

Project Title:

Streaming Video for the Internet

Client's Name (Country):

Oplayo, Helsinki, Finland

The Challenge:

The client had obtained the rights to the "MVQ" video-encoding algorithm. He required the implementation of MVQ encoding software and real-time audio/video data delivery software for transmission over the Internet.

Prior to project start-up, the client provided the Lanit-Tercom team with the following list of stringent requirements:

Encoder

- The encoder must be identical to those used in other Oplayo products;
- It must allow features to automatically speed encoding;
- Encoding must be carried out faster than in real-time in order to provide for other lags in the system;
- The audio must be interleaved with video.

Broadcaster Server

- The broadcaster must accept input from any video source, such as video boards;
- If possible, the SDP file and vector libraries must be sent from the broadcaster to the streaming server;
- Installation and setup should be automatic.

Streaming Server

- The server must provide both the streams and decoder applet because of connection limitations in a JVM browser;
- The stream must run on top of HTTP to avoid applet configuration problems. If possible, no changes should be made to the server code so that the standard version can be used;
- Relay setup must be tested;
- Installation must be simplified;
- Multi-cast support needs to be analysed.

Streaming Audio/Video Delivery via the Internet

The Solution:

The development team at Lanit-Tercom understood the component requirements in order to broadcast real-time video:

- Encoder - fast enough to encode incoming format to MVQ without causing delays;
- Broadcaster – for capturing camera output. Uses an encoding library to encode the data and send it to the streaming server, which handles client connections;
- Streaming server - accepts client requests for video streams. The streaming server, separated from the broadcaster, does not need to physically reside in the same machine. In addition, existing streaming servers that implement Internet standards can be used;
- Decoding applet - requires some changes from the standard Oplayo decoder applet in order to capture and read video stream at any point of the broadcast.

As a result of this project, the following real-time environment broadcasting system has been implemented:

- The system has 2 to 4 cameras, which are edited by a video board;
- One or two broadcasters run broadcasting with a total of two streams (high and low bandwidth), with audio provided;
- A single streaming server sends video streams to clients. Another relaying streaming server may be implemented in the same local network as the broadcaster(s);
- An applet is downloaded from the same server as the stream. The video is streamed using RTP over HTTP to avoid security problems.
- Streaming over TCP has been tested. RTP over UDP is also supported, but has not been thoroughly tested in that configuration (this would require extensive firewall/proxy set-ups).

To implement the real-time MVQ encoder the development team took their previous implementation of the MVQ encoder (written from scratch), and modified it to perform fast enough to handle realtime data compression.

The developers successfully implemented the Internet delivery task based on a strong knowledge of the Darwin Server (Apple). The broadcaster, developed from scratch, is a Windows 2000-based software, which can read output from digital cameras connected to the system. The streaming server is an unmodified Darwin Streaming Server, and provides connections and data to multiple clients.

Streaming Audio/Video Delivery via the Internet

Technologies and Tools

- C/C++
- Java
- Knowledge of image and video formats
- Optimization techniques
- Streaming Network protocols

Dates

2001

Customer Feedback

"Working with the partner enabled us to speed up the product development and thus improved our speed to market" Anttoni Vesterinen, Oplayo CEO

Streaming Audio/Video Delivery via the Internet