

White paper:
HTTP STREAMING
– With Varnish Streaming Server

Tackle the challenges of HTTP streaming delivery & performance with Varnish Software

Introduction

The ubiquity of HTTP streaming poses challenges but even more opportunities, of which companies across industries have already taken advantage. As streaming becomes the norm, and video comes to make up the majority of internet traffic, these challenges become more pressing and specific.

Varnish offers a solution that addresses the future-facing challenges of HTTP streaming delivery management and builds on the natural strengths of Varnish: performance and flexibility, both of which are major concerns as the future of streaming unfolds.


What you will learn:

- Consumer trends that are accelerating the migration to HTTP-based viewing.
- The characteristics of a caching layer that is ready for anything across live, VoD, popular and long-tail content.
- How to introduce unprecedented storage and throughput capacity using fewer servers.
- How to protect the origin/backend from the threat of overload and achieve best-in-class reliability.
- The considerations that are driving more streaming service providers toward a private, DIY CDN model.

The current state of media streaming is constant growth and change, underpinned by the near-ubiquity of and demand for high-performance streaming across devices. This includes both free and paid streaming services, ranging from on-demand YouTube videos, TV/media outlets streaming content globally 24/7 and their VoD offshoots, radio stations and premium content streaming from Spotify to Netflix and competitive contemporaries. Media streaming also enables everything from online conference calls/meetings, webcasts and live events and e-learning/distance education. Live video has spread to all kinds of

industries, such as healthcare, education and recruitment, among many others¹. The potential for mass streaming and its applications have been realized **everywhere**, as technologies have converged to make streaming both scalable and seamless for the end-user.

Delivering reliable, high-performance streaming, particularly of live, Over-the-top (OTT) and video on demand (VoD) media, is at the heart of the challenge companies face.



Given its universality, HTTP has been instrumental in the evolution of video streaming

Sources

[1] <http://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=121376>

From a trickle to a stream: Live and on-demand HTTP streaming

In the early days of media streaming - from the first live stream of a major league baseball game in 1995 - we did not necessarily foresee the long road to achieving a smooth and user-friendly streaming experience. Fast-forward more than two decades, and we are still working toward offering the optimal everywhere, anytime streaming experience.

By far the most successful application on the internet - the web - relies on the HTTP protocol for transport. Why? Other protocols often run into problems crossing firewalls and routers. This, combined with its simplicity and the extensive tooling, has made HTTP into a generic protocol that has found uses far outside its original designation. Given its universality, HTTP has been instrumental in the evolution of video streaming.

