Glass to Glass Internet Ecosystem Introduction
draft-deen-daigle-ggie-02

Abstract

This document introduces the Glass to Glass Internet Ecosystem (GGIE). GGIE’s purpose is to improve how the Internet is used create and consume video, both amateur and professional, reflecting that the line between amateur and professional video technology is increasingly blurred. Glass to Glass refers to the entire video ecosystem, from the camera lens to the viewing screen. As the name implies, GGIE’s scope is the entire video ecosystem from capture, through the steps of editing, packaging, distributed and searching, and finally viewing. GGIE is not a complete end to end architecture or solution, it provides foundational elements that can serve as building blocks for new Internet video innovation.

This is a companion effort to the GGIE W3C Taskforce in the W3C Web and TV Interest Group.

This document is being discussed on the ggie@ietf.org mailing list.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 27, 2017.
Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ....................................................... 3
2. Terminology ....................................................... 4
3. Motivation: Video is filling up the pipes ...................... 4
4. Video is different .................................................. 5
5. Historical Approaches to supporting Video on the Internet .. 6
   5.1. Video as an application ..................................... 6
   5.2. Video as a network problem ................................. 7
   5.3. Video Ecosystem Encapsulation .............................. 7
6. Problem Statement and Solution Criteria ........................ 8
7. The Glass to Glass Internet Ecosystem: GGIE ................... 8
   7.1. Related work: W3C GGIE Taskforce .......................... 9
8. GGIE work of relevance to the IETF ............................... 9
   8.1. Affected IETF work areas ................................. 9
   8.2. Example use cases ........................................... 9
   8.3. Core GGIE elements ....................................... 11
9. Conclusion and Next Steps ......................................... 15
10. Acknowledgements ................................................ 15
11. IANA Considerations ............................................. 15
12. Security Considerations ......................................... 15
   12.1. Privacy Concerns ......................................... 15
13. Normative References ............................................. 16
Appendix A. Overview of the details of the video lifecycle ... 16
   A.1. Media Lifecycle ........................................... 16
   A.2. Video is not like other Internet data ...................... 19
   A.3. Video Transport .......................................... 21
Authors’ Addresses .................................................. 21
1. Introduction

In terms of shear bandwidth, the Internet’s largest use, without any close second competitor, is video. This is thanks to the proliferation of Internet connected devices capable of capturing and/or watching streamed video. As of 2015 there are reports that YouTube users upload over 500 hours of video every minute, and that during evening hours Netflix accounts for a staggering 50+% of Internet traffic. The number of users using the Internet for both ends of the video create-view lifecycle grows daily worldwide, and this is creating an enormous strain on the underlying Internet infrastructure at nearly every point from the core to the edge.

While video is one of the most conceptually simple uses of the Internet, it is perhaps one of the most complex technically, built from standards created by a large number of organizations and groups some dating from before the modern Internet even existed. Many critical parts of this complex ecosystem were not created with either video’s particular characteristics or vast scale of popularity in mind. This has lead to both the degradation of the viewer experience and many Internet policy issues around access to bandwidth for video and the needed infrastructure to support the continued explosion in video transport on the Internet.

The pace of video growth has been faster than new bandwidth for the past many years, and all indicators are that, instead of abating, it is actually accelerating as new users, new ways of sharing video, and new types of video continue to be added. The Cisco Visual Networking Index an excellent source of detail on this subject.

The combined current high levels of bandwidth consumed by video, plus the accelerating pace of video’s growth mean that to meet users’ demand for video, we must do more than simply rely on adding more bandwidth. While other traditional improvements such as more efficient codecs with better compression ratios are expected to contribute to keep video flowing on the Internet, many in the Internet video technology world have explored options to see if any new approaches could be added to the mix to help the problem. That was the motivation behind the creation of the GGIE Taskforce within the W3C in 2014 with the charter to examine the end to end video ecosystem and identify new areas of opportunity to improve video’s use of the Internet.

The W3C GGIE taskforce explored ways that video uses the Internet and developed a series of use cases detailing specific scenarios ranging from video capture, the editing and production cycle, through to delivery to viewers. Out of these use cases there emerged a recognition that there might be a new opportunity to improve Internet
video by enabling edge devices, and the underlying network to more actively participate in making delivery optimization choices beyond the simple ways the do currently.

