Streaming Video Infrastructure and Services
By Peter Sutton

INTRODUCTION

Online video streaming is an attractive option for providing greater access to digital materials in archives, libraries, and museums. Streaming offers new and exciting ways for institutions to interact with their primary user base and creates points of discovery for new users located anywhere in the world. It enables institutions to enter their moving image and audio collections into a world where online access is paramount, a world in which content that is not online does not exist in the consciousness of many users. New tools and support for embedding and streaming video, whether born-digital or digitized, have made this process easier than ever. There are, however, a number of important factors that content providers must consider for successful streaming projects. Whether outsourcing or performing the service in-house, understanding what goes into streaming enables institutions to have more control over their digital video materials and serve their users better. This paper outlines the basic concepts, options, and issues institutions must consider to stream media online.

STREAMING

At its most basic, streaming refers to moving image and audio content transmitted from a content provider to an end-user. This media content can be transmitted in many ways, but this paper focuses on content transmitted through the Internet, private networks, and wireless networks. Such streaming requires a server that encodes and serves the moving image and audio materials through the Internet to an end user's device (e.g., computer, smart phone, tablet) with an application that decodes the transmitted content (e.g., web browser with media player) also connected to the Internet.

Such Internet streaming can occur over the World Wide Web (series of web pages and contents), wireless networks, and private networks. It can be live (webcasting), such as television broadcasts, or can offer a library of static content (video-on-demand). Users can access streamed digital video content using a number of devices and processes.

In order to stream video,

- Media obtained
- Media prepared
- Content uploaded or made available online
- Content delivered to user
- Content successfully viewed by user

True streaming and progressive download are different, although both use the Internet.

Streaming,[1]

- Generally used for live or copyrighted material
- Uses different protocols for different needs

[1] "Streaming Media" *Wikipedia,* Last modified December 13, 2013, http://en.wikipedia.org/wiki/Streaming_media

- Content not stored in hard drive, but sent in packets directly to decoding device in ways that protect digital assets
- Requires more work on the side of the content delivery network or provider
- Limited amount of file is downloaded while playing

Progressive Download,[2]

- Sent using HTTP with TCP/IP, not adaptive or multiple bit rate as a result
- Relies on header metadata and drive storage
- Whole file is downloaded while playing
- File downloaded to a physical drive on device of end user in temporary folder of decoding application

True streaming usually involves a live event, like a lecture or sporting event, or video-on-demand content that needs to be encrypted for copyright or other reasons. It requires more work and different protocols than progressive download because part of the file is not stored on a local disk but cached in the media players or browser and discarded after playback. Progressive download is used exclusively for stored content and is always delivered using Hypertext Transport Protocol (HTTP) and Transmission Control Protocol (TCP) over Internet Protocol (IP).[3]

Streaming and progressive download both segment the overall digital video file. Streaming uses the segments or packets for speed, caching, and adaptive bit rate streaming. Progressive downloads use them for searching in the content, with each segment triggering a new download. As a result progressive downloads are sent using standard web servers as opposed to streaming, which uses specialized media or video servers.

With truly streamed or live content, it is important that when a few data packets are lost during transmission, redelivery is not attempted automatically. TCP over IP automatically asks for lost packets based upon a lack of receipt transmitted to server of their delivery. Not having a data packet during video playback creates a minor compromise in video and file integrity from the lost data, but is not necessarily discernible to the user during playback. In addition, bandwidth that would be used for automatic retransmission of lost data packets that are useless to the streaming video would be more efficiently spent on the incoming useful data for playback. This is why true streaming content is delivered using the User Datagram Protocol (UDP). Progressive download is a normal video file download that is being in parallel while it is downloaded. Normal file download uses TCP and its ability to ensure whole file delivery that. Progressive download requires less exact timing, so using TCP is acceptable even if

---

[2] "Progressive Download" *Wikipedia,* Last modified December 9, 2013, http://en.wikipedia.org/wiki/Progressive_download

[3] Ozer, Jan, "Streaming Vs. Progressive Download Vs. Adaptive Streaming," *Information Today, Inc,* 2013, Print, See Ozer for a detailed discussion of progressive download vs. streaming.

it means some user delays in playback. UDP provides more options and a better workflow for streaming video.

Providing content through progressive download is a much simpler process of storing media files in temporary folders on the end users receiving device. This compromises security and can easily be ripped or copied to the users device, but provides an effective service with much less work on the provider's end.

Adaptive Bit-Rate Streaming combines streaming and progressive download.

- HTTP delivered by TCP/IP
- Videos encoded at different bit-rates and segmented
- Dynamically adapts segments and video quality based on a number of factors including available bandwidth and network congestion
- MPEG-DASH is an ISO standard

Adaptive Bit-Rate Streaming or ABR is the next step in the evolution of streaming technology. It combines both progressive download and true streaming techniques to deliver high quality digital video that adapts to a number of factors including available bandwidth and network congestion over standard web servers and existing Internet infrastructure.[4] ABR can be used for both video-on-demand and live events. Video is encoded at multiple bit rates and broken into segments to form one media stream. When a user is watching a video through ABR, different segments at varying bit-rates may be shown as the system adapts to changes in the Internet ecosystem. ABR may not provide the highest quality video at all times, but the content is seen with lag minimized. ABR combines existing network intelligence and infrastructure like content delivery networks (CDN) with a scalable and interoperable framework to provide a high quality of service for online video delivery.

One of the readily available implementations of ABR is the proprietary MPEG-DASH (Dynamic Adaptive Streaming over HTTP). It is an ISO standard and provides the best overall conceptual model and framework for ABR of all the proprietary implementations. There are a number of profiles to DASH that lend to personalization so institutions and organizations can implement as their needs dictate. DASH specifications do not provide a full end to end (client to user) service, but give the building blocks and framework to create one.[5]

Streaming workflow

Successful streaming, whether using ABR or other methods, requires safeguards and workflows that maximize service to users and minimize the problems that can occur at various stages throughout the process. Keeping synchronization information intact is fundamental. Content providers also have to take into account user knowledge and the

---

[4] Sodagar, Iraj, and Thomas Stockhammer, "MPEG DASH: The Enabler Standard for Video Delivery Over the Internet," *SMPTE Motion Imaging Journal* 121, no. 5 (2012): 40-46. See article for more detail about ABR processes.
[5] Sodagar and Stockhammer "MPEG DASH: The Enabler Standard", 40-46.

user's quality of service. Low or variable data rates from user Internet connections, error rates, and delay in delivery of content are also important.[6]

There are five main components to any streaming architecture or workflow,

- Capture of content
- Preparing video for streaming
- Servers
- Distribution and delivery
- Web browser or decoder

Prior to navigating these components, it is also helpful to develop an understanding of basic Internet architecture.

