OPERATING EUROVISION AND EURORADIO

TR 068

CDN ARCHITECTURES DEMYSTFIED

AN INSIGHT INTO CDN APPROACHES EMPLOYED BY EBU MEMBERS

SOURCE: EBU Broadband Distribution Architectures Group

Geneva
June 2022

This page is intentionally left blank

Contents

Executive Summary 3

1. What is a CDN? 4

2. Different architectures for organising distribution 5
   - 2.1 Public/Commercial CDN 5
   - 2.2 Private CDN 6
   - 2.3 How to get content to the CDN 7

3. Approaches to deal with CDN architectures 10
   - 3.1 Server-side or client-side CDN switching 11
   - 3.2 Live vs On-demand 13
   - 3.3 Monitoring CDN services 13

4. CDN value-add services 14
   - 4.1 Packaging of the CDN 14
   - 4.2 Peer-to-Peer (P2P) 15

5. Pricing models and cost considerations 15
   - 5.1 Pricing models 15
   - 5.2 Switching costs 17
   - 5.3 Tender options 17

6. Conclusion 18

7. Glossary 18

8. Bibliography 19

Executive Summary

With streaming platforms gaining in popularity, Public Service Media organisations (PSMs) are expending a lot of effort to create their own “Over-The-Top" or “Direct-To-Consumer" propositions. The breadth of the content catalogue, the scale of the viewership and its expectation of a high-quality experience pose technical distribution challenges. To overcome these challenges, complex distribution chains are constructed. This report gives an insight into the architectures of Content Delivery Networks (CDNs) used to distribute streaming media over IP networks by EBU Members, as well as the choices that need to be made when choosing amongst distribution solutions.

The report introduces the concept of a CDN, then dives deeper into the differences between commercial and private CDNs. Thereafter the multi-CDN architecture is introduced, and the different strategies needed to balance traffic between them are laid out. There follows the description of some value-added CDN services that may be offered on top of the main data distribution service and pricing models and cost considerations are addressed. To round up the report, standardisation activities related to CDN services are reviewed and a conclusion is drawn.

CDN Architectures Demystified

An insight into Content Delivery Network approaches employed by EBU Members

EBU Committee	*First Issued*	*Revised*	*Re-issued*
TC	2022		

Keywords: CDN, Content Delivery Network, Distribution, Traffic pattern, Congestion, Network Saturation, Caching, Pop, Point of Presence, MPEG-DASH, HLS.

1. What is a CDN?

A Content Delivery Network (CDN) is a service that distributes digital content to viewers and listeners. In this report the focus lies on media (audiovisual content) although other files such as webpages, images and software can also be distributed using a CDN.

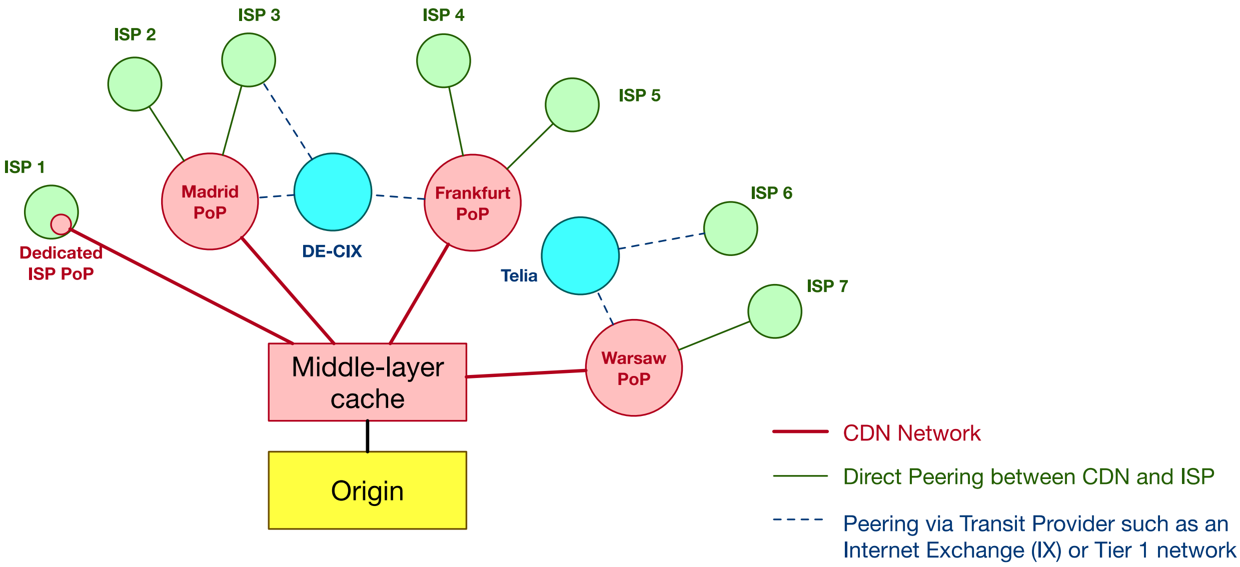
For the media industry, CDNs are an important element in the process of getting audiovisual content and services to users over the Internet. The delivery process encompasses three basic areas. The first area concerns the origin servers[1] (in this document also referred to as 'Origin') that expose the content of the media companies to the Internet. On the user side the second area concerns the access networks provided by ISPs. These can be fixed line networks such as DSL, cable or fibre, but mobile networks also fall under the class of access networks. In principle the content could be accessed by users directly on the Origin server, as they would access a web server for web pages, but with many concurrent users this would overload the Origin. To steer and control the traffic, CDNs are employed that link Origin and access networks. Moreover, a CDN can also be used for protection against DDOS attacks.

