# Experiences with Real-Time Streaming of Audio Streams

Karl Jonas Peter Kanzow Mathias Kretschmer

German National Research Center for Information Technology (GMD)
Schloß Birlinghoven, D-53754 St. Augustin, Germany
EMail: {firstname.lastname}@gmd.de

*Abstract* – **An increasing number of applications include real-time audio streaming over a communication network. Well-known examples are teleconferencing, tele-presentation, Internet Radio live streams and on-demand services.**

**In cooperation with various broadcasters GMD provides live and on-demand audio services over the Internet. Applications are based on commercial products as well as on own software developments.**

**This paper describes some approaches to provide a real-time audio service over the Internet, discusses problems and experiences and presents an outlook for a streaming service in the future.**

## I. Motivation

An increasing number of applications use local area networks and the Internet for data transmission. In addition to the traditional text and graphics transmission, provided namely by the World Wide Web (WWW), more and more audio and video services emerge. This paper focuses on audio services especially for the Internet with its currently low bandwidth.

In cooperation with several German broadcasters, GMD has implemented an integrated Internet Radio system which uses a WWW-interface to provide worldwide

- a live audio stream (the current radio programme)
- access to archives of audio streams, built of previously broadcast content. Here one can find the broadcasts of yesterday or all news on a given topic.

Since radio is very fast and about current affairs, the audio archives and additional informations have to be updated almost continuously, every minute, every day, seven days a week. This happens automatically to make the service efficient to the provider.

GMD uses audio streaming to implement Internet Radio services. Several streaming systems for audio (and video) exist on the market. This paper examines the features which make an audio streaming system suitable for an Internet Radio service and discusses how existing streaming systems meet these requirements.

Chapter two compares the two major methods to deliver audio content over a network - download and streaming. It also presents some information about encoding standards and transmission protocols.

Chapter three presents the components of typical audio streaming systems and discusses requirements a streaming system must fulfill to be used with Internet Radio services.

Chapter four presents two commercial streaming systems, Xing Streamworks and RealAudio and discusses how they meet or fail to meet the above requirements.

Chapter five describes running Internet Radio services by GMD which can be accessed over the Internet *now*.

Chapter six discusses problems and possible improvements leading to a prototype implementation for a future system described in the chapter.

## II. Delivering Audio Content Over a Low Bandwidth Network

### A. *Streaming and Downloading*

In general, audio content can be delivered over a network in two ways:

1. The audio file can be *downloaded* and then played from the local hard disk of the client.
2. The audio content can be *streamed* from the server to the client who decodes the received packets in real time, displays the content immediately and then discards the received data.

The advantages of the downloading method are:

- downloading works with any datarate and allows any audio quality one wants to offer.
- the file is transmitted error-free thus no quality reduction occurs during transmission.

But there are major drawbacks of this method of audio delivery:

- Download times are extremely long if high audio quality is provided over a low-bandwidth network. For example, a five-minute-long music clip encoded with 128kbps takes half an hour or more to download over a typical private Internet access with an effective long-term bandwidth of 20kbps. To reduce download times, audio quality must be reduced.
- The user has to load the complete file before he can listen to any part of it. He cannot preview the file to decide whether he is interested in the content.
- There is no possibility to provide a "live" service as re-broadcasting a radio programme into the network simultaneously with e.g. the terrestrial broadcast.

The advantages of streaming are:

- The user can listen to the content immediately after he has demanded for it. He can fast-forward and listen to other parts of the content without waiting for the whole file to download.
- It is possible to provide "live" services by encoding the audio signal in real time and sending the resulting audio data stream immediately to the client.

- To stream real time audio data, the transmission line has to provide the full bandwidth of the stream during the whole transmission period. This imposes a limit on the bandwidth and hence the quality of the audio stream. Concerning the Internet, the possible bandwidth is very low if you want to achieve the vast amount of private users connected with a modem and via a provider such as Compuserve, America Online (AOL), etc.
- The transmitted stream is very sensitive to network load which may cause lost or delayed data packets. This leads to drop-outs in the client audio output.
- To process the required flow control, additional server software (a “streaming server”) is needed.