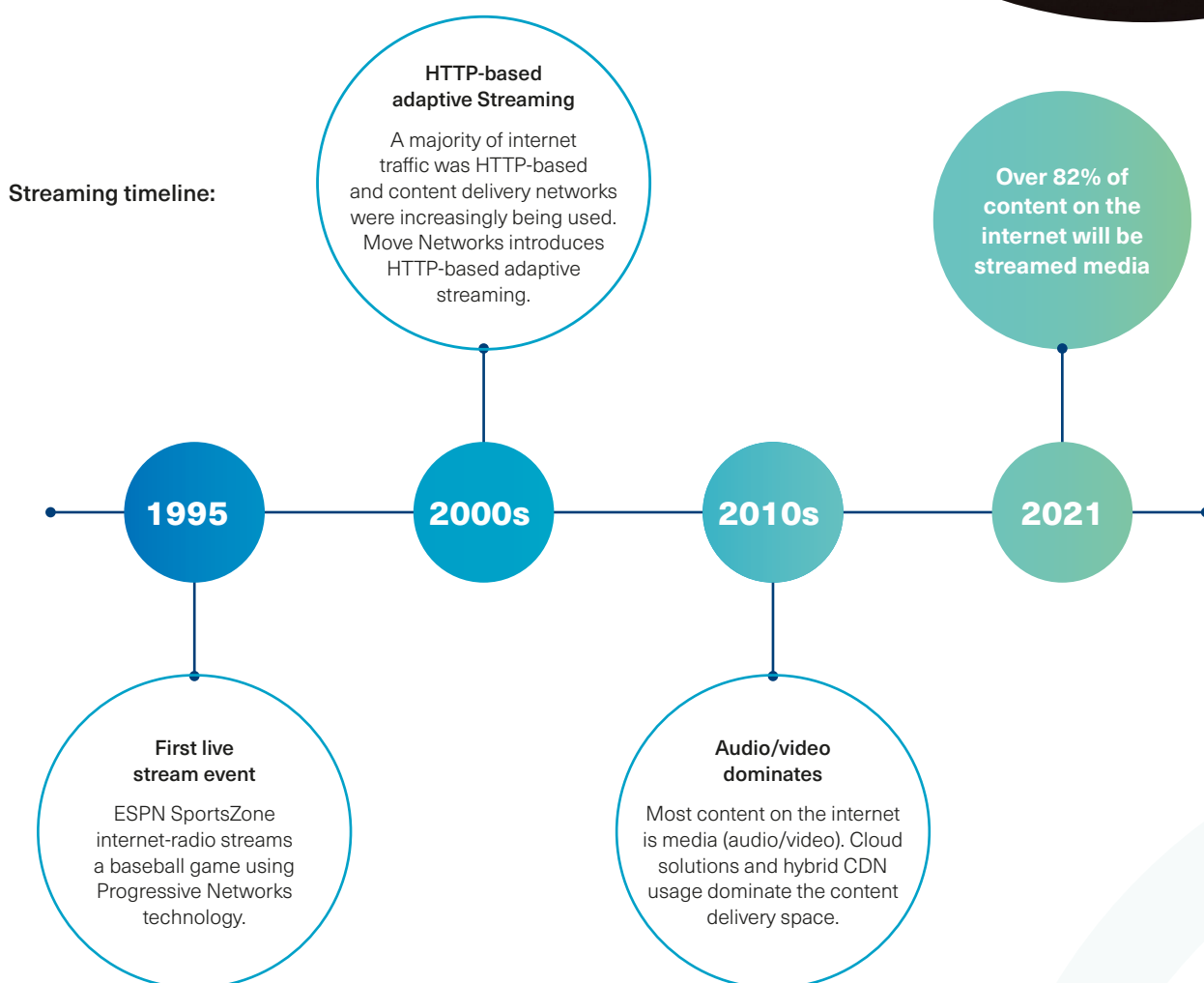


Diagram 1: Two decades in, when streamed media content dominates the internet, we are still working towards an optimal streaming experience.

Gentle HTTP waves

The HTTP protocol has been used to send media files in chunks, interacting with the streaming media player application to gain insight into network conditions. That is, only sending appropriately sized file chunks to suit the available network bandwidth, i.e. **adaptive streaming**. In this way, some of the fundamental roadblocks to mass streaming have been averted: endless buffering and connectivity problems could be circumvented using content distribution over standard HTTP (via content delivery networks) and caching.

This is not the end of the story and only illustrates how some of the platform-agnostic building blocks came to form the foundation of what we know as streaming today. There are still growing pains, including numerous shortcomings in terms of efficiency and speed.

To rectify these shortcomings, a chorus of competing HTTP-based transport layers emerged, such as, Microsoft (Smooth Streaming), Apple (HTTP Live Streaming or HLS) and Adobe (HTTP Dynamic Streaming or HDS), all of which are based on the simple principle of splitting H.264 video up into short segments and sending each of these in a HTTP response. To avoid the pitfalls of competitive infighting and incompatible protocols, the aim of interoperability has driven development.

This is essentially where we are with streaming today.

Streaming challenges

The real challenges for streaming revolve around quality and volume. With more connected devices, connectivity everywhere, and most of all, an exponentially growing mass of multimedia content, users demand high-quality, fast, seamless delivery and complete, integrated digital experiences.