Due to the scale of the audience, and the significant bandwidth used by media content, a CDN is a necessity. PSMs can choose to build their own CDN, to use a commercial CDN, or to employ a combination of both. Except for making content available, a CDN can also help with providing a better-quality experience by having distribution nodes close to the customer, independent of geographic location or type of network (e.g., Wireless Broadband, Fibre or Cable/DOCSIS). Lastly, a CDN can scale to cope with the high demand resulting from peak-events such as breaking news, a season's finale, or high-value sporting events, that is atypical from day-to-day viewing.

The architecture of a CDN consists of interconnected PoPs (Points of Presence), which are local caching servers that are distributed throughout the world. They are typically interconnected through a private network or through Internet transit connections. Depending on the type of CDN, the PoPs either connect to a mid-layer cache or directly to the Origin (the originator of the content). Most PoPs are connected to major internet exchanges such as the AMS-IX in Amsterdam, and are located

[1] The function of an Origin server is further clarified in § 2.3.

throughout Europe, North America, and some key markets in Asia. Depending on the CDN, other parts of the world are also serviced, such as Latin America, Australia, and Africa. Sometimes CDNs co-locate distribution nodes directly on the network of the ISP (the last mile network), but most CDNs limit themselves to peering arrangements.



An example of what a CDN network could look like with a dedicated PoP at the ISP (ISP 1), a PoP with a direct peering relationship with an ISP (ISP 2), or a PoP with no direct connection to an ISP but reachable via a commercial Transit provider (ISP 6). Hybrid and combination scenarios are also possible.

Figure 1

2. Different architectures for organising distribution

A CDN is a necessity, however given their complexity and cost, many media companies choose to work with a commercial CDN rather than build their own. All Members work with at least one commercial CDN, and about half of the Members also build their own. Those that build their own typical use it as the main delivery method in their primary market(s), with commercial CDNs being used for overflow capacity (when their own CDN is saturated), or for those markets that are not their primary targets (e.g., citizens living abroad).

To summarise, PSMs can choose to

- Exclusively use commercial/public CDNs
- Exclusively use its own CDN (a theoretical scenario as now, no EBU Member exclusively uses its own CDN)
- Use a hybrid approach with a combination of a self-operated and commercial CDNs

2.1 Public/Commercial CDN

A public/commercial CDN is built by a company with capital investments to offer data distribution services to third parties. This company sells capacity on its network and controls the full operation, including the allocation of capacity. Their clients can only control the quality contractually via Service Level Agreements.

The companies that are active in the CDN market either have CDN services as their primary product, or they provide it as a service alongside or on top of other related services. For example, some CDNs started out as network and co-location providers, and due to the reach and interconnectedness of their networks, it made sense to offer CDN services as well. Others are cloud providers that have CDN services as part of their cloud offering (as a logical add-on to cloud computing or storage products). Lastly, some CDN providers have originated from product companies, and having developed caching software, decided to use this know-how to build their own distribution infrastructure (as an extension of the products they sell). When selecting a CDN it is useful to learn about their background, to better understand the value they bring to the table.

Independent of the CDN's background, the key considerations for selecting one as your primary distributor are (i) capacity, responsiveness, and availability on those networks that most of your users are connected to, (ii) price, and (iii) any additional services that might be useful.

2.2 Private CDN

A private CDN is one that you build and operate yourself, one that is built and operated for you by a third party or one that has been built for someone else but that you can also access.

The reasons you might choose a private CDN is because it is cheaper, it gives you better quality or flexibility compared to a commercial CDN, or alternatively it could be that commercial CDN capacity is insufficient to your needs in the market you operate.

Like Commercial CDNs, the three components of a Private CDN architecture are the Origin (servers) where the content is stored and made available to upstream components, the caching servers that the end-user connects to, and the interconnection point(s) where the media data is handed over to the user's ISP.

The goal of any CDN is ultimately to have content cached (temporarily stored) as 'close' as possible to the end-user. Close does not necessarily imply physical distance but rather, best performance in terms of throughput, latency and availability. Delivery from the storage/cache to the end user is called Edge distribution. In the case of commercial CDNs the customer does not know how this Edge distribution is organised. However, if a PSM decides to operate its own CDN network, it must address Edge distribution also. There are 3 main options as to how close to the end-user the content can be cached.

Different Private CDN models:

1. Edge distribution via peering

   In this scenario the caching servers are in your own data centre (or at a co-location facility), and you have a peering relationship with the user's ISP. The path from caching server to the user would be completely private and would not involve traversing the public internet. When available, fibre-rings can be utilised to provide connectivity between a broadcaster's gateways and ISP-networks.

2. Edge distribution via caching nodes at the ISP

   The caching servers are installed in the data centre of the ISP and have a direct connection to the ISP's network. The data transfer between the origin and the caching servers could go over the public internet, or via direct peering to the Origin.

   In many European markets this model has been gaining in popularity. It allows ISPs to free traffic from transit links, and it makes the routing of streaming traffic in the ISP's network more manageable.

3. Edge distribution via the ISP's own distribution network

   With the move to non-linear consumption of content and viewing on personal devices, ISPs adjusted their product mix to offer video via streaming. As a result, ISPs set up their own infrastructure to ensure the video data is efficiently distributed on their network.

   Some ISPs allow third-party media companies to use this internal distribution infrastructure. It could be part of a global distribution deal with the media company, or a means of generating extra income. This is effectively a commercial CDN, limited to the scope of the Access ISP's own network.

   For a PSM to use the ISP's distribution network, it would need to integrate its streaming infrastructure with that of the ISP. Functionality such as directing the viewer to the right distribution node, pushing content to the ISP's caching servers, or retrieving logs requires tight integration. To avoid having to re-implement this functionality for each ISP, the Streaming Video Alliance is working on a standard called Open Caching. An Edge distribution network that is compatible with Open Caching would have a standard way to integrate with the media company's streaming infrastructure. The three core components of the Open Caching standard are the 'Caching Node', operated by the ISP, the 'Request Router' to direct end-users to the correct caching nodes, and the 'Control Plane' to provision and manage the distribution infrastructure.

