

MOPS  
Internet-Draft  
Intended status: Informational  
Expires: 10 September 2020

J. Holland  
Akamai Technologies, Inc.  
A. Begen  
Networked Media  
S. Dawkins  
Tencent America LLC  
9 March 2020

**Operational Considerations for Streaming Media**  
**draft-ietf-mops-streaming-opcons-01**

**Abstract**

This document provides an overview of operational networking issues that pertain to quality of experience in delivery of video and other high-bitrate media over the internet.

**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 <https://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 10 September 2020.

**Copyright Notice**

Copyright (c) 2020 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 (<https://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

<a href="#">1.</a>	<a href="#">Introduction</a>	<a href="#">2</a>
<a href="#">1.1.</a>	<a href="#">Venues for Contribution and Discussion</a>	<a href="#">3</a>
<a href="#">2.</a>	<a href="#">Bandwidth Provisioning</a>	<a href="#">3</a>
<a href="#">2.1.</a>	<a href="#">Scaling Requirements for Media Delivery</a>	<a href="#">3</a>
<a href="#">2.1.1.</a>	<a href="#">Video Bitrates</a>	<a href="#">3</a>
<a href="#">2.1.2.</a>	<a href="#">Virtual Reality Bitrates</a>	<a href="#">4</a>
<a href="#">2.2.</a>	<a href="#">Path Requirements</a>	<a href="#">4</a>
<a href="#">2.3.</a>	<a href="#">Caching Systems</a>	<a href="#">5</a>
<a href="#">2.4.</a>	<a href="#">Predictable Usage Profiles</a>	<a href="#">5</a>
<a href="#">2.5.</a>	<a href="#">Unpredictable Usage Profiles</a>	<a href="#">5</a>
<a href="#">3.</a>	<a href="#">Adaptive Bitrate</a>	<a href="#">6</a>
<a href="#">3.1.</a>	<a href="#">Overview</a>	<a href="#">6</a>
<a href="#">3.2.</a>	<a href="#">Segmented Delivery</a>	<a href="#">7</a>
<a href="#">3.2.1.</a>	<a href="#">Idle Time between Segments</a>	<a href="#">7</a>
<a href="#">3.2.2.</a>	<a href="#">Head-of-Line Blocking</a>	<a href="#">7</a>
<a href="#">3.3.</a>	<a href="#">Unreliable Transport</a>	<a href="#">8</a>
<a href="#">4.</a>	<a href="#">Doc History and Side Notes</a>	<a href="#">8</a>
<a href="#">5.</a>	<a href="#">IANA Considerations</a>	<a href="#">9</a>
<a href="#">6.</a>	<a href="#">Security Considerations</a>	<a href="#">9</a>
<a href="#">7.</a>	<a href="#">Acknowledgements</a>	<a href="#">9</a>
<a href="#">8.</a>	<a href="#">Informative References</a>	<a href="#">9</a>
	<a href="#">Authors' Addresses</a>	<a href="#">10</a>

## [1.](#) Introduction

As the internet has grown, an increasingly large share of the traffic delivered to end users has become video. Estimates put the total share of internet video traffic at 75% in 2019, expected to grow to 82% by 2022. What's more, this estimate projects the gross volume of video traffic will more than double during this time, based on a compound annual growth rate continuing at 34% (from [Appendix D](#) of [\[CVNI\]](#)).

In many contexts, video traffic can be handled transparently as generic application-level traffic. However, as the volume of video traffic continues to grow, it's becoming increasingly important to consider the effects of network design decisions on application-level performance, with considerations for the impact on video delivery.

This document aims to provide a taxonomy of networking issues as they relate to quality of experience in internet video delivery. The focus is on capturing characteristics of video delivery that have surprised network designers or transport experts without specific video expertise, since these highlight key differences between common assumptions in existing networking documents and observations of video delivery issues in practice.



Making specific recommendations for mitigating these issues is out of scope, though some existing mitigations are mentioned in passing. The intent is to provide a point of reference for future solution proposals to use in describing how new technologies address or avoid these existing observed problems.

### **1.1. Venues for Contribution and Discussion**

Note to RFC Editor: Please remove this section before publication

(To the editor: check this repository URL after the draft is adopted. The working group may create its own repository)

This document is in the Github repository at <https://github.com/GrumpyOldTroll/ietf-mops-drafts>. Readers are welcome to open issues and send pull requests for this document.

Substantial discussion of this document should take place on the MOPS working group mailing list ([mops@ietf.org](mailto:mops@ietf.org)).

