of probability a^2.This description is based on the assumption that the forward and return paths are the same. The use of s and a here, denote a general solution to the problem, which may accommodate correlated or independent events.

Having the general model, which can accommodate correlated losses, is due to the fact that if correct analysis is to be made, then the fact that the losses are independent has to be proved first. Also as Bolot states[18, 11], most of the times the events(losses) are independent and this is for the cases that the network is not congested, therefore if buffers start to overflow any analysis based on independent loss models will be highly inaccurate. Of course it can be argued that if there is the case that major buffer overflows occur at the routers, then the system will collapse anyway since FEC cannot cope with long bursts of errors, nevertheless such cases have to be distinguished and isolated using correct analysis. This can be facilitated using the utility and tests described later.

From the four states that describe the A->B->A path, only state 11 can be extracted from the ping statistics, numerically, therefore a solution to the problem of finding s can be given as a function of P_3, a and s, where P_3 is the probability of being in state 11 (all the other probabilities P_0,$P_1$and P_2 corespond to states 00, 01 and 10 respectively). Also a^2 can be measured since it is the number of consecutive arrivals and P_1 is taken to be equal to P_2. Therefore by solving the system described by this matrix operation (note that the 'dashed' probabilities are the unnormalised ones) all the probabilities may be found and may be used to model the downstream, in terms of a two state Marchov model.

Figure 3.5: Ping roundtrip 4-state diagram



$$\begin{pmatrix} P_0 & P_1 & P_2 & P_3 \end{pmatrix} =$$

$$\begin{pmatrix} P_0' & P_1' & P_2' & P_3' \end{pmatrix} * \begin{pmatrix} s^2 & s(1-s) & (1-s)s & (1-s)^2 \\ s(1-a) & sa & (1-s)(1-a) & (1-s)a \\ (1-a)s & (1-a)(1-s) & as & a(1-s) \\ (1-a)^2 & (1-a)a & a(1-a) & a^2 \end{pmatrix}$$

where by setting $P_0 = 1$

$$1 = s^2 + 2s(1-a)P_1' + (1-a)2P_3'$$

$$P_3' = (1-s)^2 + 2a(1-s)P_1' + a^2 P_3'$$

$$\text{and } P_0 = \frac{1}{1+2P_1'+P_3'}, \quad P_1 = P_1' P_0, \quad P_2 = P_1, \quad \mathrm{P}_3 = P_3' P_0$$

If it is found that $s \approx 1-a$ then it can be said that the observed channel showed independence between the losses(dropped packets) hence a more simplified version of the general solution can be used.

3.2.2 The case of independent events

Since it can be regarded as unnecessary or as an overhead to test each time each 'probing' to find if the channel showed correlated losses or not, then this test can be circumvented and assume from the beginning that $s = 1 - a$. Indeed most of the research (chapter 2) shows that UDP traffic packet losses can most of the times be modeled as independent.

In that case since P_3 can be measured from the collected ping statistics and $P_3 = a^2$ then the 'one way' estimated loss is taken to be simply $P_{downstream}[Loss] = 1 - \sqrt{P_3}$. These findings come of no surprise of course and are really 'back of the envelope' equations. Nevertheless what they offer is a 2-state Marchov model which can most of the times represent a channel's behaviour and give trustworthy results.

3.3 Analysis Software

For the extraction of the usefull information from the ping statistics, some software had to be built. This software (utility) was used in order to produce the data for all the plots and statistics shown in this report. It can be used to filter all kinds of information from the raw 'ping files' and since it is in the form of a miniature 'C++' library(`'analysis.h')` it can be used by any application or high level tool that wishes to do so, like the example utility `'analysis.cc'`

Both, the case of independent and the correlated losses, can represent each channel (downstream or upstream) as a 2-state Marchov. If these models are put in software then *observation of* a *channel's behaviour*[2] *can take place 'off-line' in a lab, under a simulation environment using though 'real' figures* from an already conducted experiment.

Such C++ code can be found in `'teststate.cc'` and `'simulation.h'`. What was done was to create two 'state objects' , one object represented the loss state and the other represented the no-loss state. All the objects know were and how to calculate one probability and have 'knowledge' of their next possible state. Therefore one state can be linked to another. In a run the current state object calculates the probability of staying as it is or possibly follow the next state.