Some content delivery challenges will be out of your hands. For example, most companies are not going to be able to influence things like network bandwidth limitations, which still hold back quality improvements. To some degree streaming challenges can be met with developments to video coding and compression efficiency. Focusing on one key issue - performance - can help mitigate what you can't control.

Tackling practical challenges

There will always be aspects of content delivery that are out of your control, which makes it all the more important to control the aspects you can. In our work with industry leaders across multiple sectors, we have learned that HTTP streaming poses similar challenges across the board, many of which are interrelated, and all of which Varnish can solve or relieve:

- Software that does not scale to traffic levels and more generally managing unpredictable demand
- Slow or latent content delivery issues
- Lack of resilience
- Lack of flexibility or adaptability in streaming software solutions
- Origin shield/backend protection
- No transparency, e.g. "black box" solutions that take flexibility and control out of your hands and possibly tack on extra costs with every feature added
- Security



The flexibility Varnish is known for makes it a perfect fit for different use cases

Technology solutions: What Varnish Streaming Server offers

Bring on the flood

HTTP is what Varnish was built for. To tackle the practical challenges listed above, Varnish's HTTP native design gives you a solution that can be used and deployed flexibly to meet your individual streaming needs.

The flexibility Varnish is known for makes it a perfect fit for different use cases. For streaming (see diagram 2), Varnish should be implemented as close as possible to the end-user, where it acts as cache and at the same time protects the underlying origin. The idea is simple: a camera records and the resulting video is processed and distributed. But what exactly do we mean when we repeat "flexibility" like a mantra? It's much more than a buzzword. Varnish Streaming Server can be used in multiple ways to manage the hardest and most persistent challenges of streaming:

- as a standalone component for serving video and an efficient way to scale out your platform
- as an efficient storage platform for serving massive amounts of content/traffic from a single location efficiently (high-volume VoD)
- as an "origin shield" when used with CDNs to protect your backend and ultimately the most valuable asset of all: your content

- as a complex policy and logic engine that enables things like authorization and authentication, rate limiting or geographically restricted content to ensure that your content can be accessed as, when and by whom you want it to be accessed
- as a performance engine for ensuring that you deliver content fast, reducing latency by putting a pre-warmed cache in place/pre-fetch technology

All of these flexible functions can be tuned to your own specific setup and needs, and Varnish is equipped with resilience and security measures to ensure that the solutions you put in place continue to work under virtually any conditions.

Serve video and scale your platform: Varnish Streaming Server

Built for high-traffic, high-volume, dynamic content, Varnish Streaming Server delivers scale, speed, performance and stability. Able to handle all kinds of files, all levels of traffic, Varnish helps you be ready for anything. But how does Varnish Streaming Server enable the solutions listed above?