The GGIE approach is to apply and evolve existing technologies to the task of optimizing Internet video transport to permit applications, video devices, and the network to more actively participate in making smart access and transport decisions. This approach recognizes that there are already extensively-deployed video infrastructure elements that need to continue to work and be part of the optimized video ecosystem. These deployed devices, applications, players, and tools are responsible for the already high levels of video bandwidth consumption, and to only address new devices would not be solving the larger, most important problem. This is why GGIE is an evolution of how video uses the Internet, and not a revolution involving wholesale replacement of existing architecture or protocols.

GGIE is not a complete solution to the video problem. It provides foundational building blocks that are intended to be used by innovators in their work to create new optimizations, and novel techniques to help address the video problem in the long term.

GGIE initially proposes a simple framework of three components that will permit improved playback device identification of viewing sources and enable network level awareness of video transport and new cache selection choices. GGIE proposes: Using existing content identifiers as a means to identify a work, or title; Data level identifiers to identify the encoded video data for a particular manifestation of the title; A mapping service that permits bi-directional resolution of these identifiers.

This document outlines the basic proposal for these three base GGIE components and introduces the overall GGIE approach to evolving the current video ecosystem by introducing basic standardized building blocks for innovators to build upon the Glass to Glass Internet Ecosystem.

2. Terminology

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Motivation: Video is filling up the pipes

   The growth in video bandwidth need is exceeding the growth in the bandwidth provisioning. This trend is in fact accelerating, meaning the growth rate of video is growing faster than the growth rate of
provisioning. Traditional techniques of caching, higher efficiency codecs, etc, are all being used to help address the problem and have helped the Internet to continue to support the growth of video thus far.

Video has been the top use of Internet bandwidth for several years and is larger than the bandwidth used by all other applications combined. This trend is unlikely to ease or reverse itself as users of the Internet continue to make Internet transported video one of their top uses of the Internet, either for uploading and sharing video they creator, or as a primary sources for viewing video to a wide variety of viewing devices: computers, tablets, phones, connected televisions, game consoles, and AV receivers.

Adding to user demand, video itself is continually experiencing innovation introducing ever higher resolutions (SD, HD, 4K, 8K...), higher video quality, new distribution services (live one to many streaming), and new user uses. The Cisco Visual Networking Index projects that by 2019 there will be nearly a million minutes of video per second transported by the Internet, a making up 80-90 percent of all IP traffic.

The motivation behind GGIE is to help find new methods that can be brought to bear, in addition to all the exiting ones, to help manage the explosion in Internet video.

4. Video is different

Video is different than other uses of the network due to its combined high bandwidth demands and high sensitivity to latency and dropped packets. Streaming of basic high-definition 1080p requires bandwidth in the low Mbps translating into Gigabytes for each hour of video, all transported with consistent low latency and very little packet loss in order to deliver a suitable watching experience the viewer. This differentiates video from other Internet applications as some have low latency and packet loss requirements but don’t need high bandwidth, while others may demand high bandwidth, they will tolerate high latency and dropped packets. An email user can tolerate an extra moment to retransmit dropped packets, and a web page user can tolerate a slow DNS lookup, but a video viewer sees latency and dropped packets as jittery playback and low bandwidth as a fundamental barrier to streaming at all. From the user’s perspective the network has failed to meet their need. (Audio has similar challenges in terms of intolerance of delay and jitter, but the data sizes are significantly smaller).

Video data sizes continue to grow at roughly 4x per format iteration as cameras and playback devices are able to capture and display
higher quality images. Early digital video was often captured at either 320x240 pixel resolution or 640x480 standard definition resolution. High definition or HD video at 1920x1080 became possible on some parts of the Internet after 2011, although even in 2016 it remains unavailable or unreliable through many connections such as DSL and many mobile networks. Camera and player technologies are currently expanding again to permit 4K or 3840x2160 pixel resolution reflecting a 4x data increase over HD.

