



SECOND EDITION

Praise for the first edition:

"For broadcasters and web developers involved in media delivery across the web, the book could be a very useful first step in understanding the basics of streaming technologies".

—European Broadcasting Union
www.ebu.ch

David Austerberry

**THE TECHNOLOGY OF
VIDEO & AUDIO
STREAMING**

A quick-start guide to digital media—includes streaming to wireless devices and up-to-date technology information for streaming companies and products



The Technology of Video and Audio Streaming

Second Edition

David Austerberry



ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Focal Press is an imprint of Elsevier




Focal Press
is An imprint of Elsevier.

200 Wheeler Road, Burlington, MA 01803, USA
Linacre House, Jordan Hill, Oxford OX2 8DP, UK

Copyright © 2005, David Austerberry. All rights reserved.

The right of David Austerberry to be identified as the author of this work has been asserted in accordance with the Copyright, Designs and Patents Act 1988

No part of this publication may be reproduced in any material form (including photocopying or storing in any medium by electronic means and whether or not transiently or incidentally to some other use of this publication) without the written permission of the copyright holder except in accordance with the provisions of the Copyright, Designs and Patents Act 1988 or under the terms of a licence issued by the Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London, England w1T4LP. Applications for the copyright holder's written permission to reproduce any part of this publication should be addressed to the publisher.

 Recognizing the importance of preserving what has been written, Elsevier prints its books on acid-free paper whenever possible.

Library of Congress Cataloging-in-Publication Data

Austerberry, David.

The technology of video and audio streaming / David Austerberry. – 2nd ed.
p. cm.

Includes bibliographical references and index.

ISBN 0-240-80580-1

1. Streaming technology (Telecommunications) 2. Digital video. 3. Sound –
Recording and reproducing – Digital techniques. I. Title.

TK5105.386 .A97 2004

006.7'876 – dc22

2004017485

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library.

ISBN: 0240805801

For information on all Focal Press publications visit our website at
www.books.elsevier.com

04 05 06 07 08 09 10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

Contents

Preface	ix
Acknowledgments	xi
Section 1. Basics	1
1 Introduction	3
500 years of print development	3
100 years of the moving image	4
The Web meets television	5
Convergence	7
What is streaming?	7
Applications	9
How this book is organized	10
Summary	10
2 IP networks and telecommunications	13
Introduction	13
Network layers	14
Telecommunications	25
The local loop	30
Summary	38
3 The World Wide Web	40
Introduction	40
WWW	42
Web graphics	44
Proprietary tools	48
Web servers	48
Summary	51

4 Video formats	52
Introduction	52
Scanning	53
Color space conversion	56
Digital component coding	61
Videotape formats	65
Time code	72
Interconnection standards	74
High definition	76
Summary	77
5 Video compression	78
Introduction	78
Compression basics	79
Compression algorithms	80
Discrete cosine transform	84
Compression codecs	87
MPEG compression	89
Proprietary architectures	98
Summary	101
6 Audio compression	102
Introduction	102
Analog compression	103
Digital audio	104
The ear and psychoacoustics	110
The human voice	112
Lossy compression	114
Codecs	117
Codec standards	118
Proprietary codecs	127
Open-source codecs	128
Summary	129
Section 2. Streaming	131
7 Introduction to streaming media	133
Introduction	133
What are the applications of streaming?	134
The streaming architecture	138
Bandwidth, bits, and bytes	147

Proprietary codec architectures	149
Summary	152
8 Video encoding	154
Introduction	154
Video capture	159
Compression	167
Encoding enhancements	170
Encoding products	173
Limits on file sizes	175
Summary	177
9 Audio encoding	179
Introduction	179
Audio formats	181
Capture	184
Encoding	186
File formats	189
Summary	192
10 Preprocessing	193
Introduction	193
Video processing	193
Audio	200
Summary	207
11 Stream serving	209
Introduction	209
Streaming	211
Webcasting	218
On-demand serving	222
Inserting advertisements	222
Playlists	224
Logging and statistics	225
Proprietary server architectures	227
Server deployment	229
Summary	232
12 Live webcasting	233
Introduction	233
Planning a webcast	233
Video capture	237

Graphics	238
Audio capture	238
Encoding	241
Summary	243
13 Media players	244
Introduction	244
Portals, players, and plug-ins	245
Digital Rights Management	256
Summary	257
Section 3. Associated Technologies and Applications	259
14 Rights management	261
Introduction	261
The value chain	264
Digital Rights Management	265
The rights management parties	270
System integration	274
Encryption	276
Watermarking	277
Security	279
XrML	280
Examples of DRM products	282
MPEG-4	286
Summary	287
15 Content distribution	289
Introduction	289
Content delivery networks	291
Corporate intranets	300
Improving the QoS	304
Satellite delivery	306
Summary	307
16 Applications for streaming media	309
Introduction	309
Summary	322
Glossary	327
Abbreviations	331
Index	335

THE TECHNOLOGY OF VIDEO & AUDIO STREAMING SECOND EDITION

David Austerberry

Learn the end-to-end process, starting with capture from a video or audio source through to the consumer's media player

A quick-start guide to streaming media technologies

How to monetize content and protect revenue with digital rights management

For broadcasters, web developers, and project managers implementing streaming media systems, David Austerberry shows how to deploy the technology on your site, from video and audio capture through to the consumer's media player.

The book first deals with Internet basics and gives a thorough coverage of telecommunications networks and the last mile to the home. Video and audio formats are covered, as well as compression standards including Windows Media and MPEG-4. The book then guides you through the streaming process, showing in-depth how to encode audio and video. The deployment of media servers, live webcasting and how the stream is displayed by the consumer's media player are also covered.

A final section on associated technologies illustrates how you can protect your revenue sources with digital rights management, looks at content delivery networks, and provides examples of successful streaming applications.

The supporting website www.davidausterberry.com/streaming.html offers updated links to sources of information, manufacturers and suppliers.

David Austerberry is co-owner of the new media communications consultancy, **Informed Sauce**. He has worked with streaming media since the late nineties. Before the move to streaming, he was a product manager for several broadcast equipment manufacturers, and can also count many years with a leading radio and television broadcaster.

CONTENTS

Basics:

- Introduction
- IP Networks and Telecommunications
- The World Wide Web
- Video Formats
- Video Compression
- Audio Compression

Streaming:

- Introduction to Streaming Media
- Video Encoding
- Audio Encoding
- Pre-Processing
- Stream Serving
- Live Webcasting
- Media Players

Associated Technologies and Applications:

- Rights Management
- Content Distribution
- Applications

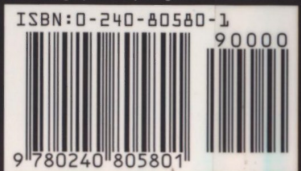
Glossary



Focal Press
An imprint of Elsevier
www.focalpress.com

If you enjoyed this book please post a review to your favorite online bookstore today.