Live streaming

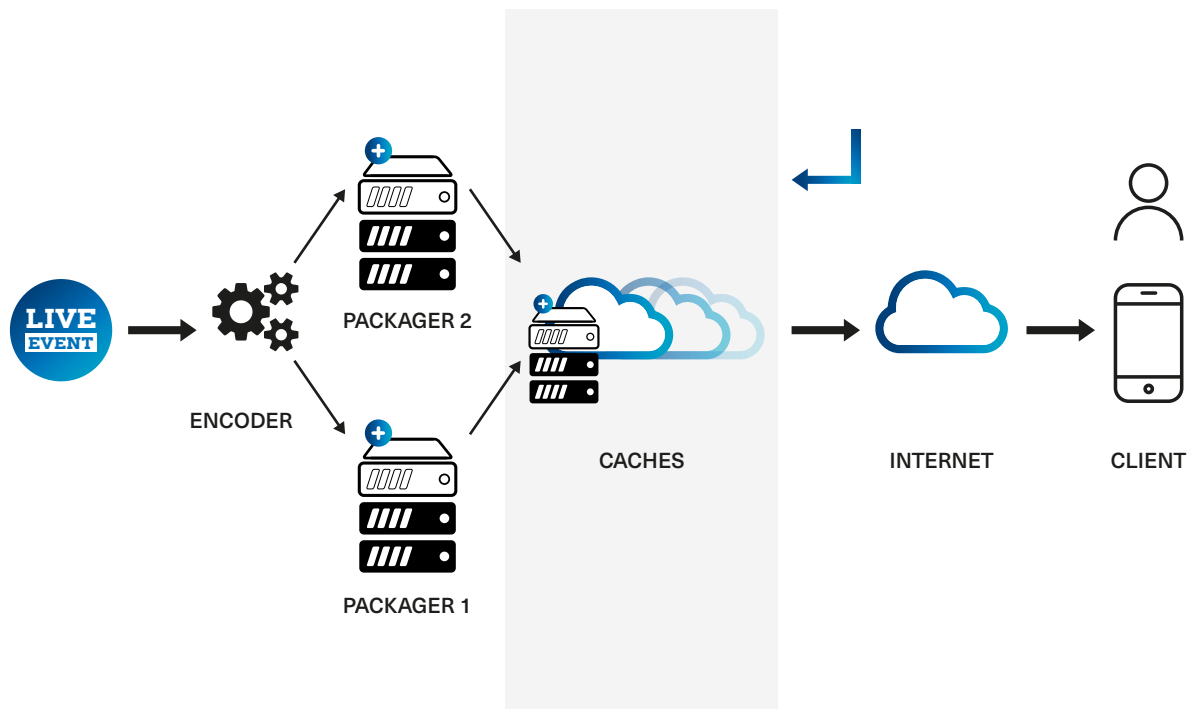


Diagram 2: Live streaming scenario leveraging Varnish Streaming Server. The scenario will be quite similar for VoD and OTT deployments.