This way after a long run, one can observe a simulated channel's behavior and also this can be made to produce such an output as to be fed in the analysis utility and produce graphical information such as loss period graphs. This can be proved to be very useful since from the collected ping statistics one we cannot have the real losses or loss periods but can nevertheless estimate them from a simulation run.

[2]This can be done also for delay and jitter, if the software model is extended in order to accomodate these extra parameters

Chapter 4

Adding Grade of Service

Initially it was assumed, that the RLA system should use as few resources as possible and that the audio data, although most importand, should occupy as less bandwidth as possible. Later these constraints were relaxed and the idea of having a system which could provide different grades of service, to users with different bandwidth budgets, was born.

4.1 Layered Audio Multicast

4.1.1 Package Based Audio Multicast

It was initially conceived that different grades of service could be brought to the system by using different multicast sessions(i.e multicast addresses) for different pre-configured QoS audio 'packages'. Such a 'package' would use a distinct FEC policy targeted towards the QoS, bandwidth requirements and properties of the transmission path towards each user or group of users, that would 'subscribe' to it.

Effectively it would resemble the tuning of a radio receiver to a particular frequency. Each receiver should know, somehow, the FEC policy of its receiving audio 'package', in order to decode it and play it back. By 'package' it is meant the FEC policy used for encoding and protecting the audio. That is effectively, the combination and order of the encoded samples, within a packet or group of packets.

Figure 4.1: Package Based GoS Multicast

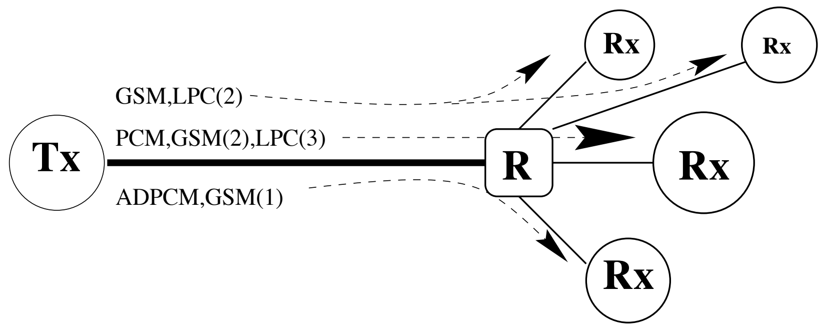
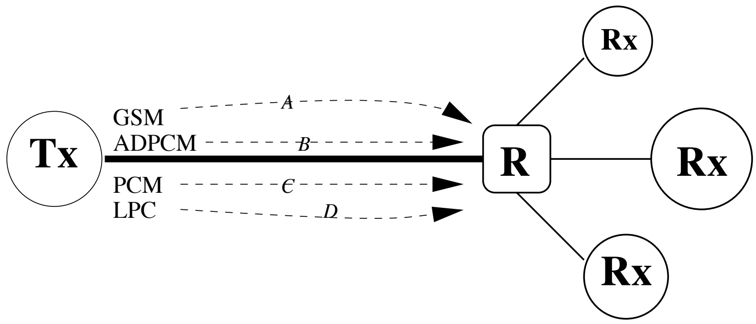


Figure 4.2: Layered Audio Multicast



Although such a solution, to bring different GoS to the transmission, may look simple and attractive at the beginning, it can rapidly become expensive and difficult to administrate. This is primarily due to its monolithic design, where everything is hardwired from the beginning. If we assume that only a small number of packages will be available and these would be transmitted on 'well known' paths, then it might be a good solution. If, on the other hand, the number, requirements and location of the receivers, are not known(see section 5.1) before hand, then there is the problem of literally guessing which packages to transmit, let alone the possibility of the changing requirements of each receiver during a multicast session.

Due to the fact that most of the times the targeted audience will be heterogeneous, many 'packages' will have to be offered. This clearly is very expensive since each package will use a different multicast address (of which there is a finite number) and a multitude of coded samples per packet, adding extra and indeed unnecessary redundancy. Additionally there is the problem of a receiver not being able to cope with a particular package because of its CPU usage/power, so that it would like to use different codecs which may not be possible due to the lack of a package that would satisfy such a need. On top of that is also the problem of a receiver changing dynamically packages after monitoring and trying to adapt to the changing loss rates of its connection.