Cover Imagery: © Getty Images



11

Stream serving

Introduction

What happens once you successfully have encoded your multimedia content? Much like publishing a web page, the file is uploaded to the delivery server. That is where things diverge. A conventional web server simply downloads the media file. A streaming server has to manage the delivery rate of the stream to give real-time playback. In addition, the streaming server supports VCR-like control of the media clip.

When a browser requests a web page, the files are delivered as fast as the network connection allows. TCP manages an error-free transmission by retransmitting lost packets, but the download time depends upon the intervening bandwidth available. TCP starts at a low rate then ramps up to the maximum that can be achieved. An accurate delivery is ensured, but timely delivery cannot be guaranteed. Streaming media has opposite requirements: the delivery must be in real-time, but reasonable levels of transmission errors can be accepted.

Streaming servers can be proprietary to an architecture or designed to handle standard formats like MPEG-4. The system architecture can vary from a single machine serving a small corporate training site, to large distributed server farms, capable of serving hundreds of thousands of streams for live events like breaking news footage, fashion shows, and rock concerts.

Streaming can be delivered as a push or pull process. Push is used to stream live or prerecorded content as a webcast – this is the television model. Push streaming can be used for web channels or live events. Alternatively, the user can pull prerecorded content on-demand. This interactive experience is akin to using a CD-ROM or a web browser.

A webcast can be a mix of live and prerecorded content. With live events the server is acting as a distribution point, just echoing the stream onto the viewers. For the prerecorded content the server has two functions. The first is to recall the content from the local disk storage arrays and the second is to control the stream delivery rate.

In the case of interactive content the client or player is requesting the files from the server. With the simulated-live webcast, the server runs a playlist, which streams files at the scheduled time to the player.

Table 11.1 Web Server versus Streaming Server

	<i>Web server</i>	<i>Streaming server</i>
Advantages	Part of existing infrastructure No additional expertise or training for IT staff	Optimized media delivery Dynamic stream control Interactive media control Multicast support Improved server hardware utilization Supports live webcasting
Disadvantages	None of the streaming server advantages Only supports progressive download	Additional equipment required

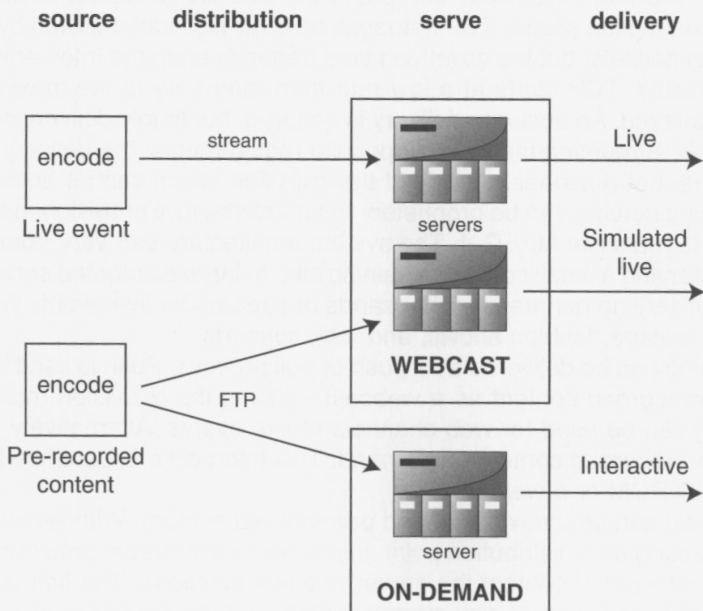


Figure 11.1 Webcasting and on-demand.

Streaming

What is a streaming server? The most-used server for the delivery of multimedia content is the web server, typified by Apache. Web servers use HTTP over TCP/IP to deliver HTML pages and their associated image files.

TCP/IP is used as the transport layer over the Internet. The files are downloaded to the web browser cache as fast as the system allows. TCP incorporates flow control to manage the download rate. There is no predetermined rate for delivery. TCP will increase the data rate until network packet loss indicates that the network is congested. At this point, the rate backs off. Another constraint is the receive buffer. TCP uses a sliding window of data in transit. The receiver processes packets as they arrive. If data arrives too fast, the receive buffer will overflow. The receiver sends messages to the transmitter to slow down, to stop the buffer from filling.

Suppose that you want to stream a stream encoded at 40 kbit/s. The TCP transmissions could start at 10 kbit/s. The transmitter then ramps up to 100 kbit/s, where network congestion sets the upper limit. Suppose other users come on to the network, and the transmission throttles back to 30 kbit/s. At no time has the data rate matched the data rate at which the stream was encoded.

Now consider if this clip lasts for 30 seconds, the complete file size is 150 kbytes. This is downloaded to the browser cache – not a great problem. Now suppose we move up to a 20-minute presentation encoded at 300 kbit/s. Now the file size is 45 Mbytes – very large for the cache. This has been the way that the Flash player handled video files, but Flash was limited to short clips.

When you stream content in real-time, the media packets are processed by the player as they arrive. There is no local caching, so the local storage issues are solved. This may not seem an issue to PC users, but many media players have very limited memory, for example set-top boxes and mobile devices. The problem with Flash also has gone, Macromedia now has developed the Flash player to support streaming of longform video, and the content is rendered then discarded.

There is still the rate control problem. If the stream is encoded at 40 kbit/s it must be delivered at that rate for satisfactory viewing. One of the functions of the transport layer protocol is to regulate the stream rate. But what happens in the example where the network is congested and the best rate is 30 kbit/s? The player runs out of data and stops, one of the main complaints about streaming.

There are ways around this, but the first is to encode at a rate below that which will suit the worst-case network conditions. That may be hard to predict, so there are more sophisticated ways; the usual is to encode at several rates,

then automatically select the optimum rate for the propagation conditions. This switching between different rate files is another task for the server.

One of the great attractions of streaming is the interactivity. The user can navigate the clip with VCR controls. The server has to locate and serve the correct portions of the clip using an index.

From these examples, it can be seen that the streaming server has several additional functions over a standard web server:

- Real-time flow control
- Intelligent stream switching
- Interactive clip navigation

HTTP does not support any of this functionality, so new protocols were developed for streaming media. Under the auspices of the IETF several new protocols were developed for multimedia real-time file exchange: RTSP, RTP, and RTCP. There are also a number of proprietary protocols using similar principles. Windows Media originally used the Microsoft Media Server (MMS) for the delivery framework (but now supports RTSP); the stream is in Advanced System Format (ASF).

Real-Time Streaming Protocol (RTSP) is the framework that can be used for the interactive VCR-like control of the playback (Play, Pause, etc.). It is also used to retrieve the relevant media file from the disk storage array. RTSP also can be used to announce the availability of additional media streams in, for example, a live webcast. Real-Time Protocol (RTP) is used for the media data packets. The Real-Time Control Protocol (RTCP) provides feedback from the player to indicate the quality of the stream. It can report packet loss and out-of-order packets. The server can then react to congested network conditions by lowering the video frame rate or gear-shifting to a file encoded at a lower bit rate. The real-time media stream can be delivered by UDP or TCP over IP; the choice depends upon propagation conditions. The control protocols use TCP/IP for the bidirectional client-server connection.