It has to be considered that the main advantage of downloading, the possible high audio quality even over low bandwidth networks, is compensated in practical use by the required high download times.

The drawbacks of streaming, however, can be reduced by dynamically adapting the amount of transmitted data to the actual capacity of the network connection. To do this, the audio data must either be stored in a format that allows dropping parts of the data resulting in a “graceful degradation” or they must be stored in different formats that the server can choose the format best suited for the bandwidth of a given connection.

In conclusion, streaming is the suitable method to offer the user a fast access to the content. It is the only solution possible to offer a live stream.

### *B. HTTP Streaming*

The „Hypertext Transport Protocol“ (HTTP) [9] describes how data is transmitted in the World Wide Web.

HTTP “streaming” tries to reduce some of the drawbacks of downloading. The file is downloaded via HTTP (i.e no special Streaming Server is involved) but displayed by the client immediately when the data arrive. Thus, the user can listen to the content immediately, as with streaming. But there is no fast forward and no possibility for the system to react to insufficient network bandwidth or temporary network congestions. Data are transported via the reliable TCP/IP protocol stack, which is slower and less appropriate for audio streaming than e.g. UDP (see section D of this chapter).

### *C. Audio Formats*

All streaming systems use more or less different perceptual audio codecs to convey as much audio quality as possible to the listener. The currently most frequently used codecs are Dolby AC-3 based ones (i.e. RealAudio 3.0) and MPEG Layer 2/Layer 3 implementations (i.e. XING Streamworks, Fraunhofer’s L3Enc). GMD additionally uses AT&T’s PAC codec. All codec implementations are real-time capable on at least one hardware platform (i.e they are optimized for Intel Pentium Processors). Hence they can be used with Realtime-Audio-Streaming applications like “MBone Live Radio”.

The perceived audio quality differs from codec to codec and depends on the given bitrate. Real-Audio seems to produce the best results at very low bitrates between 8 and 16 kBit/s. At bitrates up to 112 kBit/s, AT&T’s PAC and duce the original with the lowest noticeable distortion. In fact, the difference between the original and the codec output is smaller the higher the requested bitrate is.

We will soon examine the differences in the error tolerance of the various codecs focusing on packet losses.

Additionally, we will test the whole streaming system under (simulated) Internet conditions. We will focus our work on the system‘s behavior on packet delays und packet losses.

### *D. Transmission Protocols for Audio Streaming*

Endsystem applications do not implement all communication features, but use exinsting communication protocols instead. Typically, a network protocol is used to forward datagrams across a network, and a transport protocol is used for end-to-end services. The combination of protocols is called the *protocol stack.*

The typical protocol stack which is used for data transmission across the internet is TCP [6] or UDP [7] on top of IP [5] (fig. 1).

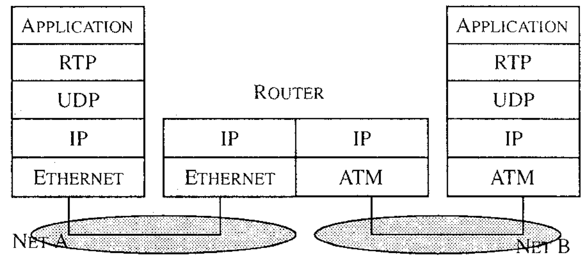


**Figure 1: Example for protocol stacks: application data is transported via RTP, UDP, IP and Ethernet/ATM**

The well-known TCP protocol provides reliable data transmission. It acknowledges received data to the sender then being able to retransmit lost data. Due to retransmission delays, however, TCP transmission is never continuous on an unreliable network. Compared to TCP, unreliability is a major *feature* of UDP, lost data is never retransmitted. For most streaming applications, packet loss is more acceptable than discontinuous presentation.