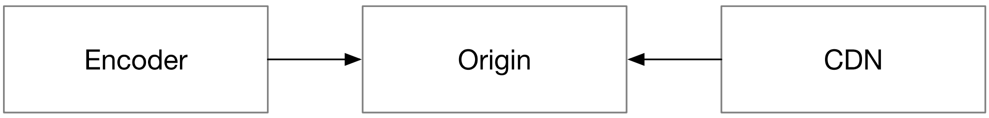
2.3 How to get content to the CDN

Above, it was stated that the data flows from the Origin (that stores the media data) to the caching nodes at the CDN. This is certainly true, but if you investigate the details, the reality is more nuanced and more complex than that.

Pull vs Push

Depending on which side initiates the data transfer, pull or push methodologies are identified. When a CDN receives a request for content that is not already in the CDN cache, the CDN requests (pulls) this content from the Origin; this is a pull-based flow. Normally CDNs cache content when it is requested a second time within a certain timeframe. First playouts will be redirected to the Origin.

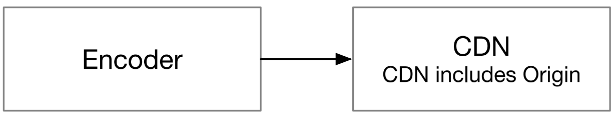
Schematically it looks like this:



The Encoder pushes content to the Origin, the CDN pulls content from the Origin

Figure 2

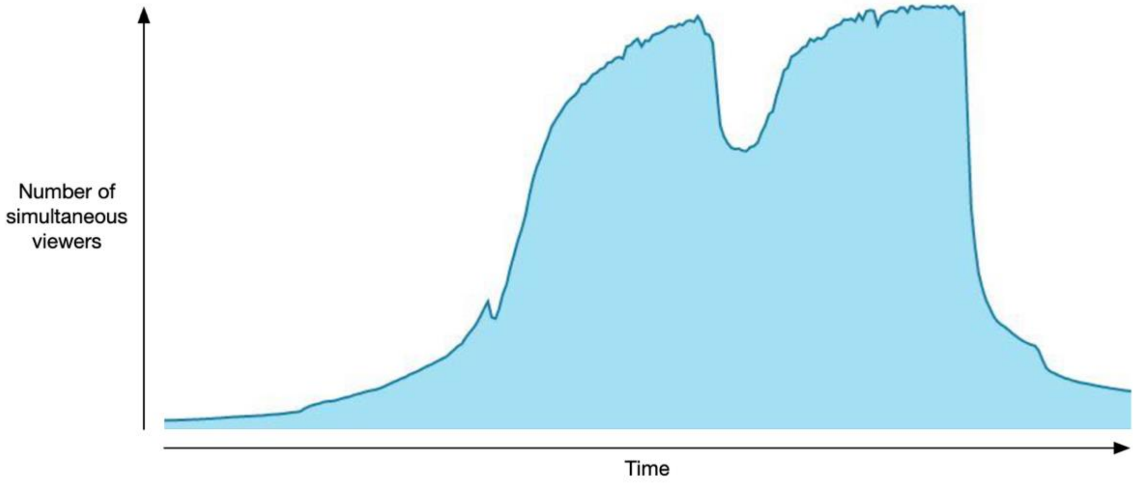
The alternative is that the Encoder pushes the content directly to the CDN, and the CDN stores the content and acts in effect as an Origin.



Push-based setup where the CDN acts both as the Origin and the distribution network

Figure 3

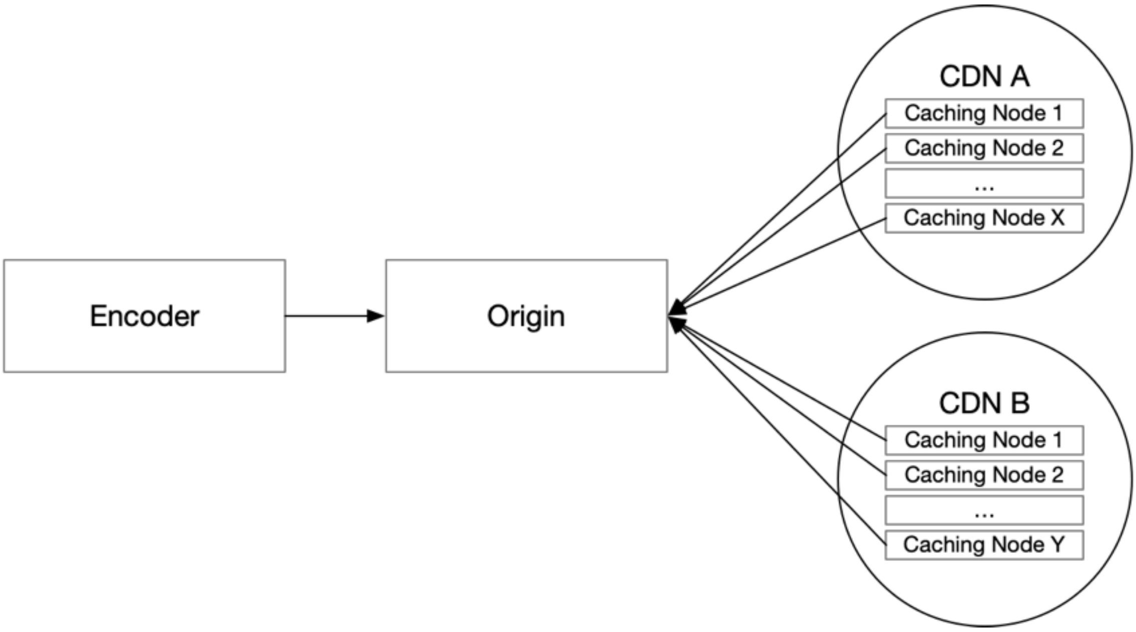
Thundering herd



An example of a traffic pattern during a football match with many people tuning in over a short timespan at the start of the match.