Streaming is very demanding, requiring consistent frame to frame playback in consistent constant time. Advanced features such as pause, fast forward, rewind, slow motion, and fine scrubbing are considered by users as standard features in players that the network must support and serve to further the challenge facing the Internet.

New video abilities such as live streaming by users (both one to one and one to many) bring what has traditionally been done by professional broadcasters with dedicated broadcast infrastructure into the realm of every day users with connected smartphones using the Internet as a real-time global broadcast infrastructure.

5. Historical Approaches to supporting Video on the Internet

5.1. Video as an application

Internet video engineering began by adapting preexisting standards used for over the air broadcast (OTA) and physical media. Video encodings, such as AVI and MPEG2, originally designed for playback from local storage connected to the player where added to the data types carried by existing protocols like HTTP, and new protocols such as RTSP and HLS. Early use of the Internet for video was a copy-and-play model replacing the use of OTA broadcast and physical media to copy video between systems.

As Internet bandwidth became sufficient to allow delivery of video data at the same rate it was being decoded, it became possible to stream video originally at very low resolutions such as 160x120 pixels (19.2 kilopixels), eventually permitting standard definition (SD) 640x480 pixels (0.3 megapixels), and later high definition of 1920x1080 pixels (2 megapixels). This trend continues with some providers beginning to offer 4K or 3840x2160 pixels (8.3 megapixels) requiring very reliable and generous Internet bandwidth end to end connection between the viewer and source.

Unlike the Web, email, and network file sharing which have been engineered and standardized in Internet focused organizations such as the W3C and IETF, video is dependent on standards developed by a very large number of groups, companies, and organizations which include.
the IETF, W3C but also MPEG, SMPTE, CTA, IEEE, ANSI, ISO, networking and technology companies, many others. In contrast to the extensive end to end expert knowledge and engineering done to create the Web and email, Internet video has largely been an evolved cobbling and adaption exercise done by engineers with their focus on a few, or one, particular aspect or problem at a time, and little interaction between other parts of the Internet video ecosystem. While it is very much possible to deliver video over the Internet, this uncoordinated cobbling has resulted in many areas of inefficiency where engineering done from an end to end perspective could provide the opportunity to vastly improve how video uses the Internet, which offers the hope of improving the quality of video and increasing the amount of video which can be delivered.

5.2. Video as a network problem

Network, video, and application engineers have constructed elaborate solutions for dealing with bandwidth and processing limitations, network congestion, lossy transport protocols, and the ever growing size of video data. These solutions commonly fall into one of several solution types:

1. Reducing data sizes through resolution changes, compression, and more efficient encodings
2. Downloading before playing instead of real-time streaming
3. Positioning the data close to the viewer via caches, typically on the network edge
4. Fetching of video data at a rate faster than playback
5. Transport protocols that attempt to deliver video data such that the data arrives as if it were done on a congestion free/lossless network
6. Dynamic reselection of sources and transport routes on either a real-time or frequent intervals, 10-15 seconds, using player feedback mechanisms or network telemetry

5.3. Video Ecosystem Encapsulation

The current delivery ecosystem for video has been primarily developed at the higher application layers of the stack. While there has been some video work done at lower levels such as general-purpose transport improvements, caching protocols in CDN, various multicasting approaches, and other efforts, the majority of video-specific work has previously been done by groups such as ISO’s Moving
The Pictures Expert Group (MPEG) which have focused on codecs and codec transport optimized for use on the Internet. These efforts have made video possible on the Internet, but they have done so largely while treating the underlying network as a basic transporter of data. This has resulted in little information being exposed to the network, information that could be used to optimize delivery of the video, and in an architecture that pushes more and more of the intelligence into an ever more complex and isolated core.

The current video model benefits from a significant amount of operational, feature, and protocol encapsulation that has come about due to different groups working independently on the components that make it up. Like any system in which distinct pieces are well encapsulated from one another, this means it is possible to engage in improvements at the networking layer without the need to coordinate with higher levels of the video architecture.

6. Problem Statement and Solution Criteria

At its most basic the problem to be solved for video delivery is how to simultaneously maximize all of the following conditions: The number of viewing devices simultaneously supported by the network; The quality of video as measured by bit-rate and resolution; The number of distinct unique streams that can be delivered.