BASIC INTERNET ARCHITECTURE

Overview of Section

- TCP/IP stack layers and principles
- Network transmission
- Encapsulation and Packetization
- Basic Principles

IP Stack

An understanding of network and Internet architecture is fundamental to streaming. Most Internet communications occur using Transmission Control Protocol (TCP) over Internet Protocol (IP), employing several network layers stacked on top of each other with the number of layers depending on the model. The TCP/IP model has an application layer at the top, followed by transport, network, data link, and physical layers, in descending order. There are other models for Internet architecture[7] that are more detailed and have different types and numbers of layers, but the TCP/IP model is stable, scalable, and useful for understanding the basic principles at work.

IP Stack Layers

Each layer in the TCP/IP stack has an interface with the layer above and below it, adding services to what it receives from below and passing them on to the next layer. Services refer to communication between layers within the same process, while protocols refer to

---

[6] Inglis, Andrew F., and Arch C. Luther, *Video Engineering*. 3rd ed., New York: McGraw-Hill, 1999, Print, Page 504.

[7] Austerberry, David, *The Technology of Video & Audio Streaming* 2nd ed., New York: Elsevier, 2005, Print. Page 15. Beaumont, Leland R. and Markus Hofmann, *Content Networking: Architecture, Protocols, and Practice,* San Francisco: Morgan Kaufmann Publishers, 2005, Print, Page 25. International Standards Organization (ISO) and the Open Systems Interconnection (OSI) model details seven rigid network layers, U.S. Department of Defense, DoD, DARPA model details five layers.

communication between processes.[8] The physical layer holds the functions necessary to transmit the bit stream of content over a physical medium to another system, for instance an Ethernet cable. The data link layer contains the details of interfacing with the physical communication medium or network layer. It organizes bits into data units called frames and delivers them to an adjacent system. The network layer forwards data packets across as many links as necessary, ensuring proper reconstruction by adding a header that contains destination information.[9] The network layer forwards these encapsulated data packets to and from endpoints (host and end systems) and routers (connectors between two computer networks). The application layer involves implementation of the application software including HTTP and File Transfer Protocol (FTP).[10]

Transport Layer Protocols

The transport layer coordinates data exchange between both end points (host and end systems) and can add value to services provided by the network layer using TCP or User Datagram Protocol (UDP). TCP sends TCP segments and UDP sends UDP datagrams to individual applications by arranging a connection between applications on remote computers. The main difference between the two protocols is that TCP requests a missing data packet from a missing receipt of that packet's arrival at the end system, while UDP datagrams do not require a receipt and can intelligently request missing packets only as necessary.[11] TCP while more reliable is not suitable for many streaming video applications since the single biggest issue for users is timely reception. At most a lost packet will result in a lost frame, hardly noticeable to most users and time is spent in retrieving it as well. UDP with its ability to send and receive information without receipts or retransmission is better aligned for streaming video uses.[12] TCP automatically requests packets even when they will not be useful for playback in streaming video, when the video has already passed that point in time. UDP can be configured to request only packets that will be used in playback of the video stream.

Network Transmission

A system connected by the Internet generally can be mapped to two different groupings, host or end systems (ES) and intermediate node (i.e. router) or intermediate systems (IS). Every connection uses the bottom three layers, physical, data link, and network, which

---

8Goralski, Walter, *The Illustrated Network: How TCP/IP Works in a Modern Network,* Burlington: Morgan Kaufmann Publishers, 2009, Print. See Chapter 1 Protocols and Layers in Kaufmann for more detailed discussion of Internet stacks and systems.

9 Dostálek, Libor and Alena Kabelová, *Understanding TCP/IP: A Clear and Comprehensive Guide to TCP/IP Protocols,* Birmingham: Packt Publishing, 2006, Print, Section 1.2.1.

10 See in particular Dostálek and Kabelová. *Understanding TCP/IP: A Clear* Section 1.2.1, Goralski *The Illustrated Network: How TCP/IP* Chapter 1, Inglis, Luther, *Video Engineering*. 3rd ed.,Pages 504-10 and Austerberry *The Technology of Video* Page 16, Beaumont and Hofmann. *Content Networking: Architecture, Protocols* page 25 for more information about basic Internet architecture and communication.

11 Dostálek and Kabelová, *Understanding TCP/IP: A Clear,* Section 1.2.2

12 Austerberry, *The Technology of Video,* pages 15 – 19.

are collectively called the network support layers. While host or end systems may use other layers as well, intermediate systems only need these three since they are not processing the data transmitted, just sending it from a host system to an end system.

Packetization and Encapsulation
Internet communication is predicated on encapsulation and packetization.[13] It relies on breaking down information, placing these segments in different containers to send, and then unpacking these containers at the receiving end. Starting with the host system, each segment or packet passes through the various stack layers starting at the application level, with each layer adding information for the next. For instance, the network layer adds a header containing information to ensure correct transmission, and the data link layer adds a header and a trailer. After passing through the five layers of the host system, the packet is known as the protocol data unit (PDU). As the PDU is processed by the end system, each layer unpacks everything that was done by the host system so it can be processed by the application. Such packets serve as the basic units of data of Internet communication.[14]

Basic Principles of the Internet
There are some basic principles associated with the TP/IP model and stacks including end-to-end, robustness, and network transparency.[15] The end-to-end principle is a concept that has evolved to reflect changes to the Internet. Originally it meant that the functions needing services were the shared responsibility of the connected ends and hosts; support was focused on the edges (ends) of the network. Over time, services such as firewalls and web content caches demanded that the network or intermediary nodes also be supported. The robustness principle means that the host should only send out a well-formed packet or datagram, although it should accept any packet it can interpret even if it is not well formed. Network transparency asks that the packets arrive in the same order they were sent, aided by a single universal addressing scheme and without transport modification.