Clearly, although such a scheme requires less intelligence from the receivers point of view and has a simple session management (for the receivers),it is very difficult to adjust to the changing needs of the users and the network's properties and it doesn't scale for large deployment.

4.1.2 Layered Audio Multicast

Another approach to the fixed configuration transmission, that *package based audio multicast* offers, is to transmit the coded streams independently. In this transmission scheme, audio samples are *layered* according to their encoding on different multicast addresses. In effect this means that simply the transmitter will encode audio in different formats and choose *one address per format*, to transmit. This way, for example (see figure-4.2) PCM audio will use address C, GSM will use address A, etc.

This way the receivers will receive and buffer *dynamically and at will* the streams that they find most appropriate based on their view of the network's condition. In effect what happens here is that *each receiving host can choose its own FEC policy from any combination of the available streams (codecs),* something that was clearly impossible before.

This layering strategy is so advantageous because :

a) The Transmitter will only *use few multicast addresses*, which can be as few as the number of codecs used.

b) The transmitter *does not have to 'advertise' the FEC policy* that it will use for the demultiplexing (of audio fragments at the receivers), since the *receivers decide dynamically on the policy to follow.*

c) It is *more economic* since it uses much less bandwidth than the packaged streams.

d) The scheme is more *flexible and configurable* because the transmitter or a router can add or drop streams, safely due to pruning (see section 4.2.3 below), congestion or excess bandwidth. This means that by dropping a stream, for any reason, if this stream is not the primary stream of any receivers, then none will be significantly or at all affected, but congestions can be avoided and the network can be eased. Moreover if at some point there is more available bandwidth or 'richer' hosts join, then more streams leading to better quality of service, can be offered.

e) More hosts with different budgets and processing power can participate, since there is no imposed demultiplexing and coding schemes (apart from any of the offered ones). For example a host with a 'fat' pipe to the MBone, which has though a weak CPU, can choose to 'listen' to the PCM and ADPCM streams which require more bandwidth but lot less processing power than the GSM streams.

4.2 Further Issues...

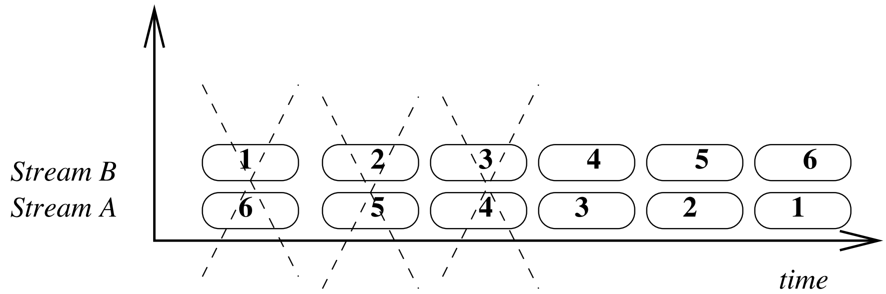
4.2.1 Coping with Loss Bursts...

It was stated that following the strategy described in section 4.1.2 the receiving hosts can dynamically choose the FEC strategy of their choice, though for this to work the transmitted samples have to be protected from bursts and from losses that may occur on more than one streams simultaneously.

To avoid the risk of having all the streams affected by the same burst, care must be taken so that the streams never be synchronised. To take this idea further, the audio fragments contained in each stream(multicast address) should be spread/scrambled in time, so that at any instance τ if any error/packet loss occurs the receiver will not lose the same audio fragment from all its receiving streams. Even more, if the fragments are spread correctly, such scheme can cope with long bursts.

Of course the amount of spreading in time and streams, is a non trivial issue since it can lead to poor or optimum immunity from bursts of error/loss. One of the major parameters affecting this strategy's performance is the buffering and jitter compensation policy of the receivers (see section 5.2).