The Realtime Transport Protocol [8] can be used on top of UDP. It provides additional presentation information (timestamps, payload type) for the price of some additional transmission and processing overhead.

A combination of IP, UDP and RTP adds a minimum protocol overhead of (20+8+12=) 40 bytes to each datagram. While this is no problem for high quality (= high bandwidth) audio, it becomes important for 5 kBit/s streams where a 20 ms datagram results in 12 bytes payload (typically, payload size is increased, e.g. to 100 ms for very low bandwidth. IP/UDP/RTP header compression to a total of two bytes per datagram is under discussion).

### A. *The Components and How They Work Together*

An Internet Radio system as discussed here contains both a live stream and automatically updated audio archives. A suitable audio streaming system for this service must include

1. permanent realtime digitizing and encoding of the (analog) live audio signal into a streamable format;
2. permanent updating of the audio archives by recording parts or the whole encoded live audio data on hard disk;
3. a streaming server, capable of serving the live stream and the recorded streams;
4. streaming clients which receive and play the audio data stream on the client host's audio system.

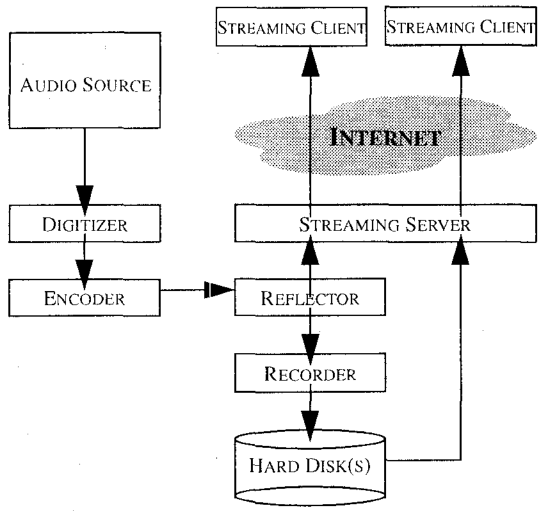


**Figure 2: Components of a typical Streaming System**

Fig. 2 shows a typical setup of a streaming system for an Internet Radio service. The *audio source* (e.g a radio tuner) delivers the content to be re-broadcast in the net. Its output, an analog audio signal, is digitized and encoded in real time by the *digitizer* (or analog-digital converter, ADC) and the *encoder* (which can be hardware or software).

The output of the encoder is compressed digital data in a streamable format. The *reflector* just doubles this stream and passes the first to the *streaming server.* The streaming server transmits the data to the listening *streaming client(s)*, if there are any currently demanding the live stream. Otherwise, the data are discarded.

The second data stream is passed to the *recorder.* The recorder stores the data on hard disk according to a schedule which determines file names, start and end times. The recorder thereby fills the archives with audio files of different length and names, depending on which content is meant to be stored. We call these audio files "recordings".

Whenever a streaming client demands a stored audio stream, the streaming server reads the according audio data from hard disk and transmits them to the client.

While working with several streaming systems we learned that for practical use the fulfilling of the above requirements is not enough. There are additional requirements which are essential for the Internet Radio to work, e.g.:

- The audio codec must produce sufficient quality at a very low bandwidth. Most Internet users receive a permanent data rate of 20kbps or less even if they are connected to their Internet provider via ISDN. To reach them, an audio stream must not exceed that datarate while providing understandable audio quality.
- The streaming server must be capable of serving parts of audio files instead of the whole recording only. A long recording can thus be divided into parts (which we call "clips") just by creating pointers instead of cutting it physically. This makes "cutting", automation and correction much faster and easier.

## IV. Commercial Streaming Systems

Today, many systems exist for streaming audio and/or video through low bandwidth networks as the Internet. Two of the most important ones are *Xing Streamworks* and *RealAudio* by Progressive Networks. We will discuss them in this chapter.