PREPARING VIDEO FOR STREAMING

High quality streaming of digital material begins with compression of the material to be streamed. During this preparatory stage content providers also need to make key decisions involving the way the video will be streamed. These include what mechanism(s) will stream the video (players in HTML, YouTube, Google Video), how many files to encode or compress from the original file, and what format and codec, among others.

---

[13] Dostálek and Kabelová, *Understanding TCP/IP: A Clear* Chapter 1, Goralski *The Illustrated Network: How TCP/IP* Chapter 1, Inglis, Luther. *Video Engineering*. 3rd ed., Page 504 Beaumont and Hofmann, *Content Networking: Architecture, Protocols,* page 25.
[14] Dostálek and Kabelová, *Understanding TCP/IP: A Clear,* Section 1.3.2
[15] Hofmann, Markus, and Leland Beaumont, *Content Networking: Architecture, Protocols*, pages 28-29 Beaumont and Hofmann, *Content Networking: Architecture, Protocols*, page 12.

Compression

Video comes from a camera in its raw or uncompressed form, and most digital video files from analog sources are kept in their uncompressed form for preservation. Uncompressed born-digital videos as well as analog sources digitized for preservation are too large to be streamed without losing quality of service. The file sizes need to be reduced or compressed to a size suitable to video streaming. This involves removing the number of bits that represent content while retaining enough to be usable and faithful to the original. Digital video consists of pixels that comprise picture data, which changes from one complete field and frame of video to the next. In uncompressed form, digital video has a number of redundant or unnecessary pixels and bits that can be removed or encoded to smaller size without losing too much picture quality.

Codecs and File Formats

A codec (H.264, DV, MPEG-2) provides a way to encode and compress the bits of digital content. The encoded content is then placed in a container or wrapped in a specific format (.mov, .mp4, .avi) for storage. For video there are two parts to the content, audio and video. This paper focuses on video encoding and codecs, but it is important that whatever streaming implementation is chosen takes into account both parts of the content and they are synchronized for proper playback.

Codec and format selection will vary by the way video is streamed, but should offer a high quality picture while significantly reducing the file size for low bandwidth requirements. No true baseline codec is universally supported by all browsers, web players, streaming services, or HTML for video streaming. There is a de facto standard of H.264 MPEG 4 Part 10 wrapped in a number of containers and paired with Advanced Audio Codec (AAC) for the audio part. Other systems exist, however, and have their advocates.

H.264 meant encodes videos with high compression while retaining high quality suited to streaming. It is, however, a proprietary format demanding royalties from groups and users. In contrast, some web developers want an open-source royalty-free system and have tried to set the baseline codec as Ogg Theora, which is natively supported in HTML5. WebM is another widely utilized container format for web video. Launched by Google in 2010, it is open-source but does not have wide support in a number of heavily utilized browsers. Due to a large support network and despite the inherent royalties, H.264 remains the most widely utilized codec for streaming video.

History of Digital Video Codecs

The multiple codecs now in existence reflect 30 years of developments in web video formats and encoding.[16] Modern digital video encoding began in 1984 with H.120 from the International Telecommunications Union (ITU) – Telecommunication Standardization Sector. For web video, groups of pixels were coded together to create a certain bitrate for transmission. This had to be done due to limitations on compression and the speed of Internet connections. The product, while standardized, was not fully

---

[16] Austerberry, *The Technology of Video,* Chapter 4 and Real Team, "The History of Digital Video File Formats," *Real Networks, Inc,* Last modified April 12, 2012, http://www.real.com/resources/digital-video-file-formats/

adequate for web video. In 1988 H.261 was rolled out by the ITU. It became the first major digital video compression standard and formed the basis of most subsequent video standards and codecs. H.261 improved on H.120 to create a product that could actually be streamed.

In 1991 the Motion Pictures Experts Group premiered MPEG-1. It was based on creating digital video of the quality of analog video formats such as VHS and then compressing them down to certain bitrates. MPEG-1 also created strict rules for bitstreaming and decoding of video files. The compressing and encoding was left open-ended allowing for developers to create multiple encoders of different standards for MPEG-1. H.262 or MPEG-2 was a joint effort between MPEG and ITU. MPEG-2 offered more options for higher resolutions and audio among other innovations. It constituted a significant improvement in video quality and served as the basis for DVD and digital TV.

After MPEG-2, streaming video began to take off with the 1997 introduction of the Real Media container for files based on H.263 and its Real Video player whose express purpose was to stream video. In 1998 MPEG-4, H.264, and Microsoft's VC-1 were introduced and became standards for HDTV and Blu-Ray. These offered improved features in resolution and digital rights management to provide a massive increase in quality. MPEG-4 Part 2 spawned video formats like DivX and Xvid. Various container formats such as VOB (DVD Video Object) for DVDs and Advanced Systems Format (ASF) for web streaming were also introduced. In 1999 streaming video formats and support was offered in Windows Media Player and QuickTime.

In 2001 a number of different container formats were premiered including the open-source Ogg for video and the Theora codec, QuickTime .mov, and 3GP for wireless handheld devices. Ogg Theora is the open-source royalty-free choice of HTML5. In 2010 Google using their VP8 and VP9 codecs for video and the Vorbis codec from Ogg for audio introduced WebM. Still the only codecs that receive universal support for streaming video in widely adopted browsers are H.264 for video and AAC for audio.

Potential Changes for Web Video Codecs and Formats

If it were open-source and royalty-free H.264 would be the ideal baseline codec. It provides high quality compression with near universal support and is a broadcast standard. H.264 has the benefit of being related and in the family of other forms of distribution like HDTV and Blu-Ray; it represents what people expect to see with digital video. Ogg Theora would have to be more widely implemented and supported before it is a true baseline codec. WebM is an exciting format and includes implementations and support from services such as YouTube, but does not wrap enough codecs, for it to be widely adopted. For the open-source royalty-free codec and wrappers to gain ground on H.264 and AAC, they need to be adopted by more proprietary sources and more browsers. This is conceivable, but in the meantime H.264 will remain the way to reach the most people with a single encoding standard. It will take a service like YouTube and Google demanding that only Ogg Theora and Vorbis are supported for their videos for H.264 to lose market penetration. Regardless of whether they become royalty-free and open source, web video formats will also increase in terms of quality of compression and smaller sizes while retaining certain quality markers, as will other digital video transmission methods.