Protecting your backends

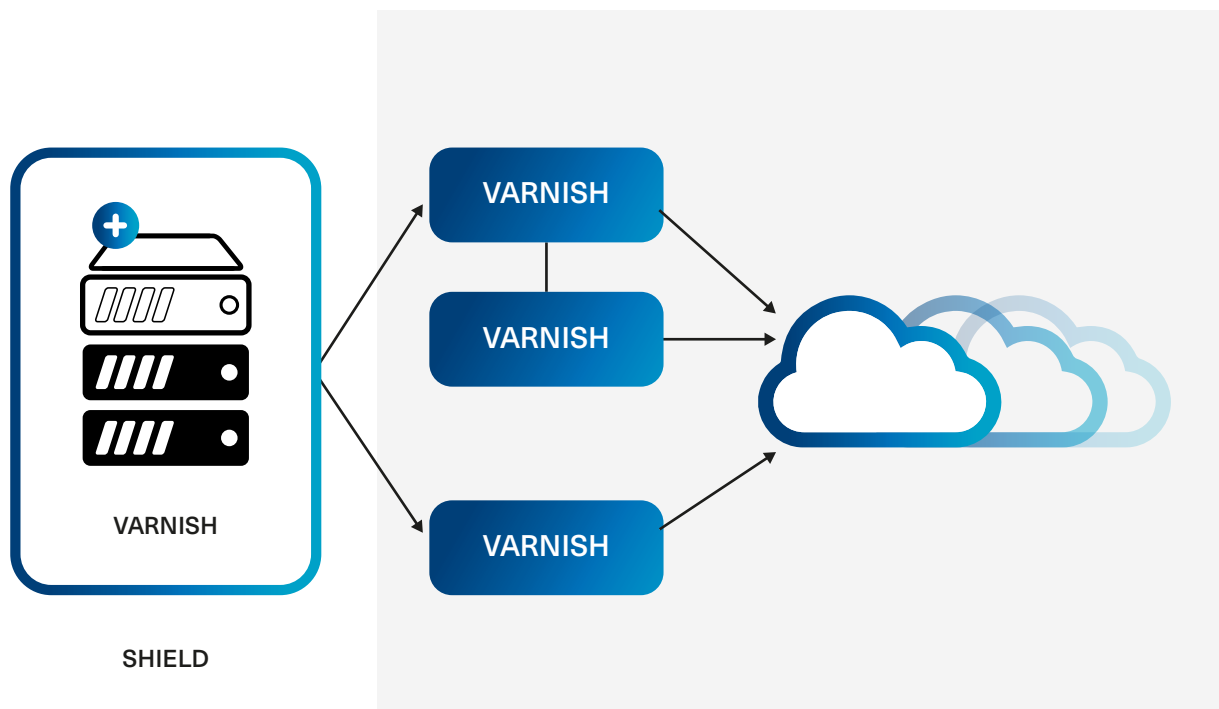


Diagram 3: Your Varnish layer will exist as close as possible to the origin servers and this layer will protect backends and serve as much content as possible to other Varnish nodes.

Varnish Streaming Server features

Store and serve massive amounts of content

One continuous challenge of VoD streaming is the issue of growing amounts of content - vast libraries and archives of media that must be stored and easily, quickly accessed when requested. Varnish Massive Storage Engine (MSE) provides a significant amount of local cache storage. Caching live streaming is easy, thanks to the short lifetime of content, but caching an entire catalogue of VoD, your storage solution needs to scale to tens and hundreds of terabytes, allowing to cache everything you want, and not just the hottest content.

- Built for up to 100+ terabytes of storage on each node
- Fragmentation-proof allocation algorithm
- Higher cache hit rates thanks to a better eviction policy
- Persistent datastore so you don't lose your entire cache on restart

Protect your backend: Origin shield

Backend overload is bad news, but with Varnish, you can shield your origin from excess traffic hitting it and achieve solid streaming performance and a reasonable cache-hit rate, even with live streams. Clearly when you plan your architecture, you will build in cache replication to ensure that you're never left with a single point of failure. Also,

because all of this traffic will pass through Varnish, you are adding an extra layer of security.

Normally, networks of CDNs turn to the origin to fetch content, and the pressure of an influx of unexpectedly high traffic or an overload drives origin servers into the ground. Varnish acts as a content replication engine at the same time as serving as a protective layer against these traffic floods, acting as a tier of caches - horizontal scaling in front of the traffic. Pressure on the origin is reduced and the reliability and resilience of your streaming service is secured.

Sources

[2] Keyed-hash message authentication code

Control logic and policy:

Varnish Configuration Language (VCL)

VCL is a major key to the Varnish flexibility offers - giving you the ability to configure and control the logic governing your content. Because of this flexibility, you can, for example, do things like:

Geo-blocking at country or city level

When content needs to be restricted by geography, Varnish Streaming Server includes a GeoIP VMOD for limiting or restricting access by specific location. While Varnish does not have this functionality built in, it is just one of the many flexible additional modules (VMODs) that can be added thanks to the flexibility of VCL. By setting a header indication instructing that a geo-based limitation should be put in place, we can set up specific restrictions.

For example, allowing requests coming from France only is as easy as:

```
import geoip;

sub vcl_recv {
    if (geoip.country_code(client.
ip) != "FR") {
        return (synth(403, "Sorry,
unauthorized country"));
    }
}
```

Token access in any shape or form

Using VCL, it's easy to quickly prototype and test the logic of various authorization schemes. This is greatly helped by the digest vmod which provides a collection of hashing functions, enabling you to build and check HMAC² very simply.

For example, here, we expect the client to hash the requested URL using a "secretkey" as a cryptographic key and placing it in the "check" header:

```
import digest;

sub vcl_recv {
    if (req.http.check != digest.
hmac_sha256("secretkey", req.url)) {
        return (synth(403, "Sorry,
bad secret"));
    }
}
```

For the more complex cases, it is also possible to create a VMOD to improve readability. That is, VCL is an extremely powerful tool, but sometimes, things can get a bit convoluted, or simply limited. Those cases can be handled via VMODs (Varnish modules) - VCL MODULES (like geoip or digest) that mask the complexity, or tap into lower-level APIs.

Flexible rate limiting and abuse suppression

Designed exactly for these use cases, the throttle VMOD will act as a guardian, directly in VCL, keeping tabs on requests, letting you refuse the too-frequent occurrences.