### A. *Xing Streamworks*

Streamworks by the American company Xing [10] is currently in the version 2.0. It supports both audio and video. In this paper, we will focus on audio only.

Streamworks uses MPEG Layer I and II audio codecs and a special "Low Bit Rate" (LBR) codec for very low bandwidths (<16kbps)

*The Components*

In terms of the setup described in Fig. 2, Xing offers the digitizer and the encoder as a single hardware component called the "Xing Audio Transmitter". Software includes a recorder which is called "Capture Tool", a streaming server ("Streamworks server") and streaming clients ("Streamworks Player"). There is no reflector available from Xing.

The *Xing Audio Transmitter* turns out to be a 486-PC with a special audio capturer/encoder ISA board, an ISA network interface board and DOS software.

The Transmitter digitizes and encodes an analog audio signal. The encoded audio data is transmitted as a stream of UDP packets into the LAN the Transmitter is connected to.

Since the Transmitter comes with no hard disk, it boots from a floppy disk containing DOS and ATRANS, Xing's encoding control software. A DOS-based, menu-driven configuration program lets the user specify parameters like the desired audio codec and datarate as well as the Transmitter's IP address and the destination IP address and port of the output stream.

The Transmitter's output stream is meant to be received by either the Capture Tool (to be stored and re-broadcast later) or by the Streamworks server (to be re-broadcast immediately as a live stream).

output stream as a recording. The tool takes two parameters - duration of the recording and the recording's file name.

The *Streamworks server*, which is available for several UNIX systems and Windows NT, receives the Transmitter's output stream and serves it as a live stream via the Internet/Intranet to listening Streamworks clients. The server is also capable of serving existing recordings as audio streams. These recordings can be:

- standard MPEG Layer I or II audio files
- LBR files, converted from Windows WAV format by DOS software available from Xing
- MPEG or LBR files encoded by the Transmitter and captured by the Capture Tool.

The server is capable of serving clips (sections of recordings). It is also able to offer different content depending on the client's requested bandwidth. To describe an audio clip, Streamworks uses small ASCII text files called "playfiles". A playfile defines a clip giving the filename of a recording and the start- and end times of the clip relative to the beginning of the recording. This data can depend on the client's requested bandwidth, making it possible to serve clips of different quality for each bandwidth.

To listen to a Xing based Internet radio one needs the *Streamworks Player* (the client software). It can be downloaded freely at Xing's WWW Site. It supports Windows 3.x and 95, PowerMacintosh, Solaris, IRIX and Linux.

*GMD's addition to Streamworks*

As delivered, Streamworks meets most of the requirements described in chapter III. When working with Streamworks, we found out that Xing's Capture Tool is not very well implemented. It consumes nearly the whole CPU power. Furthermore, it has to be re-started for every new audio file to create, which leads to unpredictable delays.

In addition, there is no reflector within the Streamworks package. A stream can be either broadcast live or stored for later use, not both.

GMD implemented a reflector and a new recorder for Xing's LBR format audio streams. The software runs on Solaris. The recorder is programmable and stores the live stream coming from Xing's Transmitter via the reflector into audio files according to a desired schedule. It consumes less than 1% of a Sun SS20 CPU (SuperSparc 2 at 60MHz). It does not need to be stopped and re-started to produce a new recording, therefore it can start faster and with more accuracy to a time-based schedule.

With these extensions, Streamworks meets the requirements described in chapter III. Its LBR audio codec delivers acceptable speech quality at datarates below 15kbps. It plays sections of recordings with good accuracy. The client software supports many platforms, thus reaching a maximum amount of listeners. GMD chose Streamworks for their Internet "PersonalR@dio" (see next chapter). Feedback mails show that this service is used worldwide with reportedly good audio quality.

In spring 1996, when we chose Streamworks for our projects, RealAudio by Progressive Networks [4] was already the market leader for Internet audio streaming, but Streamworks provided better audio quality at lower bandwidth.