WEB PAGE CONTENT ENCODING, DELIVERY, AND DECODING

Overview of Section

- HTML Principles
- HTML Before HTML5
- HTML5
- Other Internet Languages (XML, SMIL, CSS3, JavaScript, SVG)

Introduction

Once compressed and uploaded to the content provider's computer, the video content that is to be streamed must be encoded, delivered, and decoded over the Internet. Most formatted webpages are transmitted and understood by a browser using HTTP or Hypertext Transport Protocol using TCP/IP. HTTP is an "application level communication between the web browser and web server in distributed, collaborative hypermedia information systems,"[17] and a delivery mechanism for HTML or Hypertext Markup Language. This communication is based on a request-response from a server that stores the desired information. The server receives the request and forwards a response with the desired content.[18] HTTP and HTML before HTML5 were useful for most web content including images and text, but not large true streaming video or animated content. Since progressive download is halfway between streaming content and traditional file download, it was possible to use older versions of HTML using normal web servers with embedded proprietary players. HTML5, which was fully deployed in 2007, added the <video> and <audio> elements that allow for native video and audio support in HTML. Other basic programming languages besides HTML that help with video streaming sent by HTTP are CSS3 and JavaScript.

HTML Principles

HTML formats, organizes, and links text, graphics, and other information into documents. It sends messages that have a header for information about the content and a body for actual content separated by tags. HTTP delivers the HTML message and is an application-level stateless generic protocol for communication between various parts of Internet systems. "Stateless" in this case means that all of its commands happen in isolation and are not related to any previous commands in a given communication session. Adding cookies that store state information decreases the burden on servers of maintaining connection to the browser and eases the burden of concurrent user requests.

HTML Before HTML5

Before HTML5, neither the hypertext neither language nor transport protocol of HTML was suitable for streaming video services. Progressive downloads and true streaming

---

[17] Austerberry, *The Technology of Video,* Page 44. Other definitions in Dostálek and Kabelová, *Understanding TCP/IP: A Clear,* and Goralski, *The Illustrated Network: How TCP/IP*.

[18] Weinsten, Stephen, *The Multimedia Internet*, ed. Jack Keil Wolf, New York: Springer Science+Business Media, Inc., 2005. Print, Page 252.

often involved embedded video players on web pages that used HTML for other web content. Before HTML5, even dynamic HTML that allowed for content to be changed after it loaded and was used for hidden menus that displayed when the mouse moved over them or other objects along a programmed path could not fully support video on the Internet. Streaming media and other images were often referenced by links to specific video players. There were two web players (Flash and Shockwave) that could add animated or video in various forms to HTML but both required plug-ins to the web browser. Users are generally reluctant to install plug-ins and this created an additional barrier in delivering content to users. Flash is an animated vector-graphic format that supports small file sizes, user interaction, and short to medium animation. Shockwave is for longer content, including flash content.

HTML5

Adding video and audio elements to HTML was first mentioned in 2005, followed by a trial run in February 2007 and full deployment in November 2007 from the working group WHATWG (Web Hypertext Applications Technology Working Group). Some web browser companies, such as Microsoft, did not fully support HTML5 until 2010.[19] The video element is simple and allows video to be integrated in all layers of web applications including DOM, CSS, and JavaScript; plug-ins is not required. This corrected the problems of native video in HTML and browsers and allowed streaming video to be served over basic web servers. The issues discussed in the previous section concerning the lack of a baseline codec that is supported and implemented by all browser vendors remain, but HTML5 greatly increases the capability of video to be streamed over networks. Its video element has features that connect to when video is played, how it is presented, and other key aspects of the process. HTML5 reflects the growing demand, need, and importance of streaming video.

Other Internet Languages

Several other Internet languages such as extensible markup language (XML), Synchronized Multimedia Integration Language (SMIL), scalable vector graphics (SVG), CSS3, and JavaScript also have applications with streaming video. XML is used for online natural video, audio and other data content that are delivered using HTML standards but with more flexibility for content. While HTML is set in stone, XML is much more fluid. SMIL is an XML markup language used to describe multimedia presentations. It is used for HD DVD, mobile devices, and Hulu for web streaming. SMIL is generally used with XHTML, an XML based version of HTML. SVG and CSS3 both offer styling options for video in HTML5 and other HTML including the media playback look. SVG is particularly useful for phones and other mobile devices. JavaScript offers a robust interface for media elements, can set different values, monitor statistics, and control playback functions, as well as set and read attribute values.[20]

---

[19] Pfeiffer, Silvia, *The Definitive Guide to HTML5 Video,* New York: Springer Science+Business Media, Inc., 2010, Print, Chapter 1 Introduction.

[20] Pfeiffer, Silvia, *The Definitive Guide to HTML5.* Austerberry, *The Technology of Video*.

BASIC SERVER INFRASTRUCTURE AND COMMUNICATION

Basic communication of web pages and online content happens over web servers. Servers are designed to handle multiple concurrent connections through a variety of mechanisms. Servers can be as simple as a personal computer attached to a network or as complex as a large-scale collection of web servers known as a server farm. Hosting is the process of using servers to make content available over the Internet. Hosting services providers make available servers that deliver content for content providers.

The model of communication for requesting, sending and receiving data like web pages and their content between a server and user mimics general internet communication; messages pass down through the IP stack from a host system, through intermediate systems like routers, and up through the IP stack of the end system. In addition to a host processer there is also sometimes a separate data storage unit. Data storage is beyond the scope of this paper, other than to recommend that content providers select a digital repository suitable to their needs. Options include RAID arrays, hard disks, cloud storage, or a single dedicated computer, among others. A user asks for a resource using the Uniform Resource Locator, the server locates the specific file, reads the file, and then transmits it in HTML using HTTP. The content is delivered through packets that flow from the host through the network that arrives in any order but are then reassembled in proper order by the end user.