Figure 4.3: Burst Immunity



4.2.2 A Bandwidth Budget Example...

As an example of the difference in bandwidth budget between the package-based and layer-based schemes, one can take a look at figures 4.1 and4.2. The bandwidth requirement for the first example is

$$B\,(GSM, LPC) + B\,(PCM, GSM, LPC) + B\,(ADPCM, GSM) = 139.8kbits/sec$$

whereas for the second example it is only

$$B\,(GSM) + B\,(LPC) + B\,(ADPCM) + B\,(PCM) = 111.3kbits/sec$$

which is a 21% less.

Still this may not be enough to convince, if now in the equation an extra GSM stream is added, in order to have 2 GSM streams for fall-back, for something like a [PCM,GSM(1),GSM(2),LPC(4)] scheme, then for the first example, the full 92.4Kbits/sec should be added, whereas for the second example only extra 13Kbits/sec, for that second GSM stream should be added. Clearly this is a major advantage.

4.2.3 The MBone in 60"

Although the purpose of this report is not to show the internal workings of the MBone[5, 16, 10], a very quick and high level overview seems inevitable.

The MBone is the *multicast backbone of the Internet*, which allows for a source to disseminate information to a, usually large, number of recipients who have request it[4]. The smart thing about the way packets are forwarded on the MBone is that the *streams are kept as unreplicated as possible down each path* and are only duplicated, if necessary, upon request. Thus the minimum amount of paths are followed having several neighboring routers or hosts sharing the same streams as much as possible. This request and membership procedure is facilitated by the Internet Group Membership Protocol(IGMP).

Another characteristic of the MBone is that the addresses used for forwarding the packets, *do not refer to individual hosts but rather have a global significance and imply multicast services.* This addresses *resemble a radio frequency,* where each host can tune in. The multicast routers use a technique known as *pruning* in order to ensure that packets are delivered only to hosts that have requested[4] them. This is a rather receiver oriented technique, where the receivers have to subscribe to a particular service (multicast address) through their routers and, consequently, keep this subscription alive by periodically renewing it. If not, the routers prune their branches(receivers) and ease the network, this procedure propagates upstream the multicast paths and can even reach the transmitter which may stop sending packets downstream if none wishes to listen.

Chapter 5

Future Perfect...

5.1 Avoiding the worst case scenario protection

Currently because of the 'one-way' traffic mode of the MBone, it is very difficult to gather 'live' statistics of the progress and quality of each transmission/reception. Therefore, the transmitter has to guard (see section 2.2) the data with added redundancy, from a 'reasonable' worst case scenario - since a return path is almost unlikely due to the number of receiving hosts. Which may be indeed a bandwidth overkill for some or for most of the multicasts. It is also very difficult to manage all these reports at the transmitter-end in real time. And even if that was possible the transmitter could not respond and adjust to all the requests.

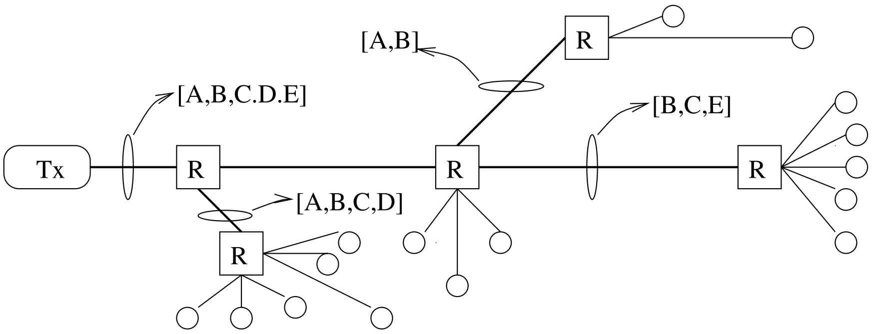
Now *if there existed some prior knowledge about a 'per-hour/location' possible worst case scenario, then possibly this could result to yet, some less bandwidth-hungry multicast.* It is a common secret that a large proportion of the internet's traffic follows the social behaviour and patterns of its users at that geographical region and although that internet's traffic and usage is spread across almost all of the time zones, making it continuous, it is nevertheless accepted that certain heuristics apply. Such heuristics that follow, mostly sociological patterns, generally yield axioms about the usage and traffic during different times of the day and days of the year. For example it is generally accepted that at 6:00pm, when most of the people leave their jobs or universities and are on the road, the Internet usage here at the UK drops and at 8:00am since most of the US users are asleep, it is a good time (at least in theory) to retrieve large files from across the Atlantic.