With the release of version 3.0 in autumn 1996, RealAudio provides several new audio codecs which excel those of Streamworks. For radio programmes containing mainly speech there is a codec at 6,5kbps delivering better audio quality than Streamworks's LBR at 11kbps. When it comes to music, jingles etc., the 6,5kbps codec is not usable any more, but there are several other low bandwidth codecs suitable for music, as well.

*The components*

The RealAudio system consists of software only. In terms of Fig. 2, digitizer and encoder are parts of a Windows PC. The digitizer may be any Windows MME compatible sound card (e.g. Soundblaster) while the encoder is software by Progressive Networks ("RealAudio Encoder"). A fast 486-PC can digitize and encode audio in real time. Again, the resulting audio stream is transmitted via a network interface to the streaming server ("RealAudio server"). The RealAudio Encoder is also capable of storing the audio data as a file, acting as a reflector and recorder at the same time.

The *RealAudio Encoder* is Windows software capable of encoding audio data directly from the digitizer to a live stream and/or a recording. It can also encode an existing WAV audio file into a streamable format. A DOS version makes it easy to encode the same WAV file into several streamable formats with different bandwidths and qualities using DOS batch processing. A Solaris CLI version encodes WAV and AU format audio files.

The *RealAudio server* is available for several platforms (Windows NT, Solaris). It can play live streams and sections of recordings with good accuracy and chooses between files according to the bandwidth of the client's connection. To describe a clip, RealAudio uses text files called "metafiles" similar to Xing's playfiles.

The *RealAudio player* can be downloaded freely for Windows (3.x/95), Macintosh, OS/2 and several UNIX systems.

With its new version 3.0, RealAudio made a great progress. It meets the requirements described in chapter III.

### C. *Which One to Use?*

RealAudio excels Streamworks mainly in terms of audio quality/bandwidth relation and player reliability. Not in the scope of this paper, but influencing the decision what to use are the video extensions. Both systems support video streaming, but RealAudio (then called "RealVideo") works much better. We are currently preparing to replace Streamworks with RealAudio/RealVideo in our projects.

*A. PersonalR@dio*

*PersonalR@dio* [3] is the first German Internet radio with both a live stream and seven days audio archives. "B5aktuell", the news channel of Bayerischer Rundfunk (BR), Munich, is the first complete radio programme which is fed into the Internet in parallel with its terrestrial broadcasting and simultaneously stored as digital audio archives on the server. This enables a subsequent on-demand access to any broadcast of the last seven days via the World-Wide Web (WWW).

*Current Standings*

*PersonalR@dio* uses Xing Streamworks with GMD's extensions described in chapter IV. Its setup (Fig. 3) is similar to the generic setup shown in Fig. 2.

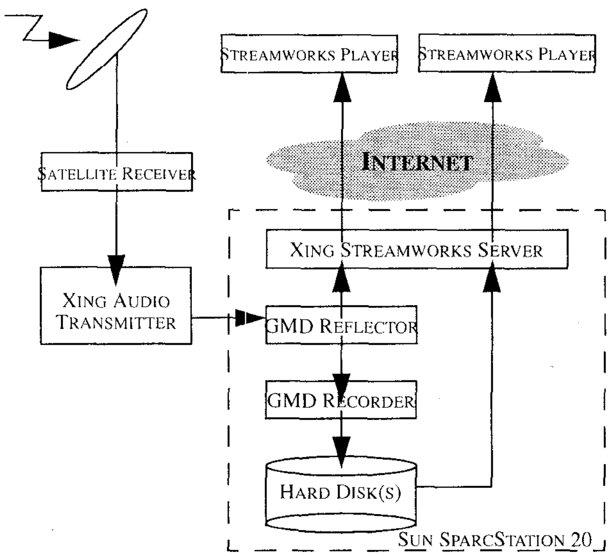


**Figure 3: Setup of PersonalR@dio**

The program is received by a satellite receiver. The analog audio signal is encoded by a Streamworks Audio Transmitter. We use the LBR audio codec at 11kbps. This leads to an understandable shortwave-like audio quality which is sufficient since "B5aktuell" is a news channel and mainly consists of spoken words. With 11kbps, the programme is accessible throughout the world as has been confirmed by many listeners via e-mail.