Solution Constraints

1. Bandwidth growth alone is not a solution
2. Codec efficiency improvements alone are not a solution
3. Existing devices, infrastructure, video delivery techniques must as much as possible continue to be supported and benefit from new solutions.

7. The Glass to Glass Internet Ecosystem: GGIE

GGIE is an effort to improve video’s use of the Internet by examining the end to end video ecosystem from the glass lens of the camera through to the glass the screen, and to identify areas of simplifications, standardization, and reengineering to make better use of bandwidth enabling smarter network use by video creators, distributors, and viewers. GGIE is focused on how video uses the Internet, and not on how it is encoded or compressed. Likewise GGIE does not deal with content protection. GGIE’s scope however does include creator and viewer privacy, content identification and recognition as a means to enable smarter network usage, edge caching, and discoverability.
GGIE benefits from the encapsulation of the video ecosystem elements enabling it to introduce evolutiveal features to elements without disrupting other distinct encapsulated parts.

GGIE is intended to work with a wide variety of video encoding codecs, and video distribution and transport protocols. While examples using MPEG-DASH are used due to its pervasive use, GGIE is not limited to MPEG-DASH or any other video distribution system or codec.

Beyond improving the simple experience of a viewer using the Internet to watch linear video, it is hoped that a set of improved Internet video infrastructure standards will provide a foundation that permits innovators to create the next generation of Internet video content (such as multisource personalized composite experiences, interactive stories, and live personal broadcasting, to name a few).

Due to the very diverse and large deployment of existing video playback devices and infrastructure, it is viewed as essential that any evolved ecosystem continues to work with the majority of the legacy deployment without the need for updates or changes to the existing ecosystem.

7.1. Related work: W3C GGIE Taskforce

A companion effort ran through 2015 in the W3C Web and TV Interest Group’s GGIE Taskforce. The W3C GGIE group developed a series of use-cases on discovery, search, delivery, identity, and metadata which can be found at https://www.w3.org/2011/webtv/wiki/GGIE_TF

8. GGIE work of relevance to the IETF

This section assumes a working familiarity with video creation and consumption "life cycle". For reference, an overview has been provided in the Appendix.

8.1. Affected IETF work areas

It is expected that significant improvement is possible in the video transport ecosystem by modest evolution and adaption of existing standards for addressing, transporting, and routing of video data flows between sources and display.

8.2. Example use cases

The following example use case help illustrate the use of the GGIE core elements
8.2.1. Alternate Source Discovery

Description: A video player is streaming a movie from a CDN cache in the core of the network. This use case illustrates the use of a media identifier to query a media address resolution service to locate additional alternate sources that offer the same movie.

1. The video player user selects a movie to watch from a list using the player application UI.

2. The video player application has the media identifier of the movie in the metadata description of the movie. This identifier is passed to the playback device when the movie selected.

3. The playback device send a search query to the Media Address Resolution Service (MARS) which includes the media identifier, and additional query parameters use to filter the results returned.

4. The MARS server searches its database and returns all the Media Encoding Networks matching the media identifier and filters the results using the additional parameters submitted in the query. Each Media Encoding Network represents a different encoding of the video.

5. The player then examines the returned list of media encoding networks and selects, from its perspective, the optimal source for the title.

6. The player then directs its streaming requests to the selected Media Encoding Network addresses to obtain the video data for the movie.

7. The video data is decoded and displayed on the screen

8.2.2. Alternate Format Discovery

Description: A video player is streaming a movie, and wants to send the audio to another device for playback. However, the current video data being streamed does not contain any audio that matches the codecs the audio device can play. The audio device uses the core GGIE services to locate an alternate encoding of the movie that contains audio it can decode.

1. The user directs the video player to send the audio portion of the playing video to an external audio device.
2. The video player application passes the media identifier for the video to the audio device as well as the media encoding network address the video player is using.

3. The audio device begins streaming from the media encoding network it was given, but discovers the data does not include audio that is able to decode.

4. The audio device sends a search query to the Media Address Resolution Service (MARS) which includes the media identifier, and additional query parameters including the list of audio codecs and language choice it is able to decode.