Using the example from the README:

```
import vsthrottle;

if (vsthrottle.is_denied(client.ip,
3, 1s)) {
    return (synth(429, "Too Many
Requests"));
}
```

You can limit a single client to download at an unfair rate. You can specify multiple limits to really fine-tune the access patterns. An example of this kind of use is, for example, when you have a VoD library available for streaming on mobile, and want to rate limit how much bandwidth can be used.

Also note that because the first argument is a string, it's possible to filter IPs, as well as URLs, countries, and more, including combinations of those parameters.

Built for high-traffic,
high-volume, dynamic
content, Varnish Streaming
Server delivers scale,
speed, performance
and stability





Latency/performance lags are still end-users' number one complaint about streaming media. Prefetching content, keeping your cache warm, is one potential way to boost streaming performance and make the streaming experience smoother

**Boost performance by anticipating the future:
Prefetch to keep your cache warm**

Latency/performance lags are still end-users' number one complaint about streaming media. Prefetching content, keeping your cache warm, is one potential way to boost streaming performance and make the streaming experience smoother. With the VMOD-http, you can act predictively, anticipating what the next logical chunks of content a user client will request, letting you prefetch content.

What exactly does this mean? You are loading data into your edge server's cache before it is even requested - making it available and ready to serve immediately

when the client requests it. When the predictive prefetch is accurate, latency should theoretically be reduced because the time lapse between roundtrip content request and return to and from the backend is eliminated. This should also save some strain on the origin server.

With live video, of course, there are only a limited number of chunks available to prefetch because the live event is happening in real-time. With VoD, of course, prefetching has more chunks to work with.

While the VMOD-http functionality extends beyond prefetching, for streaming prefetching is an important use. The VMOD lets you execute HTTP requests directly from VCL and supports synchronous and asynchronous

operations and connection pooling for higher efficiency. With async requests, you fire and forget HTTP requests without waiting for responses, i.e. VCL processing happens without waiting for responses, which speeds things up. How this fits into prefetching is that an async operation fetches the next logical object into cache: the next logical URL to be requested based on the current stream is generated by incrementing the number sequence of the original one. This may sound more complicated than it is when it boils down to the fact that with just six lines of VCL you can warm your cache with the next video segment and keep it ready to continue delivering content:

```
vcl 4.0;
import http;

sub vcl_recv{
    if (req.url ~"^/live/") {
        http.init(0);
        http.req_copy_headers(0);
        http.req_set_
method(0,"HEAD");
        http.req_set_url(0, http.
prefetch_next_url());
        http.req_send_and_finish(0);
    }
}
```

The idea, of course, is that with prefetch capabilities, you can gain a performance edge and better satisfy end-user requests.

Transparency - what you see is what you get (and more)

Many streaming solutions on the market today function something like a "black box". They do not offer users any insight into how the solution works or any flexibility to adapt or configure it to their own needs. These kinds of solutions are fine for basic streaming, but most deployments, according to the majority of our customers who have highlighted this as an issue they've encountered repeatedly, require more oversight and the ability to make changes and customizations to the streaming platform. These black box solutions are less flexible or even impossible to change - and usually come with unforeseen, unpredictable additional costs as necessary features were added to the basic 'black box solution'. This could also add delays to deployment and make vendor lock-in more likely.

Another feature that boosts transparency is the available logging facilities, which allow for very easy debugging. Varnish, compared to other caching solutions, has a very verbose log output. It writes to a circular buffer called Varnish Shared Memory Log (VSL). To avoid filling up your disk with information, it is not persistent by default. Making it persistent is an option VSL. All the Varnish tools read from the VSL, process the data and present it to the user, i.e. varnishstat and varnishlog are the most used. The former displays statistics from a running varnishd instance, while the latter shows varnish logs.

Sources

[3] To read a full overview of Varnish Total Encryption: <https://info.varnish-software.com/blog/introducing-varnish-total-encryption>

All of these issues can create roadblocks down the line as streaming changes or when it comes time to restructure or redesign a company's site architecture. Not having control over or ability to configure aspects of the streaming solution creates a number of unneeded question marks.