Servers are located through the domain name system that gives all content hosts a user-friendly name and a numeric IP address. The system translates from the name to the IP address. There is also a readily available table of names and IP addresses for many hosts, organized in a tree-structure starting from top level domain names (Internet Listing Displays) and general domain names (.edu, .com, .net), followed by domain name space and zones. This allows hosts and their content to be located more quickly for a variety of end users and requests.

Security

Authentications, state information, secure web communications and other functions provide security for a variety of transactions including video streaming. User authentication provides access control by requesting a user ID that is used to verify that that specific user is allowed to access content. Netflix, for example, employs user authentication; only once paying users are verified can video be streamed to their computers. Cookies enable state information of a web session to be stored. They are useful for tracking as well as security and user authentication. SSL (Secure Sockets Layer) and TSL (Transport Layer Security) are also used for secure web communication. These methods use certificates, data encryption, server authentication message integrity, and client authentication over a TCP/IP connection.

Web Proxies

Proxy servers are intermediaries for requests from clients seeking resources. The proxy server analyzes a particular client's request to provide better connection to other servers and therefore better service to the client. Proxy servers assist with server load balancing, security, caching, acceptable use policies, compression, and other functions useful in network communications. There are many types of web proxies, each performing specific functions to improve communication and transmission of web content in ways that lead to more individualized network strategies for organizations.

Specialized Servers

There can be problems, including lag and security, in streaming certain types of content, including video. In these cases specialized servers are created to streamline and improve delivery of this specific content. For media, such servers offer more control of stream delivery rates beyond the standard buffer overflow in TCP/IP stacks. Specialized servers should be used for large-scale projects, ones that require more quality control, or when normal HTTP over TCP/IP is not satisfying users.

VIDEO AND STREAMING SERVERS

Types of Video Streaming Connections

- Multicast
- Unicast
- Anycast

There are three types of video streaming communications, multicast, unicast, and anycast. Multicast sends a single stream to the Internet to which a range of individual clients attaches for content. Anycast[21] is not particularly relevant for this paper. It follows a more traditional broadcast method where one central and best transmission point serves most users, responding to requests. Television and radio broadcast is such a case, where there is a single point of transmission that can be used and decoded with anyone with a compatible receiver. Unicast provides a one-to-one connection between the media player of the client and the content provider. Each of these types requires different connections and server functions.

Unicast

Unicast in streaming communities is a direct and open connection to a single user. It is used for a limited number of users, private content, or unique resources. If a large user base were served using unicast, too much bandwidth and computing processing would be used for the servers to handle, and the stream would be unsuccessful.

Multicast

Multicast makes one webpage live for a large audience through one stream that is served and routed to all clients. This is similar to a broadcast model of distribution where the same data is sent to all possible destinations, but differs in that the content is served only to interested destinations. There are no headers in the HTTP, just small message files with session descriptions.

Splitting and Relays

[21] Hofmann, Markus, and Leland Beaumont, *Content Networking: Architecture, Protocols,* page 136.

Media and video servers can handle simultaneous clients through relay or splitting. Both methods intersect with the path of the content or stream so it can be sent to multiple users quickly and accurately. Relay servers are designed for large broadcasts and can quickly deliver information from the origin server to clients on many different physical machines. Splitting servers connect to an origin server then make copies of the content to send to multiple client connections.[22]

On Demand Serving

On demand serving involves non-live content being stored and served with VCR-like functionality, including pause, rewind, stop, and fast-forward options among others. Live or simulated live streaming does not have such functionality as information is sent as it is created. The on demand servers function more like typical web servers where a client has unicast connections with a server that retrieves the data from storage and serves it. The basic setup is a fast computer with plenty of RAM and at least two network-interface cards (NIC). Due to high loadings or a lot content that is served to users, on demand servers need their own network port for control messages for them. This means that their communication between server and end user is separated from general web traffic, and all information (control messages) pertaining to their transmission, reception, and any changes along the way is separated as well. It is best to have several streaming servers engaged in the process to provide redundancy and reliability.[23]

Logging and Statistics

Statistics are important to understand quality of service, users, and what can be done to improve the effectiveness of a streaming implementation. The user's media player sends statistics through a back channel for server-side reporting. Logs from all the different servers connected to a particular streaming process are also collated at a central point and stored in a database where reports and statistics are generated. Some of the statistics standardly charted for streaming services include number of concurrent steams, number of hits, time spent on viewing content, amount of content delivered, types of media players used, and bit rates. Attention is also paid to time to live (TTL) or how long a data packet is useful; round-trip time (RTT) also called round-trip delay (RTD) or ping time; and how long it takes for a data packet to be sent and received.

Server Deployment

A streaming server should read media files from storage, packetize them, and deliver at the correct rate with synchronization information for real-time playback, all for multiple concurrent streams. Such servers should be evaluated in terms of disk drive performance, net card performance, system and input/output devices bus (theoretical amount of data

---

[22] "Splitting a Stream," *Microsoft,* Accessed December 16, 2013, http://msdn.microsoft.com/en-us/library/windows/desktop/dd893272%28v=vs.85%29.aspx See for a more detailed discussion of splitting a stream with examples.

[23] Austerberry, *The Technology of Video,* page 222 for more detail of possible network setups and Hofmann, Markus, and Leland Beaumont. *Content Networking: Architecture, Protocols,* pages 49–52 for different Internet layer support for multicasting.

transferred in a given amount of time along a designated data pathway named after the vehicle that transports people) bandwidth, and whether they have enough system memory to manage multiple high speed streaming buffers. The servers should be scaled wide with multiple small servers to create better tolerance to potential faults. The available bandwidth with number of users should be used to design and optimize servers.[24]

Servers and storage can be hosted in-house if the available bandwidth and Internet connection is sufficient. Alternatively, they can be outsourced to a hosting facility or content delivery network (CDN, an intelligent overlay to the internet), an option that usually has the benefit of physical security, multiple services, and a variety of ways to upload or transfer content. The hosting facilities and CDNs should have the benefit of a dedicated T-1 or T-3 line and the revenue to invest in needed architecture. Make sure to research and understand user and institutional needs in comparison to services offered when out-sourcing.