Streaming file formats

To stream media files in real-time, they must be wrapped by one of the streaming formats. These formats have timing control information that can be used by the server to manage the flow rate. If the client is using interactive control, the file index aids the navigation.

The main formats are MPEG-4 (mp4), the Microsoft advanced system format (.wmv and .wma extensions if created by Windows Media codecs, .asf if not), RealNetworks (.rm and .ra), and QuickTime hinted movies (.mov extension).

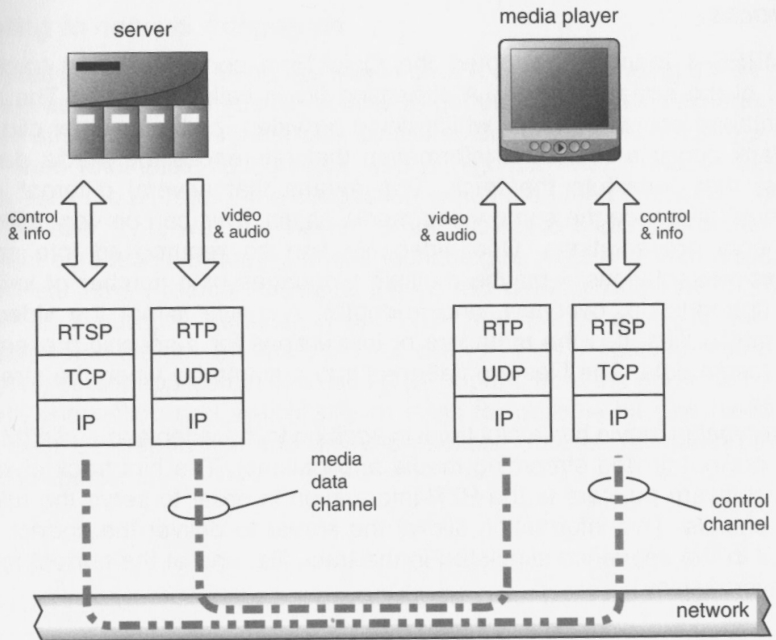


Figure 11.2 The streaming protocol stack.

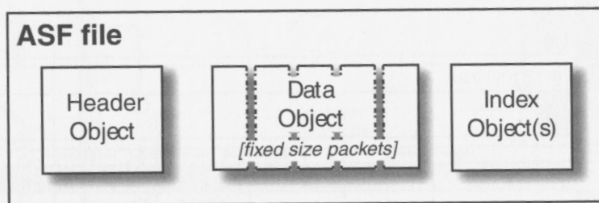


Figure 11.3 Typical streaming file format.

Web server and streaming server

If you already have a web site with web server capacity, and you stream only occasionally, it is possible to use that web server to deliver streaming files. The web server will use HTTP over TCP/IP. There will be no control of the stream delivery rate beyond buffer overflow in the TCP/IP stack.

Hint tracks

The MPEG-4 team has adopted the QuickTime concept of hint tracks for control of the stream delivery. A streaming file is called a movie. The movie file container contains tracks, which could be video, audio, or other clip data. The track consists of control information that references the media data (or objects) that constitute the track. This means that several different movie files could reference the same video media object. This can be very useful for rich media presentations. One video file can be repurposed into several different presentations – maybe multiple languages or a number of levels of detail (introduction, overview, and in-depth). A movie is not the video and audio media files; it is the metadata or instructions for a specific presentation of the media data. The files are flattened into a single file when the stream is encoded.

A streamable movie has a hint track in addition to the video and audio (MPEG-4 files are not limited streaming media applications). The hint track gives the server software pointers to the RTP information in order to serve the relevant media chunks. This information allows the server to deliver the correct video material in the sequence stipulated in the track file, and at the correct rate for the player display.

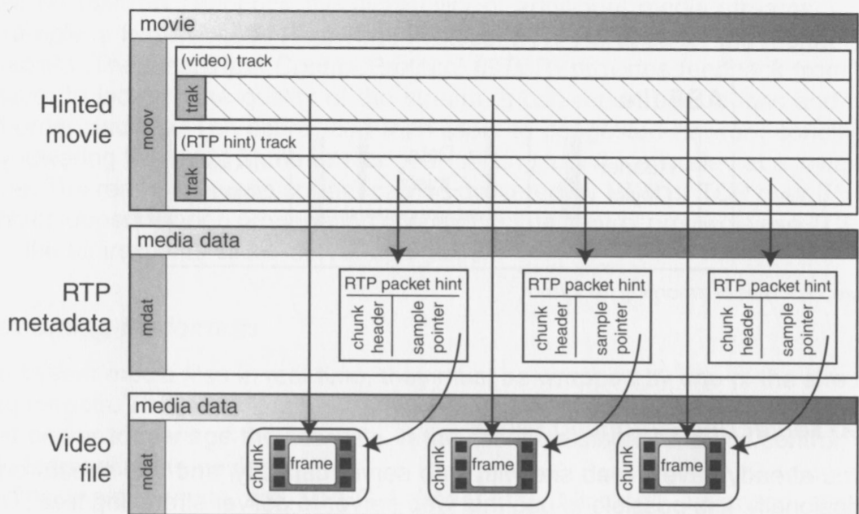


Figure 11.4 Typical streaming file format.

Adapting to network congestion

Both RealNetworks and Windows Media offer a way of changing the bit rate of a stream as network congestion varies. To get the best viewing experience we want to stream at the highest rate possible. But if the network slows down, rather than attempting to continue with a high bit rate, it makes sense to throttle back the bit rate. If the congestion eases then the bit rate can revert to a higher level. That way the viewer is not subject to stalling streams, just a graceful degradation in quality. These technologies work only with unicasting.

The Real-Time Protocol maintains the correct delivery rate over UDP/IP (or TCP/IP if bandwidth permits). The RTSP framework supports the client interaction with the stream, the VCR-like controls Play, Pause, and so on. The streaming server application can use RTCP reports from the player to measure network congestion and switch stream rates for multiple bit rate media files. The player can report lost and delayed packets, and the reception of out-of-sequence packets.