5. The MARS server searches its database and returns all the Media Encoding Networks matching the media identifier and filters the results to only those matching the language and audio codec supplied in the search.

6. The audio player examines the returned list of media encoding networks, selects a media encoding network and begins streaming data from it.

7. The external audio player decodes the returned movie data and plays it for the user.

8.3. Core GGIE elements

GGIE proposes three initial fundamental pieces:

1. Media Identifiers which identify the video at the title, or work level;

2. Media Encoded Networks which are subnets used to reference the encoded video data;

3. Media Address Resolution Service which maps Media Identifiers for a title to the Media Encoded Networks containing the encoded video versions of the title.

These three foundational elements help by exposing information that can be used in selection in a way that is independent of the video encoding and video data storage choice. It also enables more sophisticated video use cases beyond the basic single device playing a video stream from an origin server over a flow controlled protocol.
8.3.1. Media Identifiers

A Media Identifier is a URI that carries a content identifier system declaration, and a content identifier from the system that refers unambiguously to a work, or title. This can be any content identification system, GGIE does not specify the system used.

For example, a media identifier for a title identified by an EIDR value would include a declaration that the identifier is from EIDR, and would additionally contain the EIDR value.

At the application level, such as UI program guide applications, search engines, and metadata databases, it is the identification of the work or identity of the video that is typically of interest and not the encoding, bit-rate, or the location of CDN caches etc. For example, a UI would indicate that "the Minions movie" as opposed to "a 15 megabit per second, HEVC encode with high dynamic range and Dolby encoded 7.1 English audio of the Minions movie". Those additional technical details are important when choosing a particular encoded manifestation of the movie for delivery, decode, and playback, but they are not generally needed as information to be presented to the user or used to make viewing choices. Such technical information is used after the user has chosen the title to watch, but is used by the playback device not the user in selecting the video. Media Identifiers in GGIE contain only title information, and not encoding information.

There are many media identifiers in use for both personal and professional content, with new ones being introduced seemingly weekly. To try to create a single identifier to either harmonize or replace the others, repeatedly been proven in practice to be an impossible task. Recognizing this, the GGIE instead proposes to standardize a URI which would contain at least two fields: 1) A scheme identifier; 2) An unambiguous title identifier (note: this is unambiguous only within domain of the identified scheme).

For professional content, titles are increasingly identified with a scheme called EIDR that can identify both master versions of works, and edit level versions. Likewise, advertisements use a scheme called AD-ID.

8.3.2. Media Address Resolution Service (MARS)

The media address resolution service (MARS) provides bidirectional mapping of Media Identifiers to Media Encoding Networks. It is queryable using a query protocol which returns any results matching the terms of the query parameters.
A Media Identifier alone isn’t sufficient to connect a device to a video data source. The media identifier distinguishes the work, but not the technical details of an instance of the work such as codec, bit-rate, resolution, high dynamic range video, audio encoding, nor does it include information about available streaming sources etc. The Media Address Resolution Service (MARS) provides this association. It can be queried with the Media Identifier, and optional filtering parameters, and will return Media Encoding Network addresses for instances of matching encodings of the work.

This translation is used commonly in video streaming services today. The link provided in the program guide UI will include a unique identifier for the work which is then mapped by the streaming service backend into a URI containing a network identifier and other info which point to a caching server and the media data files in the cache. MARS generalizes this and make it available via query over the network.

8.3.3. Media Encoding Networks (MEN)

Media Encoding Networks are arrangements of encoded video data that are assigned addresses under a shared prefix and subnet following a scheme appropriate for the encoding used by the video data. Each Media Encoding Network instance represents a distinct instance of a set of associated encodings for a work. Different Media Encoded Network address assignment schemes would be defined under GGIE to handle different encode data such as MPEG-DASH and HLS.

For example, a single MEN instance would hold each of the different variable bit-rate encodes for a single encoding of a video. If a new encoding instance of the video was prepared, it would have separate and distinct MEN assigned to it.

8.3.3.1. Example: Using Media Encoding Networks with MPEG-DASH

A very basic form a video delivery uses persistent connection from a player to a video file source which then streams the video by transmitting the video file data, byte by byte in sequence, from the first byte of the file until the last. This trivial approach requires the device to know the server IP address and port number to connect to. Essentially this involved simply transporting the file from the source to the playback device in byte order.