GMD's recorder software stores recordings every quarter of an hour according to the time schedule of "B5aktuell". At the same time, it overwrites recordings which are older than a week. This self-updating archives run automatically with no human effort needed.

The Web interface of *PersonalR@dio* is driven by CGI scripts which always produce up-to-date HTML pages when accessed. The user can choose a date and a time out of the last seven days and immediately gets the audio broadcast of that time. Thus, the user is independent of broadcasting times.

GMD released a similar Internet radio for Mitteldeutscher Rundfunk [2].

For the future, it is planned to provide a database with additional textual information. The user can search for topics instead of broadcasting times and quickly finds the news he or she is interested in. To achieve this, GMD plans to connect to the broadcaster's internal digital schedule to get content descriptions and timestamps automatically from the station. They will be processed in our database giving audio clips. Since Streamworks (and RealAudio) is capable of playing sections of recordings without cutting them physically, information will be accessible in the Web a few seconds after the content is broadcast by the radio station.

*B. An Internet Radio using RealAudio*

GMD currently does not provide an audio-only service using RealAudio, but we implemented "Rundschau im Internet" („Bavarian TV News in the Internet") [1]. This is a German language video-on-demand service using RealVideo, the successor of RealAudio. Hence this paper is only concerning audio streaming, we describe a possible audio-only service using RealAudio which we could implement by using the experience with the system during the setup of "Rundschau".

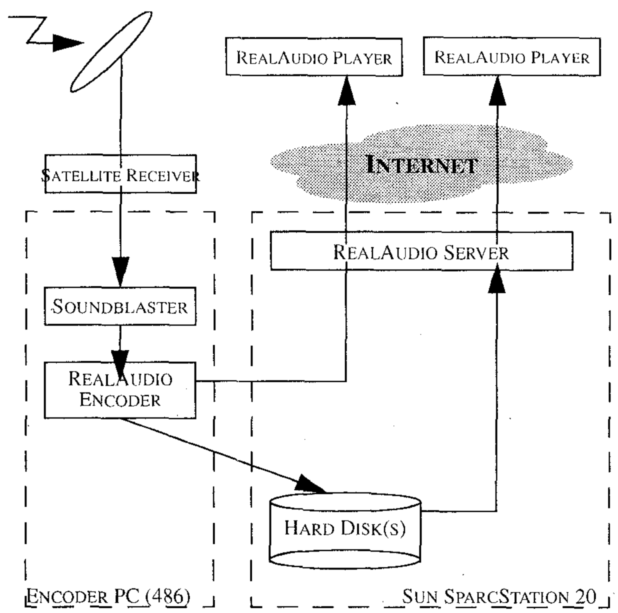


**Figure 4: Setup of an Internet Radio with RealAudio**

The programme is received by a satellite receiver or any other kind of radio tuner. The analog audio signal is plugged into Line Input of the Encoder PC's sound board. The RealAudio Encoder running on the same PC converts the digitized audio signal into the RealAudio format. It sends the RealAudio stream to the RealAudio server running on a Solaris or Windows NT machine. Simultaneously, the RealAudio Encoder stores the audio data on hard disk.

The RealAudio server transmits the live stream or clips from hard disk to listening RealAudio players.

DOCKET ALARM

# Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

## Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

## Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

## Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

## API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

**LAW FIRMS**

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

**FINANCIAL INSTITUTIONS**

Litigation and bankruptcy checks for companies and debtors.

**E-DISCOVERY AND LEGAL VENDORS**

Sync your system to PACER to automate legal marketing.

**WHAT WILL YOU BUILD?** | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.