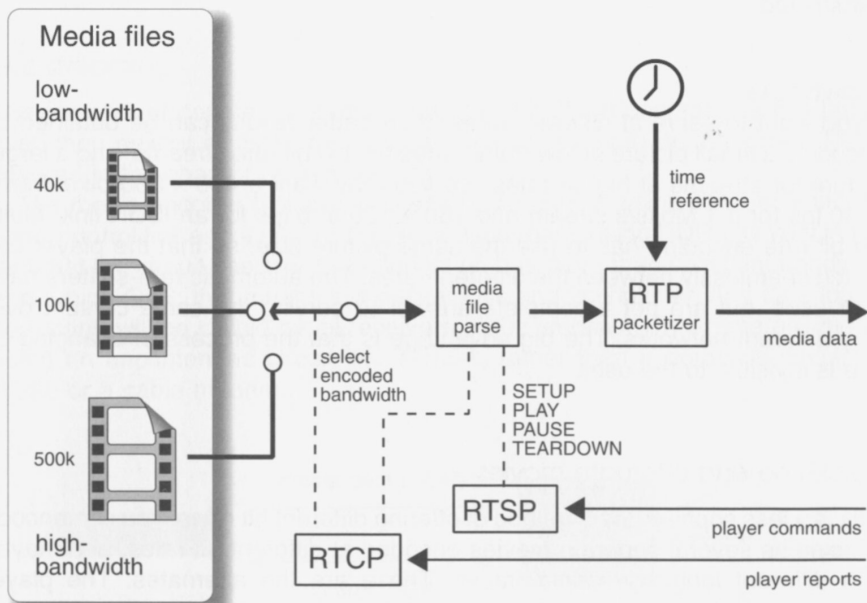


Figure 11.5 Streaming control.

RealNetworks SureStream

SureStream allows different encoding rates to be combined into a single media file. The streaming server will choose the appropriate data rate for the prevailing conditions by negotiating with the player. The lowest rate, called a duress stream, is a backstop, and will be streamed if congestion is very bad. When you are calculating the server disk space, a SureStream file takes the space of the sum of all the components. Because only the Helix Server can extract the correct component, a web server will serve the file in its entirety, including all the different rates.

Windows Media Intelligent Streaming

This is a similar feature, allowing multiple constant bit rate streams to be encoded and wrapped in a single file. The streaming server will then stream the best rate for the current network conditions. Windows Media Encoder comes with predefined multiple bit rate profiles, but the profiles also can be customized to suit your special requirements. If you want to multicast a file that has been coded at multiple bit rates, only the highest rate will be transmitted.

Drawbacks

If you want to serve at different rates, then better results can be obtained by encoding a small picture at low frame rates for low bit rate streams, and a larger picture for streams at higher rates. So you may want a 480×360 pixel frame at 30 fps for a 1 Mbits/s stream and 160×120 at 6 fps for an ISDN link. Multiple bit rate encoding has to use the same picture size, so that the player can switch seamlessly between the different rates. The automatic rate-shifters have their uses, but are not a complete answer to serving the same content over very different networks. The big advantage is that the process of changing bit rate is invisible to the user.

QuickTime and alternate movies

This is a less sophisticated method of offering different bit rates. You can encode a movie as several separate movies encoded at different bit rates, and maybe with different language audio tracks. These are the alternates: The player follows a pointer to the master movie, which then references the alternate movies. The player negotiates with the server to request the correct alternate file for the player settings.

MPEG-4 and scalable streams

The MPEG team has proposed a different way to cope with variable network bandwidth. The server transmits a basic low-resolution stream. Additional helper streams can carry more detail. If the bandwidth is available then these extra streams allow a better quality picture to be assembled by the player.

MPEG-4 also supports scalable encoding. This means that a basic player may decode only part of the stream to create the video, albeit at a lower quality than a more complex player, which can decode and display all the stream information.

Loading content

Whether you are using a managed service or doing your own serving, the first step is to deliver your content to the streaming servers. The encoding probably takes place near the video editing facility or, for a live webcast, at the venue. The servers have to be located close to an Internet backbone, unless you are streaming only over a local area. So in all probability the encoder and server are separated geographically. The simplest way to deliver the content is via a file transfer, using FTP. Some encoding systems have the ability to transfer a file automatically, immediately after the encoding has finished.

Live streaming

The file can, of course, be sent on a CD-ROM. If the content is a live broadcast, then neither of these methods is suitable; it has to be streamed. This is covered in the chapter on live webcasts.

The media encoder typically connects to the server using TCP for a bidirectional control link and a unidirectional media stream, using UDP. It is very important that the circuit used for this connection has more than sufficient bandwidth and a high QoS; that means low packet loss and timing jitter. Any data loss or corruption will be visible by all receivers of the webcast. This generally means using an uncontended circuit like T-1/E-1, rather than a domestic circuit like ADSL or a cable modem.

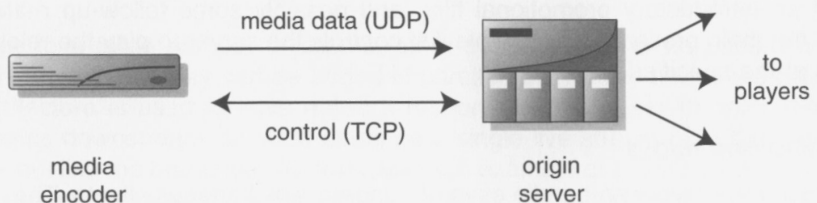


Figure 11.6 Encoder connections.

Announcing the content

The player usually locates streaming media by a hyperlink embedded in a web page. This link contains not only the URL for the content, but also the instructions to start the media player.

Web links

The usual way to announce streaming media files is by a hyperlink in a web page. The link points to a small file on the web server. Windows Media calls this the stream redirector, or ASX. Real uses the RealAudio Metafile or Ram file. Once the browser receives this file, using the MIME type, the metafile is passed to a streaming plug-in. The metafile has the full address and filename for the streaming content. The media player then transmits a request to the specified streaming server for the media file. This may use MMS or RTSP for communication rather than the HTTP used with the web server. If all goes well the correct clip is streamed to the player – success.

The metafiles can list several files to be played in sequence.

SMIL

If you are streaming rich media, then a number of different clips and images have to be synchronized for correct playback at the client. One way to do this is to use the Synchronized Multimedia Integration Language (SMIL) to write a control file.

SMIL is supported by the QuickTime, Real, and Windows Media architectures.

Webcasting

Webcasting can be live, prerecorded, or a mix of both. A webcast at its simplest can be just a single clip. If you want to play out a sequence of clips you can set up a playlist. Even if you are streaming a presentation, you may want an introductory promotional film, and possibly some follow-up material after the main presentation. The playlist controls the server to play the relevant clips at the specified time.