Server Allocations

The exact intelligence and mechanisms that go into server allocations are discussed in more detail under content delivery network solutions. This section discusses the way content should be served and the basic structure of servers. In an ideal situation, the host end should have a small number of video or media servers that send content to general web or more specific streaming video and media servers, which pushing out data packets to requesting users. Web servers are more prevalent than media and video servers, although this is shifting with the growing demand for online video. Media and video servers are specialized to allow true streaming. There are nevertheless more options and greater infrastructure for web servers, which can accomplish somewhat the same goals through adaptive bit rate (ABR) streaming or progressive download.

Throughout the streaming process, one wants to make sure that the servers and transmission network function is a cohesive manner like a server farm. Having a distributed and varied pathway with more options and devices creates more fault tolerance; if one point on the path fails, there are other parts that can respond in a timely manner. Servers and points of transmission should not be responsible for the stream of the whole file, but only a portion. This speeds up delivery and limits bandwidth use for any one server and helps create more fault tolerance as well.

Server Load Balancing

There are many ways to achieve a balanced server load in each such cluster of servers. Three broad classes of actions help balance server loads, determine the best available server for incoming new sessions, insure persistence of working connections between one client and one server, and filter clients to servers with specialized processes.[25] The

---

[24] Austerberry, *The Technology of Video,* Chapter 13 has a more detailed discussion of server deployment. Hofmann, Markus, and Leland Beaumont *Content Networking: Architecture, Protocols,* Chapter 9 focuses on how to build a content delivery networks for specific circumstances.

[25] Hofmann, Markus, and Leland Beaumont, *Content Networking: Architecture, Protocols,* Chapter 5 Navigating Content Networks has a detailed discussion of server load balancing techniques and actions.

manner in which they do this is beyond the scope of this paper, except for the actions relating to blocking performance. A threshold of clients, connections, and users must be set for each server and strictly enforced. This way more clients can be served more quickly, and a single server is not overloaded while others barely have any clients and requests. Blocking performance is a basic process that can be implemented on a number of different levels and is key to a balanced server load.[26]

CONTENT DELIVERY NETWORK SOLUTIONS AND WEB SWITCHES

Along the transmission path of online video are a number of features that streamline, balance, and create high quality service including content delivery networks (CDN), interception proxies, callout servers, and web switches. Each of these features can also present difficulties and should be implemented only on a case-by-case analysis of how much each will contribute to a transmission path that helps the content host serve users most effectively. The processes outlined in this section give a range of options for content providers to improve their quality of service for streaming and online video.

Content Delivery Networks

Intelligent serving or intelligent streaming are terms used to reflect the way CDNs constitute an intelligent structure applied to the streaming process. CDNs work on top of existing networks to provide better delivery. They are especially important for what is still the most common form of streaming delivery, HTTP by TCP over IP web servers. Some of the key elements of many CDNs are callout servers, global request routing, nodes, dynamic content, interception proxies, and service nodes.

Web Switches

Web switching presents solutions to server scaling, load balancing, reliability, flexibility, maintenance, and security. The OSI (Open Systems Interconnection) model has various web switches on all seven of its rigid network layers, including layer-two switches that are commonly referred to as web routers. The web switches that are of most interest to this paper work on layers four through seven. Layer-four switches are generally used for network address translation and directing web traffic. These switches are given a virtual IP (VIP) address to which traffic is filtered before going to web servers. They use specific port numbers that filter certain types of requests such as HTTP, FTP, and streaming communications. These switches filter and direct data packets according to certain other parameters as well, in addition including the server-load balancing techniques discussed in the previous section. Layer-four switches unfortunately shift control away from the ends, provide a single point of failure, have trouble handling multiple connections, and have difficulty with encryption. Most systems use Virtual Router Redundancy Protocol (VRRP) to provide protection from these issues.

Messages and data packets undergo several changes after the layer-four switches that are authorized solely by the content provider as opposed to both ends. Layer-seven

---

[26] Arankalle, Poorva, and Padmavathi Mundur, “Optimal Server Allocations for Streaming Multimedia Applications on the Internet,” *Computer Networks* 50 (2006): 3608-3621. Shows the importance of blocking in server allocations for streaming video.

switches operate on the application level and are associated with specialized servers with specific types of content. At the layer-seven switch, the TCP connection is ended and the message is modified as necessary; the data packets are reformatted to specific types of connections, user needs, and content provider communications. Layer-seven switches help balance server loads by Uniform Resource Locator or by content. Layer-seven switches are most often associated with web caches and interception proxies.

WEB CACHES AND PLAYERS

It is important to understand buffers and caches as both are used in progressive download and streaming video. Caching is a process that is used more for true web streaming or adaptive bit rate streaming than progressive download that uses more data buffer on the end-users device. A data buffer is a small part of disk memory (called buffer memory) used as temporary storage for the whole file for progressive download. A cache is a mechanism of temporary storage of video that is stored by servers in order to send the file without reloading it. There are many types of buffers and caches between origin and user. User web players are the ultimate decoders for what is transmitted, and they interact and benefit from both buffers and caches.

Data Buffer

Buffers are regions of physical memory storage to temporarily hold data while it is being moved. They are used for faster access and when there is a difference between the rate a file is received and rate it can be processed. Data buffers are a very temporary form of storage, and their data are deleted in a timely manner. They are used to store the full file for progressive download and small portions for streaming.

Buffers also help with the smoothing of data and with proper playback. Audiovisual smoothing mitigates the bursty nature of information that arrives by Internet. Network channels can increase delay of transmission through packet loss and congestion that leads to data arriving in bursts. This results in a jitter effect for audiovisual material if it is displayed as received. Data is stored elastically in the data buffer, however, and then played back at a constant rate that makes the material appear smoothly.