Figure 4

One of the downsides of a pull-based flow is that if there are many requests from the CDNs to the Origin, the Origin can get overloaded. This is known as the "thundering herd". This is mainly a problem for live streams where there are many people tuning in at once, at the start of a football match, for example. It can also, for example, happen when there is a misconfiguration when a unique query parameter is added to the streaming URLs, and the CDN is not configured to ignore this query parameter.

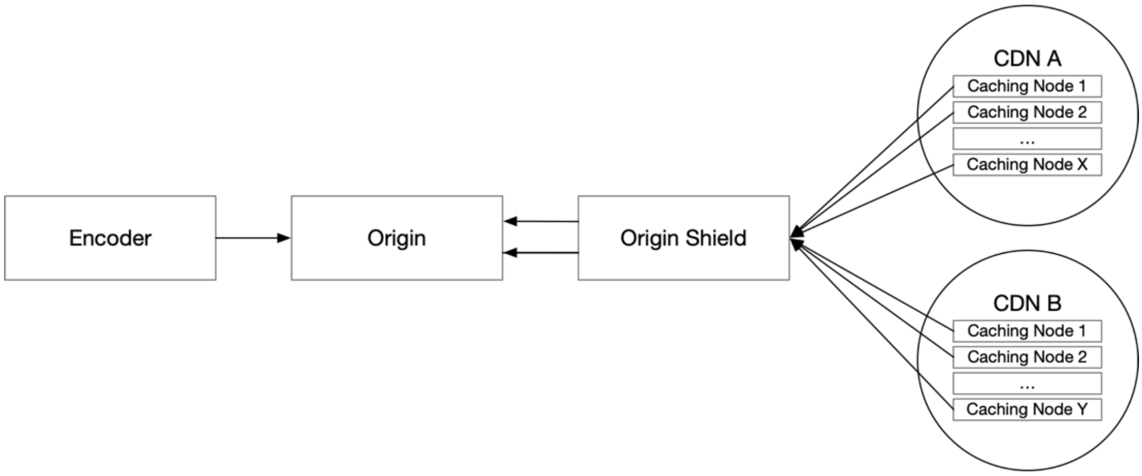


When there are many users tuning in at once and the content is not in the CDN cache, the Origin can get overloaded.

Figure 5

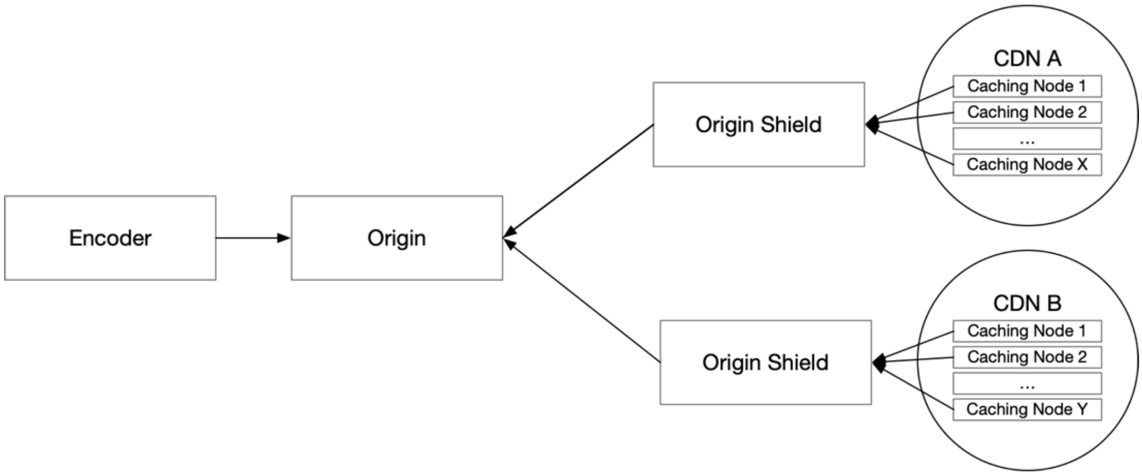
To protect the Origin, an Origin Shield can be implemented. The goal of the Origin Shield is to ensure that large numbers of requests will not bring down the complete distribution chain. This can be done by either limiting the number of requests going to the Origin, or by only letting those parts of the distribution chain fail that are experiencing the high load.

Potential Origin Shield scenarios



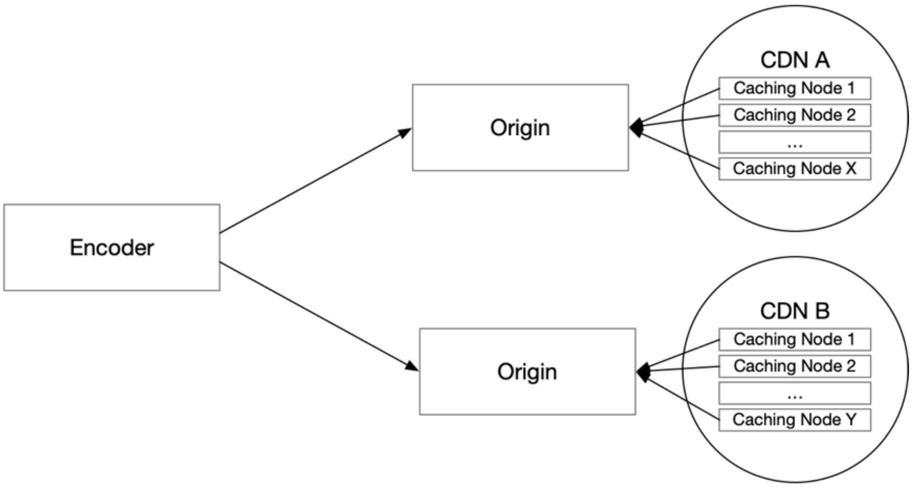
A central Origin Shield that responds to the requests from the CDN nodes.
There are still queries to the Origin, but far fewer.