Splitting and relays

A streaming server can handle several hundred simultaneous clients. The only real way to establish how many clients would be by conducting load tests. To

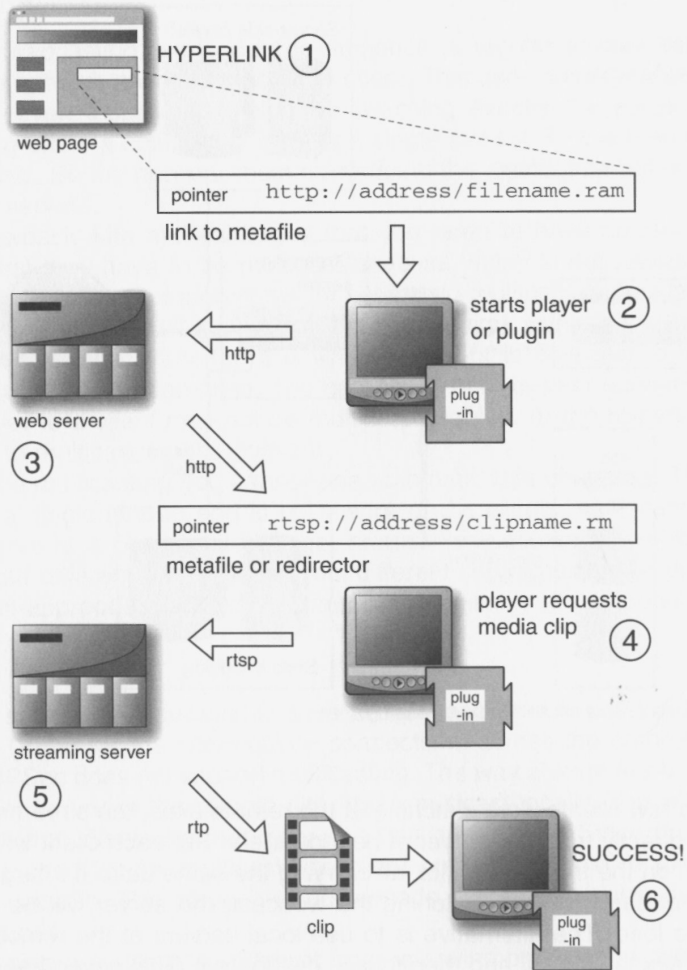


Figure 11.7 Linking to streaming media.

serve to a large number of clients, extra servers have to be used. For on-demand serving, they can be added in parallel, but for live streams a different architecture is used to save network resources. A relay server splits the live streams downstream, so what starts as a single live stream from the encoder fans out like the branches of a tree. Take the example of a CEO's webcast from the headquarters on the west coast to two other remote sites on the south and east coasts.

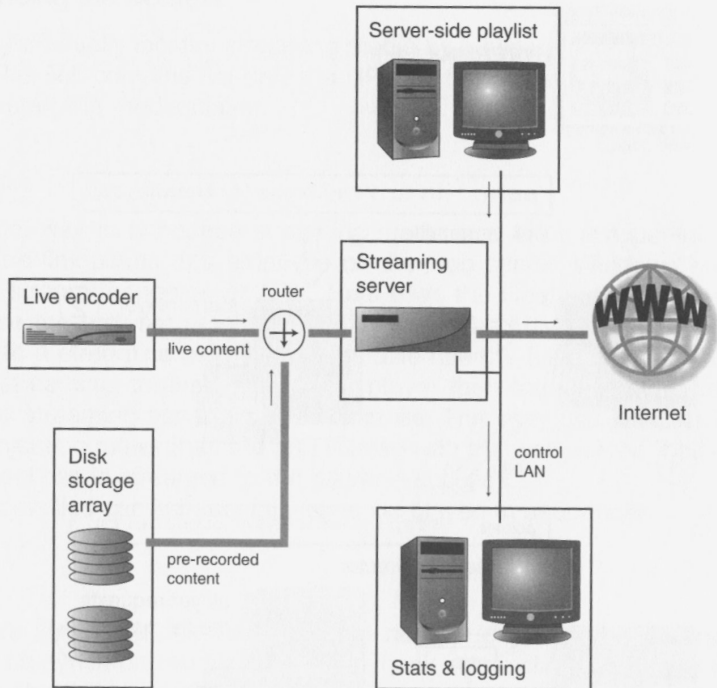


Figure 11.8 Live and simulated-live webcasting.

If only a few clients were watching at the remote sites, the streams could be delivered directly from the server at headquarters. But each client will be using bandwidth on the trunk networks, all carrying the same data. If a large number of clients at each site are watching the webcast, the server will be unable to handle the load. The alternative is to use local servers at the remote sites to receive a single stream, and then locally distribute it. This saves bandwidth on the long-distance circuits and reduces the load on the originating server.

You may be thinking that this is a really important broadcast, what happens if the link goes down? Companies like RealNetworks have come up with a number of solutions.

RealNetwork's Helix Universal Server includes the facility to set up backup links. You set up redundant transmitters, usually two servers in parallel. If the link from the designated transmitter breaks, the receiving server automatically falls back to the next available transmitting server. The clients will see the webcast has stopped; if they refresh the link, the stream will continue using the backup path.

Multicasting

If you are webcasting live to a large audience, a regular unicast establishes a separate network connection to each client. This uses considerable network resources, even though each client is watching exactly the same material. Multicasting offers a solution by serving a single stream. This is then routed to all the clients. So the network routers are doing the distribution rather than the streaming servers.

The drawback with multicasting is that you need to have control over the routers. First they have to be multicast enabled, which is not necessarily the case. There are different algorithms for setting up multicast routes: dense and sparse. If you are streaming over a corporate network within a single domain, multicasting has much to offer. It is when you want to multicast to the public Internet that problems can arise. The network of peer-to-peer connections that link a server to a client may not be multicast enabled, or the routers may not be set up to multicast across domains.

If you are multicasting you cannot use automatic rate changing. The server transmits a single stream and is not aware of the clients, so it cannot negotiate to serve at a certain rate. To get around this you may have to transmit three or four different rate streams from different server ports. The player connects to an appropriate port for the bandwidth available at their local end.

Multicast network

If you are setting up a multicast to several sites, and want to use Virtual Private Networks (VPN) for the intermediate connections across the corporate WAN, note that IPSec does not support multicasting. The way around this is to unicast to a splitter server at the remote site, then multicast locally. You will need to make sure that clients can see only one multicast server. Since the same IP address is used for the multicast, the client potentially could receive several duplicate packets. The player will not decode the stream correctly in these circumstances.

Announcing a multicast is different from retrieving on-demand content. With on-demand, the browser requests the media file. When the file is retrieved, the header carries information necessary for the player configuration. Once you join a multicast, there is no header. Instead, a small message file gives the browser/player the necessary information (like the port number to use). This message can use Session Description Protocol (SDP – RFC.2327); Microsoft uses the media station NSC file. The media station is analogous to a television station, so the station represents the channel and not the media streamed (programs transmitted) over the channel. The NSC configuration file will set up the player correctly to receive a multicast. The ASX file that announces a multicast points the player to the NSC file.

RealNetworks supports a form of multicasting with a control back-channel. This allows full statistics to be gathered from the clients, but has the advantage of multicasting the media data. It is best suited to small audiences; a very large multicast would have problems with the server capacity required to handle the unicast control traffic.