## **2. Bandwidth Provisioning**

### **2.1. Scaling Requirements for Media Delivery**

#### **2.1.1. Video Bitrates**

Video bitrate selection depends on many variables. Different providers give different guidelines, but an equation that approximately matches the bandwidth requirement estimates from several video providers is given in [MSOD]:

$$\text{Kbps} = (\text{HEIGHT} * \text{WIDTH} * \text{FRAME\_RATE}) / (15 * 1024)$$

Height and width are in pixels, and frame rate is in frames per second. The actual bitrate required for a specific video will also depend on the codec used, fidelity desired and some other characteristics of the video itself, such as the amount and frequency of high-detail motion, which may influence the compressability of the content, but this equation provides a rough estimate.

Here are a few common resolutions used for video content, with their typical per-user bandwidth requirements according to this formula:



Name	Width x Height	Approximate Bitrate for 60fps
DVD	720 x 480	1.3 Mbps
720p (1K)	1280 x 720	3.6 Mbps
1080p (2K)	1920 x 1080	8.1 Mbps
2160p (4k)	3840 x 2160	32 Mbps

Table 1

### 2.1.2. Virtual Reality Bitrates

Even the basic virtual reality (360-degree) videos (that allow users to look around freely, referred to as three degrees of freedom - 3DoF) require substantially larger bitrates when they are captured and encoded as such videos require multiple fields of view of the scene. The typical multiplication factor is 8 to 10. Yet, due to smart delivery methods such as viewport-based or tiled-based streaming, we do not need to send the whole scene to the user. Instead, the user needs only the portion corresponding to its viewpoint at any given time.

In more immersive applications, where basic user movement (3DoF+) or full user movement (6DoF) is allowed, the required bitrate grows even further. In this case, the immersive content is typically referred to as volumetric media. One way to represent the volumetric media is to use point clouds, where streaming a single object may easily require a bitrate of 30 Mbps or higher. Refer to [\[PCC\]](#) for more details.

## 2.2. Path Requirements

The bitrate requirements in [Section 2.1](#) are per end-user actively consuming a media feed, so in the worst case, the bitrate demands can be multiplied by the number of simultaneous users to find the bandwidth requirements for a router on the delivery path with that number of users downstream. For example, at a node with 10,000 downstream users simultaneously consuming video streams, approximately up to 80 Gbps would be necessary in order for all of them to get 1080p resolution at 60 fps.

However, when there is some overlap in the feeds being consumed by end users, it is sometimes possible to reduce the bandwidth provisioning requirements for the network by performing some kind of



replication within the network. This can be achieved via object caching with delivery of replicated objects over individual connections, and/or by packet-level replication using multicast.

To the extent that replication of popular content can be performed, bandwidth requirements at peering or ingest points can be reduced to as low as a per-feed requirement instead of a per-user requirement.

### **2.3. Caching Systems**

TBD: pros, cons, tradeoffs of caching designs at different locations within the network?

Peak vs. average provisioning, and effects on peering point congestion under peak load?

Provisioning issues for caching systems?

### **2.4. Predictable Usage Profiles**

Historical data shows that users consume more video and videos at higher bitrates than they did in the past on their connected devices. Improvements in the codecs that help with reducing the encoding bitrates with better compression algorithms could not have offset the increase in the demand for the higher quality video (higher resolution, higher frame rate, better color gamut, better dynamic range, etc.). In particular, mobile data usage has shown a large jump over the years due to increased consumption of entertainment as well as conversational video.

TBD: insert charts showing historical relative data usage patterns with error bars by time of day in consumer networks?

Cross-ref vs. video quality by time of day in practice for some case study? Not sure if there's a good way to capture a generalized insight here, but it seems worth making the point that demand projections can be used to help with e.g. power consumption with routing architectures that provide for modular scalability.

### **2.5. Unpredictable Usage Profiles**

Although TCP/IP has been used with a number of widely used applications that have symmetric bandwidth requirements (similar bandwidth requirements in each direction between endpoints), many widely-used Internet applications operate in client-server roles, with asymmetric bandwidth requirements. A common example might be an HTTP GET operation, where a client sends a relatively small HTTP GET request for a resource to an HTTP server, and often receives a





significantly larger response carrying the requested resource. When HTTP is commonly used to stream movie-length video, the ratio between response size and request size can become quite large.

For this reason, operators may pay more attention to downstream bandwidth utilization when planning and managing capacity. In addition, operators have been able to deploy access networks for end users using underlying technologies that are inherently asymmetric, favoring downstream bandwidth (e.g. ADSL, cellular technologies, most IEEE 802.11 variants), assuming that users will need less upstream bandwidth than downstream bandwidth. This strategy usually works, except when it does not, because application bandwidth usage patterns have changed.