Figure 6



An Origin Shield per CDN. Each Origin Shield still queries the Origin in response to requests from its CDN nodes, but much less aggressively than without an Origin Shield.

Figure 7



An Origin per CDN. The data is pushed to multiple Origins, one per CDN.
This ensures that if one CDN brings down its Origin, it will not impact the other CDNs.

Figure 8

When working with CDNs and Origin Shields, you need to consider the characteristics of the connections between the components. Low connection speed to, or limited server response from the Origin will block playout and caching requests. When, for example, the Origin Shield is located geographically in a different gateway than the CDN ingest point, this can result in longer return paths and this delay will cause long start-up times.

3. Approaches to deal with CDN architectures

In defining the right set-up for your CDN architecture there are several options to take into consideration. Most of the EBU Members apply a multi-CDN strategy in which, to optimise the delivery, traffic is load-balanced over the different, available distribution networks. There are options for performing the necessary CDN-switching either server-side or client-side or as a combination of the two. The differences between these options requires a bit more technically detailed explanation. CDN requirements specific for live and on-demand traffic are also discussed.

Multi-CDN strategies

The advantage of a multi-CDN strategy is that you do not have to rely on the functionalities, capacity, and availability of a single CDN. A multi-CDN setup enables you, for example, to use a cheaper CDN with limited capacity for day-to-day traffic and to add a more expensive CDN when you need to scale for large events. Furthermore, a multi-CDN setup increases the reliability of your media delivery by being able to re-direct users away from a CDN that is unavailable or performing badly.

There is also an important business-related advantage to choose a multi-CDN strategy. It will improve the negotiation position with CDN vendors. A potential downside is that the per- Gigabyte costs are higher when using multiple CDNs compared to committing all traffic to one CDN supplier. Also, implementing multi-CDN can introduce additional complexity and costs. In general, being able to switch easily between CDNs will reduce the risk of vendor lock-in and it creates a healthy competition between vendors in the market.

A survey amongst EBU Members showed that the following strategies are in use:

- Use own CDN till maximum capacity and commercial CDNs as overflow
- Use own CDN for on-demand, and commercial CDNs for live content
- Use own CDN for in-country requests, and commercial CDN for international viewers
- A fixed percentage request split between CDNs
- Different CDN for video and audio

In a multi-CDN scenario you need to specify parameters and rules to distribute traffic. It is possible to use static rules (e.g., 40% of requests to CDN A, 60% of requests to CDN B), or dynamic rules where changing conditions are used in real-time decision logic (e.g., send all traffic for this region to CDN A until start-up time moves above 1 second, then send next request to CDN B). Available parameters for making load balancing decisions include:

- Marginal price (i.e., per terabyte cost)
- Volume used this accounting period
- Current bandwidth usage
- Quality metrics (available bandwidth, latency, CDN availability, etc.)
- Geo-location
- ISP/Network specific
- Content based (e.g., live vs on-demand or video vs audio)
- Device capabilities

- …

3.1 Server-side or client-side CDN switching

The choice of CDN can be made server-side, in which the video player is redirected to the CDN of choice, based on rules such as described above. The other option involves client-side decisions in which the client has knowledge of the available options and runs client-side logic to select the best CDN at that moment in time. Both approaches have their pros and cons; a combination of both proves to deliver the most reliable environment.

In general, server-side switching provides a CDN recommendation based on a more holistic set of (real-time) data without individual implementation within the player. For example, by combining CDN server logs, QoE data from different video playout sessions and data probes in the networks with business rules. The disadvantage is the introduction of a single point of failure in the distribution chain.

On the other hand, the client-side approach does not depend on external decision logic, and it can operate independently. The biggest disadvantage of this approach is the fact that a player needs to be able to act alone on the (limited) information it has available.

Server-side DNS switching is used when there is no or limited option to run client-side logic (for example if you have no control over the player, such as via a smart speaker or older devices), or when the logic is relatively static, and you do not need to switch CDNs in a playout session. There is also an option to integrate server-side switching through a player API, which allows implementation of specific optimisations such as switching to another CDN during a session when this can avoid a playout interruption or can improve quality (switching to a higher bitrate) or for business reasons.

Client-side, the player has information about local conditions such as latency and throughput to the CDN and can make more informed decisions as to which CDN to pick. However, in the client there is no real-time information about the global CDN load or marginal pricing information, resulting in a decision that might not be optimal in aggregate.

The best option is a hybrid approach in which one uses the server-side approach to recommend CDNs at start-up of the content and then let the client control the rest of the playout session. For example, the server sends a pre-filtered list to the client with information such as the predicted availability of CDNs and lets the client make an informed decision combining the information from the server with its own playback statistics.

Diving a little deeper into the different options for server- and client-side approaches:

<u>Server-side options to consider</u>

On the server-side approach there are the following options:

1. Simple DNS based switching: The player receives a CDN playout location from a traffic manager or DNS-server.
   a. **Pro:** Simplicity and backwards device compatibility as you only rely on basic internet tools.
   b. **Con:** Treats all requests similarly and switching CDNs can be slow
2. Complex DNS based switching: The player receives a CDN playout location from a traffic manager through DNS-logic.

   a. **Pro:** Backwards device compatibility and allows user specific decisions based on rules.
   b. **Con:** Decision logic can become complex and switching CDNs can be slow

3. CMS based switching: The player receives CDN playout location through a CMS
   a. **Pro:** Flexibility and control. Allows for rule based CDN decisions per user. Backwards device combability.
   b. **Con:** Requires implementation of rules set in the CMS or site/application.