On-demand serving

On-demand serving is more like running a web server. The viewers choose their own content, and then a fast disk storage array delivers the content, as required, to the streaming servers. Each client has a unicast connection with the server, so the more viewers, the higher the server loading. A popular site will use many servers in parallel. The Internet traffic loading can be balanced across all the servers.

The server hardware does not need many facilities: a fast CPU, plenty of RAM, and at least two network-interface cards (NICs). If the server has very high loadings or there are network problems, you will need access by a separate network port for control messages, so always install at least two NICs. The system will be more reliable if the load is spread over several small servers, rather than one large multiple processor server. This also gives redundancy against hardware failure.

Inserting advertisements

If you are running a commercial site you will want to add advertisements to the content. They can be the same banner ads used on web pages. The alternative is to insert clips into the stream just like a regular television spot. This is called interstitial advertising.

The simplest way to place video advertising around an on-demand clip is to top-and-tail the content with preroll (gateway) and postroll ads (bumper) using an SMIL file (Synchronized Multimedia Integration Language) to play the clips serially. SMIL has a time container that can be programmed with a fixed sequence of media elements. The sequence command is used to place ads before and after clips. The player will run the playlist of content and advertisements as programmed. The viewer cannot step through the playlist manually to jump over the ads. An associated SMIL element, parallel, commands the group within the time container to run together. This ensures that the following clip is correctly prerolled to avoid any glitches as the playlist is running.

Windows Media Services offers two ways to deliver advertising and other interstitial material: with either a client playlist or the server-side playlist. The

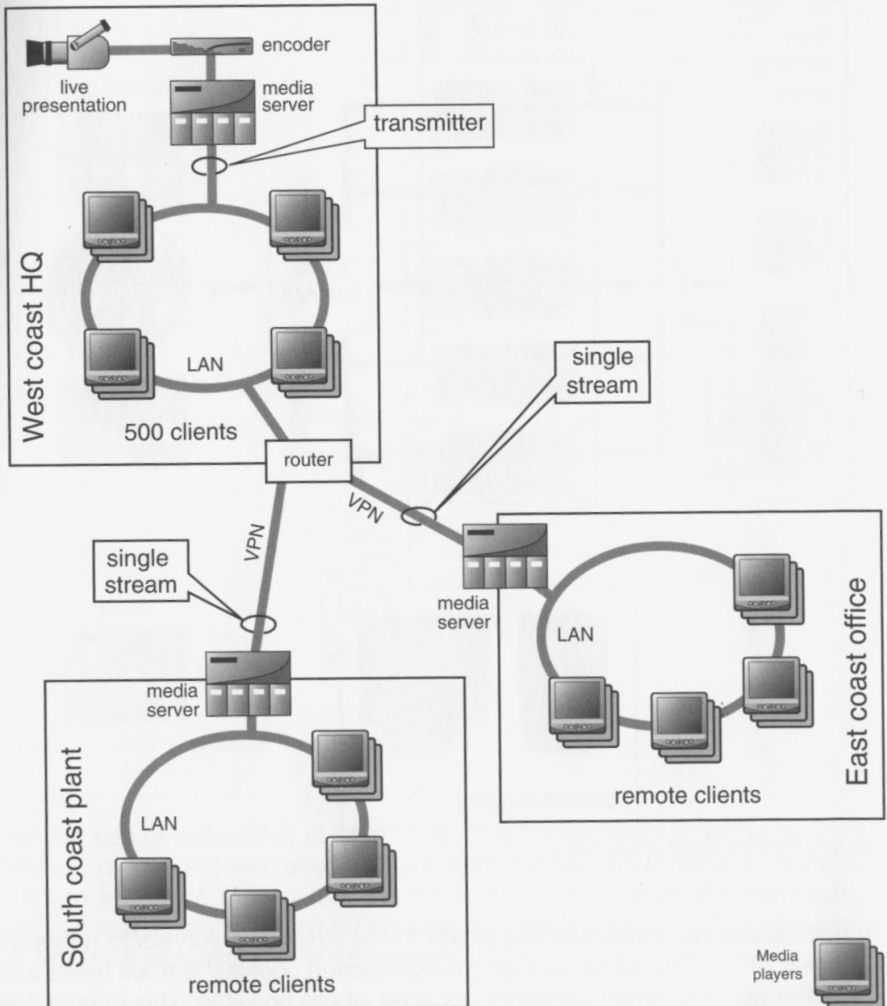


Figure 11.9 Splitting a live stream at remote sites.

client playlist cannot be changed once received, whereas the server-side allows dynamic changes to the playlist; for example, targeted ads. Windows Media use active server pages to generate dynamic playlists. RealServer can use a proprietary ad tag in the SMIL file (`<RealAdInsert/>`) to dynamically assign URLs to the advertisement clip. Since the advertisements may be changing hour to hour, this avoids the necessity to keep updating the SMIL file.

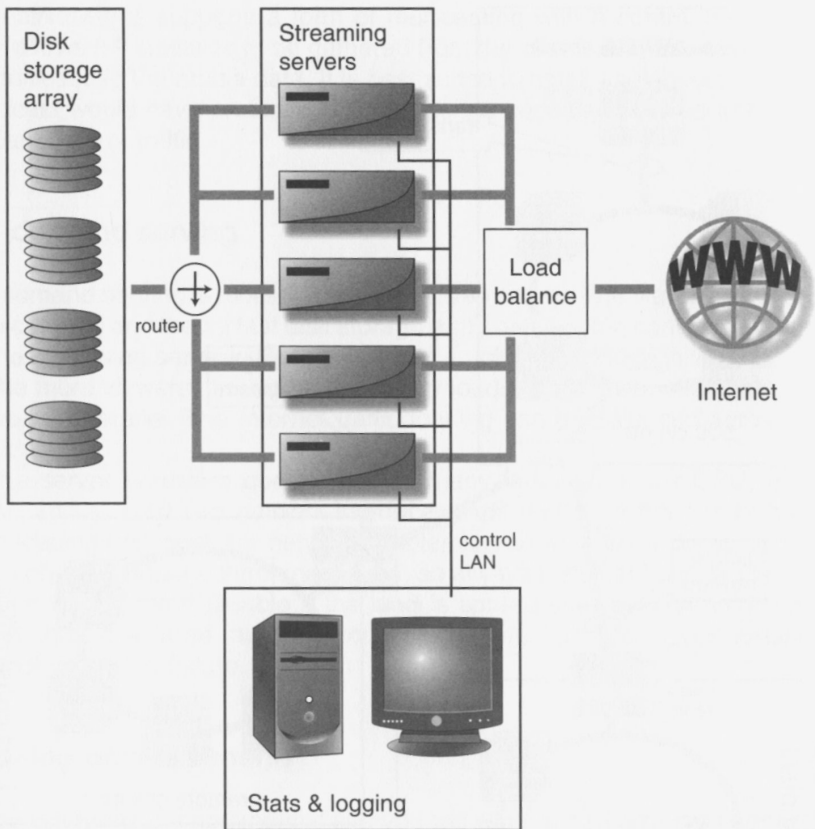


Figure 11.10 On-demand server farm.