If now these heuristics and axioms could be translated as to extract 'numbers', then the economics, revenue and quality of an RLA system could greatly benefit, since a more correctly targeted error recovery policy would be used. What is suggested here, sounds like clashing with what was discussed in section 2.1.1, though it does not.

5.1.1 Adding Some Intelligence

Intelligence can be added to the system by having a 'memory' of the past transmissions' behaviours and by interpreting such 'knowledge of the past' as to estimate possible patterns for the future. In order for this to happen, some kind of memory has to be added to the system, then this memory has to be accommodated with usefull data and finally these data have to be interpreted to usefull

Figure 5.1: Targeted FEC using layered multicast



information.

The obvious way to support such a memory for the system, is to couple it with a database system. The database should not be essential for the operation of the RLA system but it should be beneficial. No dependence should therefore exist from behalf of the RLA 'transmission' system towards the database - this is a key factor in obtaining maximum maintenability.

The database will store all the 'Internet probing' statistics from the *pinging,* receiver reports (see section 5.1.3) and channel analysis. The statistics should be probably stored according to calendar dates, times and locations(channel paths). This can be both usefull for periodic multicasts, like a lecture every Monday afternoon or newscasts, but also it will be of use for sparse or rare events like conference multicasts. Such events as the latter can benefit, simply by 'knowing' and estimating the behaviour of the channels at that time, from prior observations during the same hours and days.

In this way the FEC policy can be more targeted towards the loss characteristics of a particular day and time. It can be argued that because of the multitude and anonymity of the receivers this is impossible, because as said before "there is no single FEC policy " that can do the job for all the MBone paths. But this problem can be eliminated by using the layering strategy (see figure 5.1 and section 4.1.2) or by simply evaluating an average scenario. In both cases there will be bandwidth savings, since the transmitter will not try to guard the data against a worst case scenario all of the times. If the system is designed to cope with 30% loss whatever the circumstances, then this means that when loss is down to 5 %, lots of bandwidth can be wasted not to mention that the extra redundancy might lead to congestion (unlikely) or simply cost more.

Also the extra redundancy at these times could be used to provide better quality of service which itself can bring more revenue. Added to that is the fact that FEC error recovery systems do not degrade gracefully when their limits are reached, meaning that when the loss rate, for example, surpasses the (lets say) 30% mark by even 1%, the system will quickly collapse. Hence it might be the case that this could be avoided with the correct planning discussed above.

5.1.2 Exponential Averaging

In order for the stored data to be of any use to the system, they have to be interpreted as to give meaningfull results and 'numbers' by which to tune the RLA system. This can be achieved in two ways, the first obvious way is to analyse the ping statistics from probing the MBone paths as described in chapter 3 and the second way is by augmenting such knowledge by estimating future behaviour.

Such prediction of future 'values' in, what can be described with, a time series can be achieved using a common technique[21, 13] known as *exponential averaging* or *exponential smoothing.* In this technique more recent or past values in a series are given more or less weight in such a way as to give more quick or conservative responce to the changes in a process's behaviour, therefore tracking closely and hence predicting future outcomes. Using this technique properties like the loss, arrivals or jitter can be very effectively estimated, as it is done in TCP Retransmission Time Out, or in ATM cell rate prediction, or in some multitasking operating systems scheduler algorithms, RTP sessions etc. For example a smoothed average $SV(k)$ of a series of measured values $V(i)$ could be

$$SV(k) = (1-a)V(k) + a(1-a)V(k-1) + a^2(1-a)V(k-2) + ... + a^k(1-a)V(1)$$

where $0 < a < 1$ hence giving a small value to a will quickly reflect a rapid change.

By utilising exponential averaging the system's operator can not only use the stored data to plan the best FEC policies, but also extract from these data, possible future fluctuations or change in behaviour of some channels, hence 'insulating' the multicasts even more.