One example of this type of change was when peer-to-peer file sharing applications gained popularity in the early 2000s. To take one well-documented case ([\[RFC5594\]](#)), the Bittorrent application created "swarms" of hosts, uploading and downloading files to each other, rather than communicating with a server. Bittorrent favored peers who uploaded as much as they downloaded, so that new Bittorrent users had an incentive to significantly increase their upstream bandwidth utilization.

The combination of the large volume of "torrents" and the peer-to-peer characteristic of swarm transfers meant that end user hosts were suddenly uploading higher volumes of traffic to more destinations than was the case before Bittorrent. This caused at least one large ISP to attempt to "throttle" these transfers, to mitigate the load that these hosts placed on their network. These efforts were met by increased use of encryption in Bittorrent, similar to an arms race, and set off discussions about "Net Neutrality" and calls for regulatory action.

Especially as end users increase use of video-based social networking applications, it will be helpful for access network providers to watch for increasing numbers of end users uploading significant amounts of content.

### **3. Adaptive Bitrate**

#### **[3.1. Overview](#)**

Adaptive BitRate (ABR) is a sort of application-level response strategy in which the receiving media player attempts to detect the available bandwidth of the network path by experiment or by observing the successful application-layer download speed, then chooses a video bitrate (among the limited number of available options) that fits within that bandwidth, typically adjusting as changes in available



bandwidth occur in the network or changes in capabilities occur in the player (such as available memory, CPU, display size, etc.).

The choice of bitrate occurs within the context of optimizing for some metric monitored by the video player, such as highest achievable video quality, or lowest rate of expected rebuffering events.

### **3.2. Segmented Delivery**

ABR playback is commonly implemented by video players using HLS [[RFC8216](#)] or DASH [[DASH](#)] to perform a reliable segmented delivery of video data over HTTP. Different player implementations and receiving devices use different strategies, often proprietary algorithms (called rate adaptation or bitrate selection algorithms), to perform available bandwidth estimation/prediction and the bitrate selection. Most players only use passive observations, i.e., they do not generate probe traffic to measure the available bandwidth.

This kind of bandwidth-measurement systems can experience trouble in several ways that can be affected by networking design choices.

#### **3.2.1. Idle Time between Segments**

When the bitrate selection is successfully chosen below the available capacity of the network path, the response to a segment request will typically complete in less absolute time than the duration of the requested segment. The resulting idle time within the connection carrying the segments has a few surprising consequences:

- \* Mobile flow-bandwidth spectrum and timing mapping.
- \* TCP slow-start when restarting after idle requires multiple RTTs to re-establish a throughput at the network's available capacity. On high-RTT paths or with small enough segments, this can produce a falsely low application-visible measurement of the available network capacity.

A detailed investigation of this phenomenon is available in [[NOSSDAV12](#)].

#### **3.2.2. Head-of-Line Blocking**

In the event of a lost packet on a TCP connection with SACK support (a common case for segmented delivery in practice), loss of a packet can provide a confusing bandwidth signal to the receiving application. Because of the sliding window in TCP, many packets may be accepted by the receiver without being available to the application until the missing packet arrives. Upon arrival of the



one missing packet after retransmit, the receiver will suddenly get access to a lot of data at the same time.

To a receiver measuring bytes received per unit time at the application layer, and interpreting it as an estimate of the available network bandwidth, this appears as a high jitter in the goodput measurement.

Active Queue Management (AQM) systems such as PIE [[RFC8033](#)] or variants of RED [[RFC2309](#)] that induce early random loss under congestion can mitigate this by using ECN [[RFC3168](#)] where available. ECN provides a congestion signal and induce a similar backoff in flows that use Explicit Congestion Notification-capable transport, but by avoiding loss avoids inducing head-of-line blocking effects in TCP connections.

### **[3.3.](#) Unreliable Transport**

In contrast to segmented delivery, several applications use UDP or unreliable SCTP to deliver RTP or raw TS-formatted video.

Under congestion and loss, this approach generally experiences more video artifacts with fewer delay or head-of-line blocking effects. Often one of the key goals is to reduce latency, to better support applications like videoconferencing, or for other live-action video with interactive components, such as some sporting events.

Congestion avoidance strategies for this kind of deployment vary widely in practice, ranging from some streams that are entirely unresponsive to using feedback signaling to change encoder settings (as in [[RFC5762](#)]), or to use fewer enhancement layers (as in [[RFC6190](#)]), to proprietary methods for detecting quality of experience issues and cutting off video.