In both cases the ad servers are linked to the streaming servers to generate the correct URLs. The ad servers often use solicited cookies to track users and ensure that targeted advertisements are used where possible. This means that a webcast, although streamed to a wide geographic area, can carry advertising that is local or relevant to the viewer.

Playlists

If you are streaming regular webcasts you will need a means of playing out clips to a prepared schedule. Just like a television station, clips can be put in playlists and streamed on cue. This means that corporate presentations and distance

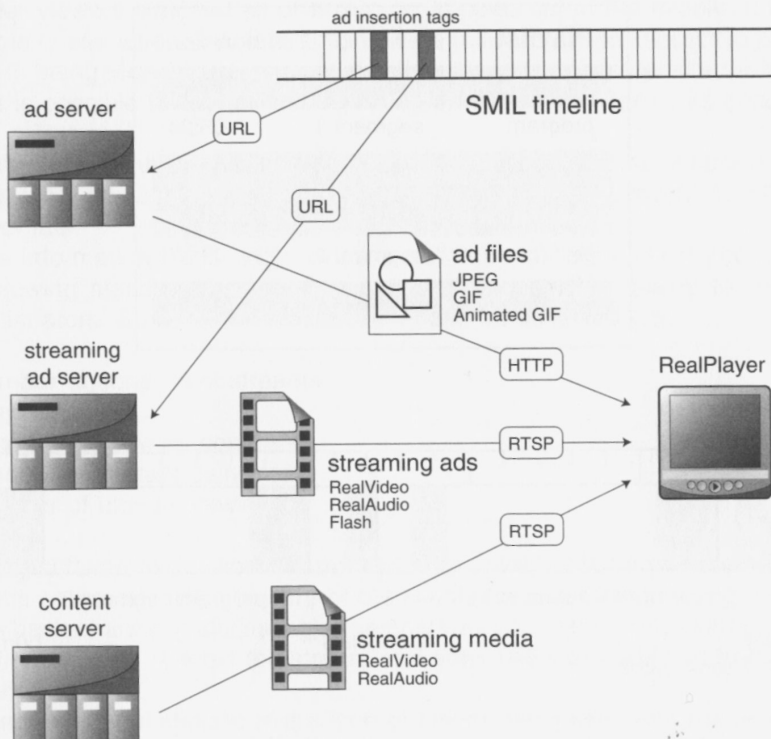


Figure 11.11 Helix Universal Server ad serving.

learning can be scheduled at fixed times. The viewer tunes in at the appropriate time to catch the required stream. Again SMIL can be used to program the playlist.

If you are using the Apple QuickTime Streaming Server, the administration terminal has a user interface for creating and editing playlists. This can be used for MPEG-4 or audio-only MP3 playlists.

Logging and statistics

So, you have set up a stream-serving infrastructure. But what is the return on the investment? With any system it is vital to monitor the traffic to the clients. This is partly to see who is watching what, and partly to find out the utilization of the servers and network elements. You will want to know who the clip's audience is, and perhaps more important, how many are watching the entire clips and how many watch only the first 10 seconds.

Playlist		duration
gateway	ident	00:30
program	segment 1	05:24
interstitial	spot 1	00:30
	spot 2	00:15
program	segment 2	09:37
bumper	closeout	00:15

Timeline

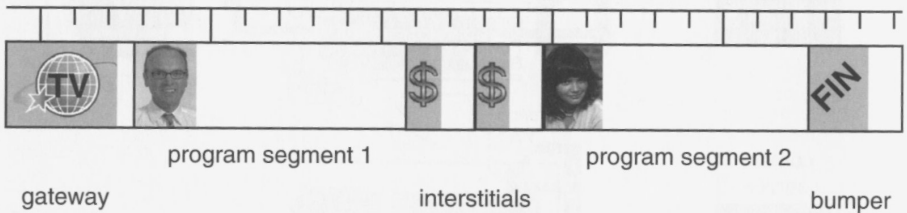


Figure 11.12 Server playlists.

The monitoring can be performed at the client or at the server. The television model is to monitor activity at the receiver, like Nielsen reporting. With streaming media, there is a back-channel from the player so server-side reporting also can be used.

Several companies have products that can collect server logs, and then process the data into reports. These include EnScaler, SyncCast, and WebTrends. Arbitron specializes in audience measurement with their MeasureCast service. They use a combination of comprehensive server logging, and complement this with client-side statistics gathered from selected streaming media users. The client-side statistics can give more demographic information than the raw server data. The two are combined to provide comprehensive statistics for the webcasters.

The servers log can include basic information about the client such as client IP address, connection protocol (RTSP, HTTP), player version, operating system and version, language setting, and the CPU speed.

The logs also collect session information. This includes the time the request was logged and how long the stream was watched. You can see from the reports

whether viewers watched all of a clip, or dropped out in the middle. It is also possible to see when user interactions were taken. So if an interactive presentation is being viewed, you can log whether the viewer has selected a hotspot link. It is possible to turn off this reporting on the players for users concerned about privacy.

More detailed information also can be collected about the transmission quality packet loss, number of resends, and how many packets arrived out of order, early or late.

This information for a single client is collated with other client access logs into viewing statistics for content producers and service quality for network administrators. Some of the statistics that can be collected are:

- Number of concurrent streams
- Number of hits
- Time spent viewing content
- Amount of content delivered
- Number of unique viewers

Logging software will collect information from unicasts, but if you are multicasting, the server will not be aware of the clients (unless you are using RealNetworks back-channel multicasting). The only way you can identify clients is when they first log in to request the stream redirector (ASX or RAM file) from a web server.

One very useful statistic is the type of player being used and the stream bit rates. This information can be used to justify (or otherwise) the number of formats to encode. You may be able to drop one of the architectures because it has a very small audience, which will save you the costs of both the encoding and serving.

A typical example of a reporting package is WebTrends. Logs from all the different servers are collected and collated at a central point. The results are stored in a database, from where any number of reports can be generated for the interested parties.

Proprietary server architectures

Windows Media Services 9 Series

Microsoft's streaming server is bundled with Windows Server 2003 as Windows Media Services. Windows Media Server supports several different delivery mechanisms in a process of rollover. This is so that the streams can be

delivered through corporate firewalls. With the release of Windows Media 9, Microsoft added support for RTSP as well as their proprietary MMS (Microsoft Media Server) protocol.

RTSP and TCP

This is the streaming mode of choice. This gives full support for real-time delivery, fast cache, and client interactivity. The fast cache allows the player to use the player buffer as a data cache to help average out network bandwidth variations (only with TCP).

RTSP and UDP

This is the fallback mode if RTSP and TCP are not supported by the player. Some network administrators set up the firewall to block UDP traffic, so this protocol cannot always be used.