Web Caches

Caches are components that store data so future requests can be served more quickly. They achieve this in two main ways: reducing the amount of information that needs to be transmitted and reducing access to slower underlying storage. Segments of the video and audio content are made available at several servers thus allowing faster transfer to users. The video is segmented into chunks for caching and then divided further to disk block size. Each video segment is cached and replaced independently, reducing network congestion and improving efficiency. Traffic to the original content server is unfortunately increased due to the independent nature of the segments and their need to be replaced. Segments are thus only replaced when not being accessed and are preserved as long as possible. The sizes of the chunks are a trade-off between numbers of gaps in the picture and the flexibility of replacing segments. Several users can be served from a single cache without pulling from storage and opening a new connection each time.[27]

There are different ways that caches help with online video, including fast prefix transfer and dynamic caching. Fast prefix transfer is a way to remove the two most common delays in streaming, buffer delays and connection delays. It is a process where the cache stores the beginning of the stream so it can quickly fill users' requests while the rest of the stream downloads. The request for content goes directly to the cache where it sends what it has and requests the rest from the origin server. The remainder of the stream is stored in the cache as it continues to fill user requests. Key to a successful fast prefix cache is having the rate of transmission be greater than the rate of playback.

Dynamic caching has one stream serve several clients on a small time delay using what is known as a ring buffer.[28] Dynamic caching provides a small moving window of time between a stream going to a specific user and then being reused for another client. When the new client connects to the stream they are behind the original client and data transfer, marking a small time difference between the two requests. When content needs to be retrieved from the origin server to make up the time difference between the two requests for the second client, a stream patch is opened to retrieve the data. Basically, a user joins an ongoing stream after it is in progress and only a small patch is opened to get the missing data limiting bandwidth usage, origin server usage, and network delay. The more users, the more segments in the ring buffer are active and used to serve more users. Multiple ring buffers can cooperate to form a streaming distribution network and patches go not to the origin server, but to other caches. Ring buffers and dynamic caching are especially useful for serving multiple clients in a short time span.

REAL-TIME PROTOCOLS

For full functionality during streaming video network transmission, certain protocols need to occur among servers, host, end, and intermediary systems. These include real-time protocols and languages that are layered and used in conjunction with one another. The protocols listed in this paper are open-source, not proprietary ones like the formerly ubiquitous Adobe Flash RTMP (Real-time Messaging Protocol). These protocols offer flexible solutions for streaming, adding the functions of random access, time-based access, and timing control.

Real-Time Transport Protocol (RTP)

This protocol is designed for end-to-end network delivery services for applications transmitting real-time data. RTP is implemented over UDP to use its multiplexing and checksum services, while switching to TCP for firewall access. Messages using RTP have a payload-type ID, sequence numbering, timestamp, and delivery monitoring. RTP is especially useful for multicast distribution and when it is important to ID parts or frames of the video and audio streams. RTP sends two streams, one for audio and one for video, and encodes synchronization information using Synchronization Source IDs

---

[27] Hofmann, Markus, and Leland Beaumont, *Content Networking: Architecture, Protocols,* pages 95–104.

[28] Hofmann, Markus, and Leland Beaumont, *Content Networking: Architecture, Protocols,* pages 101–104.

(SSRCs) in its message header. When the stream is recompiled, the audio information is synchronized first and then data is combined and transformed into new messages.

RTP Control Protocol (RTCP)
This protocol is used in conjunction with RTP to periodically control packets in a communication session. RTCP also offers feedback on quality of data distribution similar to flow and congestion controls in other transport protocols. This feedback is provided in a receiver report (RR) packet. The packet details packet loss rate between two receiver reports and information on jitter resulting from network congestion.

Real-Time Streaming Protocol (RTSP)
RTP delivers the stream in a manner similar to continually moving videotape, while RTSP provides VCR functionality like play, pause, and others. RTSP generally operates over TCP while intersecting with other protocols over UDP. There are a series of profiles in RTSP for specific applications. These profiles define payload (data packets) type codes and mapping too specific formats as well as extensions and modifications for particular classes of applications.

CURRENT SERVICES AND OPTIONS

There are a variety of readily available services for online video content. Some apply more intelligence structures and have more options in serving clients. Two of the largest and most important services are among the most simple in terms of delivery, YouTube and Vimeo. Both offer only progressive download over basic web servers with some intelligence and infrastructure applied to their networks. Since the services use the larger standard web server infrastructure, serving content to remote users is not a problem. They are not as secure as other options and their content is easily ripped or copied during transmission, but they are easy to use and have built-in audiences. YouTube is the largest form of online video content delivery today and reach the largest audience possible. Vimeo was originally designed for the independent film or movie making community and retains some barriers to discovery for users, but like YouTube offers an easily embeddable player for any device and decoder. Neither of these options offers permission controls or other security methods to safeguard content. They are, however, easily discoverable through a simple web engine search. They also offer some level of storage redundancy by saving a copy of the video to their storage and servers.

Progressive Download
- YouTube
- Video

The Internet Archive is a fully dedicated digital library and archive designed to collect, preserve, and make accessible as much content as possible. It uses standard web servers for online video delivery offering adaptive-bit-rate streaming, progressive download, and regular file download. Adaptive-bit-rate streaming uses more protocols and provides a higher degree of security than progressive download. It transmits small segments of the video using 128-bit encryption through a Secure Sockets Layer for security. This does not matter as much with the Internet Archive since it makes all materials available for immediate download, but the stream itself is secure. The Internet Archive is widely available on the Internet, but does not have the market penetration or reach of YouTube. It provides redundancy and creates derivatives for the content provider free of charge, although the provider has to give up a certain degree of control over their content.

Adaptive-Bit-Rate Streaming and Progressive Download

- Internet Archive

There are a variety of services that offer full-range and highly personalized solutions for any online video or online content delivery. These include the Real Media Helix Server, Windows Media Server, Apple QuickTime, Wowza, Brightcove, Kaltura, and Akamai. These services offer full content delivery network solutions, security measures, specialized servers, web servers, infrastructure, and support for online video delivery. They, unfortunately, are not free and in comparison to services with large and free user-bases like YouTube, and suffer as a result. The benefits and options these services present, however, oftentimes outweigh these costs. They can setup and run video delivery services to any specifications; they are recommended for large collections of online video content.