DNS or CMS based CDN switching is realized by using either a DNS Load Balancer or by providing the URL (or URN) through the CMS. This URL (or URN) is used as entry point for the stream requests. It provides the maximum backwards compatibility towards all devices, including older set-top boxes, HbbTVs, older internet radios, etc. Additionally, it can be used for any distribution, either to Web, Application or to device specific integrations.

Client-side options to consider

On the client-side approach there are the following options:

1. Simple manifest based switching
   a. **Pro:** Lowest complexity and supported in MPEG-DASH and Apple-HLS
   b. **Con:** Need to add this information to the manifest
2. More complex switching based on processing of local real-time QoE data
   a. **Pro:** Increased performance and reliability
   b. **Con:** Clients need to be able to both retrieve and use this data
3. API based switching: The client receives CDN playout location through interacting with an API
   a. **Pro:** Allows the introduction of server-side information
   b. **Con:** Client specific implementation

Manifest-based source switching relies on individual players' implementation of the appropriate specifications. It is independent of content packaging formats to supply a separate initial 'playlist' of multiple CDN sources (usually the manifests) to players at (e.g.,) the content mediation stage (i.e., when rights are checked, a CDN access token is issued, or a GeoIP check is performed), prior to a manifest or CMAF container being loaded.

The two main HTTP-based media packaging formats, MPEG-DASH and Apple HLS, support the supply of multiple content sources in content manifests:

- MPEG-DASH: multiple BaseURL support in MPD
- HLS: Redundant Streams in M3U.

The Common Media Application Format (CMAF) delegates to the individual stream manifests of the transfer formats in a CMAF presentation object, so in practice either the DASH or HLS manifests can implement multiple sources as above.

An important consideration is that Clients' implementations of failover behaviour can vary wildly as the packaging specifications are not explicit in this regard. Different players can behave very differently in both their initial and subsequent selection of alternative content sources, or in some cases can ignore the alternatives completely.

3.2 Live vs On-demand

In terms of network distribution and content preparation, requirements for live services are quite a bit more challenging than on-demand services.

In fact, user consumption patterns for on-demand services are quite predictable; daily consumption curves can easily and automatically be derived from the past, and the year-on-year traffic increase can be predicted. This allows a proper scheduling of the size of CDN resources in advance. Moreover, on-demand services do not have strict latency constraints, allowing for a more efficient video encoding process (e.g., multi-pass encoding, larger chunks), providing high quality pictures at low bitrate, so reducing network requirements in terms of throughput and storage.

Conversely, popular live events can produce very high traffic peaks concentrated in small time periods and affecting a single country or region, sometimes with little notice. These are the situations where the CDN backbone may become congested unless properly designed and oversized. Moreover, latency requirements on live services are more stringent, resulting in a less efficient video encoding process, generating higher bitrate streams to provide an adequate video quality. Additionally, shorter video chunks, needed for low-latency streams, cause a significant increase in the number of HTTP requests generated by users' players, which, in turn, require more demanding computational resources in the edge nodes.

Therefore, the design of CDN networks for OTT services should keep account of such specificities, differentiating between live VOD (and Video On Demand) services.

For instance, assuming reliance on a commercial CDN operator (but similar considerations can be derived for an owned CDN infrastructure), content assets relevant to on-demand services could be made available on the Origin server as encoded files, leaving to the CDN operator the task of pulling them, generating the ABR (Adaptive Bitrate) segments and distributing them to the edge nodes according to users' requests.

Live services could instead be treated differently; for instance, they could be transcoded and packaged in ABR format in-house, possibly in Low-Latency mode, and then pushed directly to the CDN network according to an efficient media ingest workflow (e.g., DASH-IF Live Media Ingest Protocol) to reduce the transfer latency as much as possible.

3.3 Monitoring CDN services

Since the performance and capacities of CDNs are constantly changing it makes sense to monitor the Quality of Experience and Quality of Service and adapt the CDN distribution based on these metrics. This information can be used to report the actual playout quality the audience experiences.

There are several commercial QoE/QoS solutions available that can be utilized to monitor the current state of the CDNs. They usually work through a player plugin that collects the performance metrics directly at the source and sends them to a backend for aggregation. These solutions can offer in-depth insights into the correlations between CDNs, user location, user ISP, playback device and playback apps for individual streaming issues.

As an addition or alternative, the status of the CDNs can also be monitored through probes. In this case dummy clients are distributed on the internet. These probes regularly request the streams, monitor the performance and send the metrics to a backend for aggregation. This solution provides information on the general performance of a CDN.

4. CDN value-add services

CDN and cloud vendors are increasingly offering additional media-related services in addition to content distribution capabilities, that can enhance and accelerate the adoption of an end-to-end media processing and distribution chain amongst PSMs and other content providers.

4.1 Packaging of the CDN

Using the on-the-fly packaging of a CDN is, like any other tool, something that is useful or not depending on the situation and the objective. There is a substantial issue related to vendor lock-in, but on the other hand, the time-to-market is much faster with some standard prebuilt functionalities.

If the specific objective is to move quickly and in a well delimited set of consumption and distribution contexts, the use of packaging solutions in the CDN should be considered and could be a good approach.

Also, in the case that there is no packaging solution foreseen in your infrastructure, it is an alternative.

On the other hand, if the scope of the solution is quite heterogeneous and the requirements might have legacy needs or if there are many different environments, it will not be the best solution. All technology providers have a business strategy and may take internal decisions to discontinue functionalities that do not fit with your intended purpose.