5.1.3 Using receiver quality reports...

During the multicast sessions the receivers may not report reception quality and progress to the transmitter, for the reasons already described, but they can store such reports locally and later send these reports back to the transmitter for batch processing. Of course not all receivers need to give such reports, since representing receivers from each multicast isle can do so.

For this reporting, in batch mode, to take place the receivers may need to store their reports in miniature databases and based on some random variable, later send them off. This storage of the reports though, can be of benefit to the receivers, not only indirectly but also directly. If a receiver uses the information stored in the reports in conjunction with the exponential averaging technique (or not), then it can *cleverly tune itself for the next receptions.* In fact such knowledge of the past events can be very effectively used to adjust a very important parameter of the receiver, namely its *initial buffer.*

Although this may not seem so important it is. Maybe it is one of the most important tuning parameters of the receiver. Why this happens, is because the *initial buffering is responsible for the variable bit rate and jitter smoothing.* It can be argued that a large initial buffer will do this job fine. Though what is not known is how big this buffer should be and even if it was known, it should not be forgotten that the reception buffering is variable[12, 24, 9, 21] and that a very large buffer may cause a big lag between transmission and reception. Such as to be difficult to synchronise with the other medium streams, or ever worse, not to have enough memory to accommodate them.

Hence the receivers learn about the different sessions and *get ready to cope with even more jitter*, if necessary, next time without manual tuning . This statefull property of the receivers can help *newcomer receivers solicit seniors* about the appropriate buffering and even multicast layer selection, before the beginning of a session, or even during one.

5.1.4 The benefits of a Databases system

Therefore by using a database system to store statistics, the operator is offered two extra parameters by which to tune and customise the system. The first is *knowledge*, based on statistical observations of channels, in order to *decide on appropriate FEC* and the second is *prediction* of possible future phenomena which may otherwise degrade or improve the system's quality.

Also by using the receivers' reports, the transmitter can further augment its knowledge base by considering the receivers' view and opinions of each multicast session. Therefore the transmitter can check, contrast and compare its view of the channels's behaviours against the receivers's and calculate differences and any variance. Hence the prediction and analysis models can be contrasted and double checked against 'real transmission reports'. Such a feedback is so useful since it can be used to fine tune the prediction scheme and also to pin-point any deficiencies in transmission and FEC planning.

The great thing about this knowledge database approach, in both the receiving and transmitting ends, is that *the whole system can mature in time and tune itself* . Hence bringing *evolution and progress without human intervention*, yielding this way fewer maintenance costs and probably better quality.

5.2 On reception...

Although much talk and effort has gone so far to the 'transmission', nothing has been said about the effort that has to be put on reception. The *Receiver*, admittedly has to be very efficient in order to cope with the added complexity of the layered transmission session management and scrambling of packets (see chapter 4) as well as the advanced buffering, re-ordering policy, jitter prediction and delay smoothing.

For a lecture transmission the end-to-end delay does not really matter allot, what matters is the delay variation and the added packet loss that this may lead to. In order to deal with this situation, a receiver has to adapt to the jitter and since there is no feedback loop to the transmitter, it has to do this cleverly and autonomously. To achieve this the receiver needs to support an adaptive playout strategy and to predict any delays and delay variations of the incoming packets. For doing so exponential averaging (section 5.1.2) has to be employed together with variable buffering techniques. Hence silence (or rather noise) has to be added or deleted (less likely) at the beginnings of talkspurts.

To make this work propertly, the initial buffer length has to be appropriately larger than the time needed to receive, recover, decode and reorder a number of packets (and hence samples), equal to the audio fragment time (size) that is equivalent to the audio device's internal playout buffer. Hence avoiding buffer overruns and underruns (more likely). Also if layered multicast is to be used, the demultiplexing of the piggybacked samples will not be necessary and thus making the receiver less complex. Nevertheless a number of different buffers, equal to the number of 'listened multicast addresses' has to be present. These buffers have to be filled with data from different codecs and be used for FEC, therefore they should be valued according to priority(class). Using therefore different threads of execution, these buffers have to be written and 'decoded' synchronously and read out asynchronously according to loss and priority.

The receiver design and implementation is very complex, here only the 'tip of the iceberg' is just remotely presented, since it needs a study of its own. Such a receiver design was partially layed out, but was never completed since the project orientation changed.