## **[4.](#) Doc History and Side Notes**

Note to RFC Editor: Please remove this section before publication

TBD: suggestion from mic at IETF 106 (Mark Nottingham): dive into the different constraints coming from different parts of the network or distribution channels. (regarding questions about how to describe the disconnect between demand vs. capacity, while keeping good archival value.) [https://www.youtube.com/watch?v=4\\_k340xT2jM&t=13m](https://www.youtube.com/watch?v=4_k340xT2jM&t=13m)

TBD: suggestion from mic at IETF 106 (Dave Oran + Glenn Deen responding): pre-placement for many use cases is useful-distinguish between live vs. cacheable. "People assume high-demand == live, but not always true" with popular netflix example.



(Glenn): something about latency requirements for cached vs. streaming on live vs. pre-recorded content, and breaking requirements into 2 separate charts. also: "Standardized ladder" for adaptive bit rate rates suggested, declined as out of scope.

[https://www.youtube.com/watch?v=4\\_k340xT2jM&t=14m15s](https://www.youtube.com/watch?v=4_k340xT2jM&t=14m15s)

TBD: suggestion at the mic from IETF 106 (Aaron Falk): include industry standard metrics from citations, some standard scoping metrics may be already defined. [https://www.youtube.com/watch?v=4\\_k340xT2jM&t=19m15s](https://www.youtube.com/watch?v=4_k340xT2jM&t=19m15s)

## **5. IANA Considerations**

This document requires no actions from IANA.

## **6. Security Considerations**

This document introduces no new security issues.

## **7. Acknowledgements**

Thanks to Mark Nottingham, Glenn Deen, Dave Oran, Aaron Falk, Kyle Rose, and Leslie Daigle for their very helpful reviews and comments.

## **8. Informative References**

- [CVNI] Cisco Systems, Inc., "Cisco Visual Networking Index: Forecast and Trends, 2017-2022 White Paper", 27 February 2019, <<https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html>>.
- [DASH] "Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats", ISO/IEC 23009-1:2019, 2019.
- [MSOD] Akamai Technologies, Inc., "Media Services On Demand: Encoder Best Practices", 2019, <<https://learn.akamai.com/en-us/webhelp/media-services-on-demand/media-services-on-demand-encoder-best-practices/GUID-7448548A-A96F-4D03-9E2D-4A4BBB6EC071.html>>.
- [NOSSDAV12] al., S.A.e., "What Happens When HTTP Adaptive Streaming Players Compete for Bandwidth?", June 2012, <<https://dl.acm.org/doi/10.1145/2229087.2229092>>.





- [PCC] al., S.S.e., "Emerging MPEG Standards for Point Cloud Compression", March 2019, <<https://ieeexplore.ieee.org/document/8571288>>.
- [RFC2309] Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J., and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", [RFC 2309](#), DOI 10.17487/RFC2309, April 1998, <<https://www.rfc-editor.org/info/rfc2309>>.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", [RFC 3168](#), DOI 10.17487/RFC3168, September 2001, <<https://www.rfc-editor.org/info/rfc3168>>.
- [RFC5594] Peterson, J. and A. Cooper, "Report from the IETF Workshop on Peer-to-Peer (P2P) Infrastructure, May 28, 2008", [RFC 5594](#), DOI 10.17487/RFC5594, July 2009, <<https://www.rfc-editor.org/info/rfc5594>>.
- [RFC5762] Perkins, C., "RTP and the Datagram Congestion Control Protocol (DCCP)", [RFC 5762](#), DOI 10.17487/RFC5762, April 2010, <<https://www.rfc-editor.org/info/rfc5762>>.
- [RFC6190] Wenger, S., Wang, Y.-K., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", [RFC 6190](#), DOI 10.17487/RFC6190, May 2011, <<https://www.rfc-editor.org/info/rfc6190>>.
- [RFC8033] Pan, R., Natarajan, P., Baker, F., and G. White, "Proportional Integral Controller Enhanced (PIE): A Lightweight Control Scheme to Address the Bufferbloat Problem", [RFC 8033](#), DOI 10.17487/RFC8033, February 2017, <<https://www.rfc-editor.org/info/rfc8033>>.
- [RFC8216] Pantos, R., Ed. and W. May, "HTTP Live Streaming", [RFC 8216](#), DOI 10.17487/RFC8216, August 2017, <<https://www.rfc-editor.org/info/rfc8216>>.

#### Authors' Addresses

Jake Holland  
Akamai Technologies, Inc.  
150 Broadway  
Cambridge, MA 02144,



United States of America

Email: [jakeholland.net@gmail.com](mailto:jakeholland.net@gmail.com)

Ali Begen  
Networked Media  
Turkey

Email: [ali.begen@networked.media](mailto:ali.begen@networked.media)

Spencer Dawkins  
Tencent America LLC  
United States of America

Email: [spencerdawkins.ietf@gmail.com](mailto:spencerdawkins.ietf@gmail.com)