HTTP and TCP

This is an alternative for firewalls that will allow regular HTTP web traffic. The Windows Media Services still provides controls like fast forward, reverse, and so on. The stream will suffer from the TCP retransmissions and rate adaptation.

The server will attempt to connect by each protocol in turn until a satisfactory dialog is established. Windows Media Services also support multicasting over IP.

Versions of Windows Media player prior to version 9 do not support RTSP, so has a further sequence of rollover starting with MMS and UDP. If UDP traffic is blocked, then HTTP will be used.

RealNetworks Helix Universal Server

RealNetworks offers the Helix Universal server for the distribution of RealVideo and RealAudio. It can also be used to stream Windows Media, QuickTime, and MPEG-4, hence the universal tag. The Helix server can be deployed on Windows and UNIX platforms (AIX, FreeBSD, Linux, Solaris, HP-UX).

The mobile version of the server adds a feature set for the delivery of 3GPP content to wireless players.

RTSP and TCP

This is used for the control connection. It gives full client-server interactivity.

RTP and UDP

This is the optimum choice for streaming the media content.

RTP and TCP

This is second choice for streaming the media content if the firewall blocks UDP data.

HTTP and TCP

This can be used for progressive downloads if no other delivery is possible. The media file is wrapped with the HTTP format, a process called HTTP cloaking.

RealServer supports SMIL for rich media presentations.

Apple QuickTime

QuickTime started as a CD-ROM format, and was developed for progressive downloads. True real-time streaming has been supported from QuickTime version 4. Apple adopted the RTP for streaming over UDP. If this is blocked by corporate firewalls the alternative is progressive download over HTTP. Apple has two solutions for streaming: one is the QuickTime 5 Streaming Server (QTSS) that is part of the Apple OS X Server, the other is the open source Darwin server.

Server deployment

Video server hardware

The streaming server has to read the media files from disk, packetize it, and deliver at the correct rate to give real-time playback. These tasks then have to be performed for multiple concurrent streams. The real-time requirement contrasts with the asynchronous operation in a regular office server application. In the latter case, at times of peak resource demand, file delivery is delayed. With a video stream this would result in stalling playback.

As the performance of server components improves with time, it is becoming easier to meet the demands of streaming.

The areas to focus upon include:

- Disk drive performance
- Network card performance; it can be advantageous to use multiple cards for streaming and control

- The system and I/O bus bandwidths
- Symmetrical multiprocessor CPUs
- The stripping out of unnecessary software services, leaving the resources for uninterrupted stream delivery
- System memory large enough to manage multiple high-speed streaming buffers

The basic configuration can be calculated from the bandwidths, but test and measurement will be required to determine the true capacity of a server configuration. To aid testing, Microsoft supplies a Load Simulator that can provide dummy loads during the server tests.

One way to get good performance is to scale wide, with many small servers, rather than one big multiprocessor server. This also makes the system more tolerant to faults; you can lose a single server from a cluster, without losing the whole site.

Once you have determined your available bandwidth, and how many users will connect to the system, you can then decide on a server design. You may want a separate staging server to test content before it is placed on the public server; alternatively, this could be a different directory.

Hosting

If you want high-performance streaming to large numbers of public clients you are going to need a fat pipe to the Internet backbone. If you do not want to install a T-3 line, the easiest way is to use a hosting facility or a content delivery network. The hosting providers usually are located at hubs of the Internet with intimate and very wide band connectivity to the backbone.

If you are running a web site, you already will have looked at the pros and cons of outsourcing the servers. Outsourcing has advantages from the point of view of physical security. The service providers usually are located in secure buildings, with standby power facilities, halon fire extinguisher systems, and multiple paths to the Internet backbone. If all the servers are remote from your corporate network, there are no issues with unauthorized access and firewall configurations. On the other hand, if your files contain confidential information you may want the servers on your own premises, managed by local staff. If you are using DRM with pre-encryption, this may not be so much of an issue.

The companies will offer a number of different services:

- Turnkey hosting
- Shared hosting
- Co-located hosting

Turnkey hosting is the simplest to implement; you upload your media files using FTP. Everything else is done for you.

Co-location gives you more control; you rent secure cages and install your own server plant. You get reliable power and connectivity. Your servers can be monitored by telnet from your own location.

Take care to study the service level agreement (SLA). Think carefully about the service you really need. Will your business suffer because of site outages? Can you work with more relaxed service reliability?

High availability

If you want to set up a system with high availability the system should be secure, reliable, and easy to maintain.

It may not be unusual for video servers to be called upon to serve up video content at any time, day or night, to anywhere on the globe. To achieve a higher level of system availability will require a fault-tolerant design. This may come in the form of redundant hardware such as power supplies, fans, or NICs, or in the form of redundant multiple server architectures. To minimize downtime, hot-pluggable components such as disk drives, power supplies, and fans are also desirable.

Security

Don't forget security when planning your systems. If you are streaming around the clock, and it becomes a core business service, then server outages could impact your business.

The security of your streaming provision is determined by many factors. There are the physical threats, like fire and theft. Redundant, mirrored server sites are a good way to deal with these issues. Then there are power outages – do you need uninterruptible power supplies?

If you are using a reputable hosting service, then most of these issues will be covered, but check the service level agreements.

The other threats come from hackers. If they can gain access to your servers they can wreak havoc. There are also the denial of service attacks. The best route is to call in a network security consultant to advise and audit your systems.

Authentication and authorization

Access to the origin server has to be restricted to authorized users. There are three types of users: the system administrators, the content creators uploading content, and the viewers of the content.

Authentication is used to verify the identity of a client. You may not want to know, in which case anonymous authentication can be used where no user name or password is required. Authorization then allows authenticated clients access to confidential information. A database can store lists of authorized users in the three categories. The users gain access by password.

If your media is of commercial value, or confidential, then some form of digital rights management provides much greater protection against unauthorized access to your content.

Summary

The streaming server is a type of content server that uses a special software application to deliver streaming media in real-time to the players. It differs from a normal web server in having constant flow control, rather than the rate-adaptive transmissions of TCP.

There are two types of transmission: live and on-demand. On-demand gives the interactivity that sets rich media applications apart from traditional video delivery systems. The user interaction is via an out-of-band bidirectional control channel.

The control channel can also control the streaming delivery rate to react to network congestion. By encoding material at several different rates, the server can select the appropriate stream for the prevalent conditions.

Streaming is usually over a one-to-one connection between the server and the client, called a unicast. The alternative option for a webcast is the multicast. The server transmits just one stream, and any number of clients can connect to the stream. This potentially saves network utilization, but has yet to be deployed universally across the Internet.

One of the most important aspects of managing a streaming system is the server logging. This is where you can measure the audience, and establish the return on your investment. It also provides vital information for the network administrators to tune the systems and identify possible bottlenecks.

The streaming server is an effective replacement for other means of delivering multimedia content that can offer lower costs, immediacy, and interactivity.