5.3 Software Architecture

With Software Engineering being the discipline of applying engineering and quality-control methods to software development, the only way to tackle with the system's implementation difficulties would be to follow this concept.

The RLA code should follow the Object Oriented Technology paradigm, as close as possible, and should be layered according to different levels of abstraction in accordance with the 'black box' approach that is so often used in the electronics world. Software components should be created in such a way as to model the real and 'networking world' objects, such as the receivers ,transmitters, net sockets,codecs, screens, transparencies etc.

This approach should offer modularity and expandability since different components will interface together, and due to that interface-only view and layering, modules and components should be able to be changed and extended without distracting the whole system and/or architecture. For example when a new codec or transmission scheme has to be added the rest of the modules should stay unaffected and unaware of the change. The target should be towards a modular and multiplatform design, avoiding any monolithic implementations.

5.3.1 Software Built so far...

The original orientation of the project was to built a multiplatform framework that would facilitate an RLA implementation and that would support the work of Dr S. Ramanan and John Woods, but since backround information was needed in order to avoid any cowboy design and in order to truly appreciate the problem plus the constraints of the MSc course, this direction changed towards simulation, analysis and survey.

Nevertheless many software components were partially or fully designed, derived, extended (from previous work [14]) and built, these components can be said that 'did not make it to this report', though they can be found, together with the ones already mentioned in previous sections, as appendices to this report. Such components are networking components for UDP and TCP, a silence compressor, smart buffer objects, simple receiver and transmitter, components for audio recording and playback and more.

5.3.2 Emphasis on Multi-platform development

One of the most important aspects of any such remote lecture attendance system, should be its ability to inter(net)work and exist on multiple platforms. It is only logical (and true) to take for granted the heterogeneity of the Internet, hence the greater the deployment of the system, the greater the heterogeneity. This property therefore calls for one approach, this is multiplatform development (and hence deployment) of the 'future' system. It would be unrealistic to assume that such a system could only run on one platform and/or architecture.

Chapter 6

Concluding...

What has been discussed and presented in this report dealt with the audio transmission aspects of a system, that could be used to allow remote lecture attendance using the Internet's MBone infrastructure. The literature survey followed and unveilled what is already there and what work has already been done, in relevant areas (though by no means in all of them). Some experts's results and opinions were briefly reflected. Some of the experiments conducted tried to (remotely) match as possible these results, and they did. These experiments, analysis and observations were made, in order to support and extend the appreciation of the subject. In order to be in a strong position to suggest ways of implementing the system.

It is believed and suggested, based on the Internet traffic survey, that any FEC policy needed to protect the audio is very much dependent on the channel that is disseminated on. therefore no single policy is sufficient for the entire of the Internet's channels. Hence for each multicast many FEC schemes may be needed simultaneously. This multitude of FEC policies was shown to be able to be accommodated using layered transmission, hence to be efficient, flexible and to offer different grades of service. Since, like it was discussed, the receivers should decide on the FEC policy that best suits them.

A definition and description of how such a system would look like was presented and lacking any experience with something similar or close to it, its is believed that a good approximation was given and that any reader could be now in the position to appreciate its merits and properties. No specific quantitative requirements were given for the system, intentionally, since as said before an RLA system is so much human oriented, that in order to correctly target and design the system, the audience's 'likes' and 'dislikes' have to be present first.

Also some future views and suggestions of how the system should evolve were pictured. Proposing ways of tuning and panoplyding the system using knowledge of past events, stored in databases, both the transmitter and receiver ends.

The human factor is so prevalent in this work that really any correct implementation and design should follow first extensive experimentation on this factor. All the results, decisions and implementations during this project, were only driven by minimal subjective experimentation, based on the team's opinions which clearly do not represent these views of the potential audience.

At all times that this project was going on, care was taken to research and suggest for a remote lecture attendance system, in terms of feasibility, usability and current Internet technology developments, using todays technology rather than future 'would likes'. In some parts of this report maybe the style looked like a business case study and this was due to the fact that occasionally the system was viewed in terms of its commercial viability and potential, which certainly can have if backed appropriately, especially if the model based video coding research shows that loss can be tolerated.