Progressive Download, Adaptive-Bit-Rate Streaming, True Streaming

- Real Media Helix Server
- Windows Media Server
- Apple QuickTime
- Wowza
- Brightcove
- Kaltura
- Akamai

Rather than just choosing one of these options, a selection of several will broaden user bases and increase the presence of online video on the Internet. One could upload to YouTube or Vimeo, for general web discovery, while using Akamai or Wowza to deliver to certain communities or on a personal website. The possibilities for personalization of web delivery services are there. Each organization has to find the right combination for its users, content, and mission.

CONCLUSION

Online video delivery is the next great transmission standard for moving image materials held by cultural heritage, educational, and similar institutions. The rise of cheap storage, development of Internet architecture, and growth of digital video production on all levels has combined with the demand for video delivered over the Internet to make this so. Streaming and progressive downloads are not just possible delivery mechanisms, but necessary ways to engage and keep current with users. It has never been easier to make online video a reality, now with a variety of options and with security. For many users, if the content, not just the record, is not available online it will not be consulted. For all practical purposes, it does not exist for them. While downloading plug-ins for web browsers adds a barrier to entry, since users generally do not want to take the time to download and install additional software, a bigger impediment to serving users would be no online video presence in the first place. Cultural heritage and other institutions that collect moving images need to assess the possibilities that are out there, understand them, and implement solutions so that their moving image materials are not lost to the way both researchers and the public now access materials.

Bibliography/Webography

Arankalle, Poorva, and Padmavathi Mundur. "Optimal Server Allocations for Streaming Multimedia Applications on the Internet." *Computer Networks* 50 (2006): 3608-3621.

Austerberry, David. *The Technology of Video & Audio Streaming* 2nd ed. New York: Elsevier. 2005. Print.

Beaumont, Leland R. and Markus Hofmann. *Content Networking: Architecture, Protocols, and Practice.* San Francisco: Morgan Kaufmann Publishers, 2005. Print.

Çelik, Turgay, and Zeki Yetgin. "Efficient Progressive Downloading Over Multimedia Broadcast Multicast Service." *Computer Networks* 56 (2012): 533-547.

Cicco, Luca De, and Saverio Mascolo. "An Experimental Investigation of the Akamai Adaptive Video Streaming." *Presentation USAB 2010, special session Interactive Multimedia Applications (WIMA)*, Klagenfurt, Austria, 3-4 November 2010, LCNS 6389, pp. 447-464, Springer-Verlag, doi:10.1007/978-3-642-16607-5. PDF e-journal.

Clark, Jason A. "YouTube University: Using XML, Web Services, and Online Video Services to Serve University and Library Video Content." *Library 2.0 Initiatives in Academic Libraries.* ed. Laura B. Cohen. Chicago: Association of College and Research Libraries. 2007. Print.

Dostálek, Libor and Alena Kabelová. *Understanding TCP/IP: A Clear and Comprehensive Guide to TCP/IP Protocols.* Birmingham: Packt Publishing. 2006. Print.

Goralski, Walter. *The Illustrated Network: How TCP/IP Works in a Modern Network.* Burlington: Morgan Kaufmann Publishers, 2009. Print.

Hilderbrand, Lucas. "YouTube: Where Cultural Memory and Copyright Converge." *Film Quarterly* 61, No. 1 (2007): 48-57.

Hofmann, Markus, and Leland Beaumont. *Content Networking: Architecture, Protocols, and Practice.* New York: Morgan Kaufmann. 2005. PDF e-book.

Houmkozlis, Christos N., and George A. Rovithakis. *Automation and Control Engineering: End to End Adaptive Congestion Control in TCP/IP Networks.* New York: CRC Press, 2012. PDF e-book.

Inglis, Andrew F., and Arch C. Luther. *Video Engineering. 3rd Ed.* New York: McGraw-Hill, 1999. Print.

Kwon, Soon-kak, K.R. Rao, and A. Tamhankar. "Overview of H.264/MPEG-4 Part 10." *J. Vis. Commun. Image R* 17 (2006): 186-216. PDF e-journal.

Liu, Yunqiang, Songyu Yu, and Jun Zhou. “Adaptive Segment-Based Patching Scheme for Video Streaming Delivery System.” *Computer Communications* 29 (2006): 1889-1895.

Nelson, Dave. “Windows Media Server or Web Server?” *Microsoft,* Last modified May 15, 2008. http://www.iis.net/learn/media/windows-media-services/windows-media-server-or-web-server

“Web Server vs. Streaming Server.” *Microsoft Corporation,* Last modified 2010. http://web.archive.org/web/20100821050807/http://www.microsoft.com/windows/windowsmedia/compare/webservvstreamserv.aspx

Ozer, Jan, and Stefan Richter. *Hands-on Guide to Flash Video: Web Video and Flash Media Server.* New York: Elsevier. 2007. Print.

Ozer, Jan. “Streaming Vs. Progressive Download Vs. Adaptive Streaming.” *Information Today, Inc,* 2013. Print.

Pfeiffer, Silvia. *The Definitive Guide to HTML5 Video.* New York: Springer Science+Business Media, Inc., 2010. Print.

“Progressive Download” *Wikipedia.* Last modified December 9, 2013. http://en.wikipedia.org/wiki/Progressive_download

Real Team. “The History of Digital Video File Formats” *Real Networks, Inc.* Last modified April 12, 2012. http://www.real.com/resources/digital-video-file-formats/

Scharr, Mihaela van der, Thomas Stockhammer, and Deepak S. Turaga. *MPEG-4 Beyond Conventional Video Coding: Object Coding, Resilience, and Scalability.* New York: Morgan & Claypool, 2006. PDF e-book.

Sodagar, Iraj, and Thomas Stockhammer. “MPEG DASH: The Enabler Standard for Video Delivery Over the Internet.” *SMPTE Motion Imaging Journal* 121, no. 5 (2012): 40-46.

“Splitting a Stream”. *Microsoft.* Accessed December 16, 2013. http://msdn.microsoft.com/en-us/library/windows/desktop/dd893272%28v=vs.85%29.aspx

“Streaming Media” *Wikipedia.* Last modified December 13, 2013. http://en.wikipedia.org/wiki/Streaming_media

Weinsten, Stephen. *The Multimedia Internet*, Edited by Jack Keil Wolf. New York: Springer Science+Business Media, Inc., 2005. Print.