Security

Secure connections with TLS/SSL safeguard the data we're sending; this has not always been key for streaming. But it has become standard practice to stream over SSL, and SSL is required to stream in certain markets and on certain sites, such as Facebook and Google. SSL and non-SSL traffic cannot be mixed. For example, if you want to run a video on nytimes.com or any other site, which is served over SSL, your video needs to be served over SSL. Also, looking at sites like Netflix and YouTube, content is all served over SSL, as major browsers have begun showing warnings for non-SSL traffic. In line with this, the video industry as a whole is in the process of standardizing on SSL delivery.

With the European GDPR data processing and storage rules, security is more essential than ever. Also, more and more data are being delivered on mobile - mobile connections usually deliver content faster if the data are encrypted (even if it has not been standard practice to encrypt video in the past).

Varnish Total Encryption³ is another security measure that will encrypt the entire cache as a means to safeguard cache data against bugs and vulnerabilities, such as Meltdown and Spectre and Cloudbleed respectively. However, it does more than just make your cache unreadable to unwanted eyes. Varnish Total Encryption prevents an entire class of cache vulnerability, the cache leak. Caches are designed for speed and efficiency, and are designed to be fast and open and not secured and locked down. With Varnish Total Encryption, assigning each and every cache object its own unique AES256 encryption key provides the lockdown the cache historically lacked.



Who should use Varnish Streaming Server?

Everyone is cutting the cord in one way or another, creating use cases in multiple sectors and industries.

With these considerations in mind, virtually anyone will benefit from using Varnish as their solution for VoD, OTT and live streaming, including:

CDNs

Content providers

- Broadcast networks/channels/telecoms companies (who do not want to be seen only as the “dumb pipe” supplying the bandwidth)
- Media outlets – TV, radio, online
- Streaming content sites/aggregated content, such as Dailymotion
- Sports associations (as mentioned, the NFL has not managed on its own on this front very well, while Major League Baseball was so technically adept that they have their own spin-off, in-house streaming media department⁴ that handles not only baseball but streaming digital distribution for a lot of other very big media names).
- Corporations, universities, healthcare organizations and other large institutions moving into live streaming of lectures and training, etc.

This is really only a sampling of who should be and can benefit from using Varnish for streaming.

Sources

[4] <http://www.theverge.com/2015/8/4/9090897/mlb-bam-live-streaming-internet-tv-nhl-hbo-now-espn>

Get your feet wet: Start HTTP streaming with Varnish Streaming Server

As the internet and the protocols governing it have matured, the “consumerization” of streaming multimedia, complete with stability, speed and high definition, has fueled progressive development, making scalable streaming mainstream. HTTP has been the backbone of what got us to where we are - and it will continue to be the standard through which we deploy technologies that deliver the ever-growing stream of multimedia content to ever-hungrier end-users.

We have helped our customers globally to build advanced, scalable and fast streaming solutions on their own terms through the whole lifecycle of the software: Design, feature development and enhancements, implementation and optimization. Varnish Streaming Server offers all the flexibility and performance to make the streaming experience high-performance, robust and efficient while giving end-users what they want and expect.

About Varnish® Software

Varnish® Software is a global pioneer in high performance digital content delivery. Powered by a uniquely flexible caching technology, Varnish® Software's solutions are indispensable common denominators among some of the world's most popular brands, such as Sky, Emirates and Tesla. Our solutions enable organizations worldwide to provide a superior user experience with fast, digital content delivery at any scale, while giving them the flexibility to maintain control over their content and make the technology their own.



Everyone is cutting the cord in one way or another, creating use cases in multiple sectors and industries. With these considerations in mind, virtually anyone will benefit from using Varnish as their solution for VoD, OTT and live streaming

www.varnish-software.com

Los Angeles - Paris - London
Stockholm - Singapore - Karlstad
Dusseldorf - Oslo - Tokyo



www.varnish-software.com