In practice simple file streaming is not used beyond local device to device playing in home networks as it doesn’t permit dynamic bit rate selection, source or session fail over, or trick play (pause, skip forward, skip backward) etc. Instead manifest files contain lists of available servers holding MPEG-DASH encodings of the larger video
file divided into fragments containing short portions (e.g. 2-15 seconds) of the video called chunks by MPEG-DASH. (GGIE generalizes the MPEG-DASH chunk term into the more general shards). Each shard is a distinct file typically named to reflect the video encode it belongs to, and it's sequence position.

For example the shards for MY-VIDEO might be names MY-VIDEO-001, MY-VIDEO-002, ... MY-VIDEO-nnn. The player then requests the shards in the order it wants them over a data transport protocol such as http, with the translation of the actual data sent in response to requests for the named shards being handled by the data server.

So under MPEG-DASH the player is sent a manifest file containing the address of the data server and the shard name to request. The player then iterates over the available shards in the order desired by the user. The manifest then contains URI’s with the SERVER-ADDRESS and the CHUNK name. This file can be sent once per video play, or more commonly is sent at an interval of ~15 seconds to permit the sending CDN to customize for each player, and to respond quickly to changes in the network delivery performance and availability.

Each shard request by the device involves a network level server IP address and port number, and an application level shard name. The network is thus able to manage the routing of request to the server, and the routing of the response, but it lacks the information needed to do anything else to help optimize the video data transport.

GGIE proposes using Media Encoding Networks an evolution of this that has the benefit of being backward compatible with manifest files, while enabling the transport network and video ecosystem to have more information to the network about the video transport flowing over it.

Using Media Encoding Networks for MPEG-DASH will be described in another Internet-Draft, but the basic proposal is to assign the shards into a sequence of IP addresses organized to reflect the same ordering association that the chunk names followed in the MPEG-DASH scheme. These shard addresses form a Media Encoding Network, and they expose to the network layer knowledge of the specific video data being transported between requesting device and the file server holding the data.

This in practice means that Media Encoding Network addresses refer to the shard and not the server holding the shard. This then permits the network to be involved in the routing of the request for the shard, as opposed to the CDN preparing the manifest file. Among other benefits, this permits the network to provide path failover functionality beyond the CDN manifest.
This enables the network to be involved in shard source selection. Consider the use case wherein the network becomes aware of a local cache that holds the requested shard, and is closer to the device than another cache deeper in the network. The network can direct the request to the local cache and save the transit cost and bandwidth of sending the request and response exchange with the deeper cache. This can reduce network congestion as well as deliver faster transport for the shard to the playback device.

8.3.4. Media Encoding Network Gateways

In this new approach, the server providing the shard data is possibly better viewed as acting as a gateway to the shard addresses versus being just a file server. In practical terms, existing CDN caches can perform this role by mapping the requested shard address to the on disk file containing the shard. However, new CDN caches can be developed work directly with the Media Encoding Network scheme, and can act as smart caches proactively provisioning data within the Media Encoding Network address space.

9. Conclusion and Next Steps

GGIE seeks to help address this problem by establish standards based foundational building blocks that innovators can build upon creating smarter delivery and transport architectures instead of relying on raw bandwidth growth to satisfy video’s growth.

Next steps will include describing the working prototypes of the GGIE core elements and more extended use cases addressed by GGIE many of which were defined in the W3C GGIE Taskforce.

10. Acknowledgements

Contributions to this document came from Bill Rose, Gaurav Naik, John Brzozowski.

11. IANA Considerations

None (yet).

12. Security Considerations

12.1. Privacy Concerns

The assignment of persistent IPv6 Prefixes to MEN permits the video being streamed to be identified at the network level by observing the destination addresses sent from the player to the media gateway. In situations where it is desired by the user to prevent this level of
observation is necessary to obscure the true MEN prefix of the video being streamed.

12.1.1. Privacy via VPN

One remediation is the use of a VPN that will encapsulate and hide the traffic between the player and the streaming cache, or at least between the trusted network the player resides on and the streaming cache network. This will make identification of the actual video title from the open Internet during transit.