DRM (Digital Rights Management) functionality is an important decision to make if there is a need to do it during packaging, and there will be a vendor lock-in associated to the solution design. The CDN provider must be integrated with the DRM provider, and this will bring major constraints on the decision to purchase the service and potential switching needs. Other features such as support for multi-audio and subtitles may vary between providers and may not even be available.

Working together with other CDNs in a Multi-CDN scenario may be hampered due to the possible need for standardisation of solutions and specific encoding requirements.

Another point that should be thought about is the traffic between the Origin and the CDN. Since the Origin is remote, it will make sense to think of effective storage-cache solutions in the CDN and efficient communication networks through peering or other alternatives.

Finally, on-the-fly packaging may be seen as a CDN functionality and not as the core business. Because of this, the future strategy of the CDN supplier may not be compatible with yours concerning the adoption of new functionalities and the discontinuation of others, since the effort of having a complete offer may be complicated for the supplier.

Summing up some pros, cons and points to keep in mind:

- **Pros**
  - Fast Time-to-market with standard features
  - No need for investment
  - Possible offload of packaging on current infrastructure
- **Cons**
  - Vendor lock-in on features, vision alignment and future provider changes
  - Constrains on other partners (i.e., DRM providers)
  - Multi-CDN solutions may become more complex or not viable

- **Keep in mind:**
  - Types of connection to the CDN and storage solution next to the CDN nodes
  - Types of packaging that will be needed (HLS, DASH, Smooth Streaming, etc.)
  - Types of content (video / audio)
  - Use of DRM and other features that will be required
  - Use of Multi-CDN
  - Existing encoding requirements of the assets

4.2 Peer-to-Peer (P2P)

Peer-to-Peer CDNs normally make use of WebRTC connections to access cached content on the end-user-devices. These distributed caches are federated in the network and content is exchanged directly between the playout devices. In most cases these are peer-assisted models, meaning there are also caches in the network that feed the distributed edges. This is essential to avoid playout interruptions when content is not yet available on the clients.

The advantages of this distribution model are the reduction of CDN costs, as popular content is quickly cached deep into the network without having to be played out by 'commercial caches'. From trials run by EBU Members this benefit manifests mainly with large live events in densely populated areas. Because traffic stays in the local loop of access ISPs, playout is not hindered when interconnection lines get congested or have reduced capacity for other reasons (e.g., limited interconnection capacity to specific ISPs).

This model does create upload traffic as the devices of audiences are used as caching sources for playout for other end-users. In general, ISPs are not in favour of these 'unpredicted' traffic flows in their network. From the content provider's perspective, P2P distribution models can be difficult to explain to their audiences as their devices and internet connection are used to upload traffic for other people's video playouts. Also, some content owners object to this model because their content is 'stored' on end-user equipment.

- **Pros**
  - Can improve playout quality with highly popular content in congested networks
  - Can reduce costs as not all content will be played out from 'commercial caches'
- **Cons**
  - Can create problems in ISP networks due to unpredictable upload traffic
  - Difficult to explain to audiences and content owners that end-user devices are used to store/cache content
  - Increases monitoring complexity

5. Pricing models and cost considerations

In this chapter we will describe the most applicable CDN pricing models and cost considerations. The last paragraph provides an overview on tender frameworks used within the EBU Membership to procure CDN services.

5.1 Pricing models

There are many possible pricing and commercial models available from different CDN providers. The options available to a customer will depend on a few factors that relate to their business:

- The amount of traffic they intend to distribute through a provider over the contract period. Typically, this is quoted in gigabyte, petabyte or exabyte.

- The length of contract a customer is prepared to commit to – 12 months is typical, however longer deals can be used to negotiate better rates.
- The amount of traffic (and therefore spend) a customer is prepared to commit to using over the course of a contract. Upfront commitments (typically quoted either in traffic volumes or as a spend-commit per month) can be used to secure better volume rates.
- Peak capacity requirements: expected BAU (business as usual) traffic peaks, and exceptional events e.g., live sport.

Unit Pricing

Typically, the rates quoted in CDN marketing literature (and websites) are their base prices for brand-new customers that are just starting out or that are unable or unwilling to make usage commitments. Customers with existing traffic can leverage historical data to forecast their usage and to negotiate better rates.

CDNs will typically quote and charge for the following units for media delivery. Not all providers will provide all services, but these are the ones usually most relevant to media streaming.

- A cost per gigabyte of traffic delivered
  - This can be a flat rate (e.g., $0.05/gigabyte) or
  - A 'waterfall' model of different delivery tiers, e.g.:
    - 0-10 petabyte/month: $0.05 / gigabyte
    - 10-50 petabyte/month: $0.04 / gigabyte
    - 50-100 petabyte/month: $0.035 / gigabyte
- Alternatively, but less common to cost per gigabyte of traffic delivered, CDNs also provide bandwidth arrangements paid in dollars/euros per Gigabit per seconds, through 95/5 arrangement.
- A cost per HTTP request – typically quoted in units of e.g., 10000. This could also be tiered as above.
- Provision, configuration and hosting of security TLS (Transport Layer Security) certificates. You can provide your own TLS certificate, which is usually cheaper than relying on the CDN to provide them but brings an additional management overhead. (The use of 'shared' TLS certificates, whilst often very cheap or free, is to be discouraged due to the security concerns inherent)
- Enhanced support provision – e.g., Premium / Enterprise support, dedicated customer engineer resources, enhanced monitoring services
- "Event Support" – typically a per-hour rate for enhanced hands-on operational support from skilled engineering / operational teams to ensure a high-traffic live streaming event goes as expected
- Professional services, bundled in hours – for advanced consultation & configuration that can't be self-serviced
- Origin Shield services – typically the introduction of additional caching tiers and/or Origin PoP mapping into customer configurations to reduce requests to content origins, and therefore cost.
- "Direct Connect" network interconnection services, to directly connect or peer the CDN's network with that of the customer, for the purposes of better CDN-to-Origin connectivity rather than what can be achieved over Internet transit or public peering.