12.1.2. Session Prefix Renumbering

Another technique is to have the player and streaming cache remap the IPv6 prefix for the streaming session to a new prefix. Under such a renumbering the cache will advertise to the routing layer and respond to requests sent from the player to the session prefix just as it would to the original video MEN prefix.

13. Normative References


Appendix A. Overview of the details of the video lifecycle

This section outlines the details of the video lifecycle -- from creation to consumption -- including the key handholds for building applications and services around this complex data. The section also provides more detail about the scope and requirements of video (scale of data, real-time requirements).

Note: this document only deals with streaming video as used by movies, TV shows, news broadcasts, sports events, music concert broadcasts, product videos, personal videos, etc. It does not deal with video conferencing or WebRTC style video transport.

A.1. Media Lifecycle

The complex workflow of creating media and consuming it is decomposable into a series of distinct common phases.
A.1.1. Capture

The capture phase involves the original recording of the elements which will be edited together to make the final work. Captured media elements can be static images, images with audio, audio only, video only, or video with audio. In sophisticated capture scenarios more than one device maybe simultaneously recording.

A.1.1.1. Capture Metadata

The creation of metadata for the element, and for the final video begins at capture. Typical basic capture metadata includes Camera ID, exposure, encoder, capture time, and capture format. Some systems record GPS location data, assigned asset ids, assigned camera name, camera spatial location and orientation.

A.1.2. Store

The storage phase involves the transport and storage of captured elements data. During the capture phase, an element is typically captured into memory in the capture device and is then stored onto persistent storage such as disc, SD or memory card. Storage can involve network transport from the recording device to an external storage system using either storage over IP protocols such as iSCSI, a data transport such as FTP, or encapsulated data transport over a protocol such as HTTP.

Storage systems can range from basic disk block storage, to sophisticated media asset libraries

A.1.2.1. Storage Metadata

Storage systems add to the metadata associated with media elements. For basic block storage, a file name and file size is typical, as are a hierarchical grouping, creation date, and last-access date. For library system an identifier unique to the library is typical, as well as grouping by one or more attributes, a time stamp recording the addition to the library and a last access time.

A.1.3. Edit

Editing is the phase where one or more elements are combined and modified to create the final video work. In the case of live streaming, the edit phase maybe bypassed.
A.1.4. Package

Packaging is the phase in which the work is encoded in one or more video and audio codecs. These maybe produce multiple data files, or they may be combined into a single file container. Typically, creation or registration of a unique work identifier, for example an Entertainment Identifier from EIDR, is assigned in the packaging phase.

A.1.4.1. Package Metadata

A.1.5. Distribute

The distribute phase is publishing or sharing the packaged work to viewers. Often it involved uploading to a site such as YouTube, or Facebook for social media, or sending the packaged media to streaming sites such as Hulu.

It is common for the distribution site to repackage the video often transcoding it to codecs and bitrates chosen by the distributor as more efficient for their needs. Distribution of content expected to be widely viewed often includes prepositioning of the content on a CDN (Content Distribution Network).

Distribution involves delivery of the video data to the viewer.

A.1.5.1. Distribution Metadata

Distribution often adds or changes considerable amounts of metadata. The distributor typically assigns a Content Identifier to the work, that is unique to the distributor and their content management system (CMS). Additional actions by the distributor such as repacking and transcoding to new codecs or bitrates can require significant changes to the media metadata.

A secondary use of distribution metadata is enabling easy discovery of the content either through a library catalog, EPG (electronic program guide), or search engine. This phase often includes significant new metadata generation involving tagging the work by genre (sci-fi, drama, comedy), sub-genre (space opera, horror, fantasy), actors, director, release date, similar works, rating level (PG, PG-13), language level, etc.

A.1.6. Discovery

The discovery phase is the precursor to viewing the work. It is where the viewer locates the work either through a library catalog, a
playlist, an EPG, or a search. The discover phase connects interested viewers with distribution sources.

A.1.6.1. Discovery Metadata

It is typical for discovery systems to parse media metadata to use the information as part of the discovery process. Discovery systems may parse the content to extract imagery and audio as additional new metadata for the work to ease the viewers navigation of the discovery process perhaps as UI elements. The system may import new externally generated metadata about the work and associate it in its search system, such as viewer reviews, metadata cross reference indices.

A.1.7. Viewing

The viewing phase encompasses the consumption of the work from the distributor. For Internet delivered video it is typical for delivery to involve a CDN to perform the actual delivery.

A.2. Video is not like other Internet data

Video is distinctly different from other Internet data. There are many characteristics that contribute to video’s unique Internet needs. The most significant characteristics are:

1. large size of video data (Gigabytes per hour of video)
2. high bandwith demands (Mbps to Gbps)
3. low latency demands of streamed video
4. responsiveness to trick play requests by the user (stop, fast forward, fast reverse, jump ahead, jump back)
5. multiplicity of formats and encodings/bit rates that are acceptable substitutes for one another

A.2.1. Data Sizes

Simply put compared to all other common Internet data sizes, video is huge. A still image often ranges from 100KB to 10MB. A video file can commonly range from 100MB to 50GB. Encoding and compression options permit streaming videos using bandwidth ranging from 700Kbps for extremely compressed SD video, to 1.5-3.0 Mbps for SD video, to 2.5-6.0 Mbps for HD video, and 11-30Mbps for 4K video.

Still images have 4 dimensional properties that affect their data size:
1. number of horizontal X pixels
2. number of vertical Y pixels
3. bytes per pixel
4. compression factor for the image encoding.

Video adds to this:

1. frames per second playback rate
2. visual continuity between frames (meaning users notice when frames are skipped or played out of order)
3. discontiguous jumps between frames such as skipping forward or backwards to inserting frames from other sources between contiguous frames (advertisement placement)

Each video format roughly increases by x4 the data needs of the previously resolution: (1) SD is 640x480 pixels; (2) HD is 1920x1080 pixels; (3) 4K is 3840x2160 pixels.

Video, like still images, assigns a number of pixels to store color and luminance information. This currently evolving alongside resolutions after being stagnant for many years. The introduction of high dynamic range videos or HDR has changed the color gamut for video and increased the number of bits needed to carry luminance from 8 to 10 and in some formats more.

Compression is often misunderstood by viewers. Compression does not change the video resolution, SD is still 640x480 pixels, HD is still 1980x1080 pixels. What changes is the quality of the detail in each frame, and between frames.

Video is in its simplest form a series of still images shown sequentially over time, adding an additional attribute to manage.

A.2.2. Low Latency Transport

Viewers demand that video plays back without any stutter, skips, or pauses, which translates into low latency, high reliability transport of the video data.
A.2.3. Multiplicity of Acceptable Formats

One of the unique aspects of video viewing is that there can exist multiple different encodings/versions of the same video, many of which are acceptable substitutes for one another. This is a unique aspect of video viewing and differentiates video delivery from other data transports.

Other application data types don’t have or leverage the concept of semantic equivalences to the same extent as video. Even email, which supports multiple encodings in a multipart MIME message, has a finite number of representations of "the message", shipped as one unit, whereas video often has many distinct different encodings each as separate file or container of files managed as a distinct entity from the others.

A.3. Video Transport

A.3.1. File vs Stream

There are two common ways of transporting video on the Internet: 1) File based; 2) Streaming. File based transport can use any file transport protocol with FTP and BitTorrent being two popular choices.

File based playback involves copying a file and then playing it. There are schemes which permit playing portions of the file while it progressively is copied, but these schemes involve moving the file from A->B then playing on B. FTP and BitTorrent are examples of file copy protocols.

Streaming playback is most similar to a traditional Cable or OTA viewing of a video. The video is delivered from the streaming service to the playback device in real time enabling the playback device to receive, decode, and display the video data in real time. Communication between the player and the source enable pausing, fast forward and rewind by managing the data blocks which are sent to the player device.

Authors’ Addresses

Glenn Deen
NBCUniversal

Email: rgd.ietf@gmail.com
Leslie Daigle
Thinking Cat Enterprises LLC

Email: ldaigle@thinkingcat.com