There can be different rates for traffic delivery out of different global PoPs, particularly with the smaller CDNs. Traffic to North America/EMEA tends to be the cheapest, whereas APAC/Africa/South America is more expensive.

Not all services need to be taken, and additional value-add media services such as object/origin storage, media transcoding and packaging need to be quoted separately.

Commits

If there is a baseline of traffic or spend per month that a customer can commit to always using, this can often be used to secure better delivery rates. Traffic above the committed baseline is then usually charged as a monthly overage fee, as per an agreed rate card. Higher committed traffic levels achieve better overage rates.

This gives the customer the ability to balance flexibility and the financial risk around high peak traffic events or unexpectedly popular content going into overage. This is obviously much easier when there is existing traffic to forecast future consumption from. To begin with, a 'pay as you go' model with no commit is usually the most practical until reliable traffic forecasts can be made. Some CDNs allow (part) of the unused commitment into the next month(s) spend.

Alternative models

Recently, as well as traditional variable rate cards and commit levels for traffic, in the currently very competitive CDN market, some providers are interested in securing larger amounts (in the hundreds of petabyte/month range) of traffic for a fixed spend-per-month regardless of what is actually used. This 'all you can eat' model can be attractive in terms of having predictable rather than variable costs over a contract term, which can be particularly important for fixed-income PSMs. This is easier to manage in a single-CDN environment rather than in multi-CDN, depending on overall traffic levels.

5.2 Switching costs

There can be significant direct and indirect costs incurred as part of moving from using one CDN to another. Whether or not you operate in a single or multi-CDN environment, media streaming is complex and integrations between content providers and CDNs can be deep. On-boarding a new CDN into a mature media streaming environment is usually a project that requires dedicated resources on both sides.

Complexity of migrating media streaming configurations - ensuring aspects like access control/content protection, traffic routing, origin protection/request collapsing, etc. continue to work as intended

- Set-up costs – executing proofs of concept (could be pre- or post-contract signature), performing live trials and measuring the performance, ramping up the traffic, setting up and reviewing distribution analytics / log file delivery etc.

5.3 Tender options

Depending on a broadcaster's local legislative environment and funding model, different routes may be available for actual procurement and contracting of CDN supplier services. For example, due to the sums of money typically involved, PSMs in the EU may be obliged to run an OJEU (Official Journal of the European Union) tendering process and invite multiple providers to tender for their CDN business. Depending on the scope of the tender, CDN services are exempted from tender obligation within EU procurement frameworks. Running a tender is a complex process and can run to many months of effort before a contract can be awarded.

There are good experiences within the EBU Membership with a multi-year framework agreement, selecting 3 CDNs who enter a yearly price competition. The winner with the best total support / price gets the highest volume for that year, for example, divided in the ratios 50% : 25% : 25%. This approach reduces the tender overhead and can drastically reduce the Commercial/Public CDN cost.

6. Conclusion

With growing online audiences, all EBU Members will have to implement an online media distribution strategy to meet the demand. Defining the right strategy depends on the amount of traffic that needs to be served, available budgets for online distribution and availability of specialized staff.

To reach these objectives CDNs are crucial elements. There is a logical evolution to start from a single (public) CDN and grow into a multi-CDN approach that can start simply with a server-side decision logic based on static rules. This evolution may naturally lead to running your own Origin capability/servers that contain all published content, to grow independent of third party CDNs or to use static packaging that will unify the contribution feeds towards CDNs. Evolution from that stage can involve introducing more player-side fallback solutions and increasing complexity with dynamic switching rules and the possibility of benefiting from a private CDN setup.

The increase in complexity will involve more monitoring, larger tenders, and higher total costs. On the positive side it can reduce the cost per gigabyte delivered and provide a strategic advantage of more control of online delivery with higher network resilience, and finally as a result realise better audience experiences.

In deciding the right CDN architecture, it also is essential to know your audiences. To make a viability calculation you need to consider where your viewers are located and what kind of average/peak traffic they create. It is also important to identify who your main business partners are. All EBU Members deliver most of their traffic within their own country through 3-5 (Access) ISPs. Engaging with these partners to inform them on upcoming peaks and requirements will help to improve your online distribution and provide input to your CDN requirements list.

In general terms, when considering building your own (hybrid) CDN the following considerations are advised:

1. If you plan your own CDN from scratch, have at least one public CDN as a contingency solution.
2. Experience matters, hence specialized staff is essential
3. Keep focus on business and customer value
4. Find the most pragmatic trade-off between build or buy
5. Start small, if your own CDN brings value keep it and grow, otherwise better to switch it off.

7. Glossary

API	Application Programming Interface which exposes functionalities of online services for consumption.
Access ISP	An Internet Service Provider used by consumers to access to the internet from the home.
Caching server	An intermediate storage for files that are frequently requested.
CMAF	Common Media Application Format (CMAF) - An MPEG specification that provides a single encoding and packaging of segmented media objects for delivery and decoding on your devices in adaptive multimedia presentations

CMS	Content Management System, which is software to create and edit online content and metadata.
DDOS	Distributed Denial of Service, a cyberattack where a target server is overloaded by an excessive number of requests coming from multiple systems, usually one or more web servers.
DNS	Domain Name System, which converts human-friendly domain names to numerical IP addresses.
DOCSIS	Data Over Cable Service Interface Specification - a standard for high-speed data transport over legacy video coax cable networks to the home.
M3U	A computer file format or more specifically a manifest file for HLS streaming that enables creating a single-entry playlist file pointing to a stream on the Internet.
MPD	Media Presentation Description, which is a computer file format or more specifically a manifest file for MPEG DASH streaming that enables creating a single-entry playlist file pointing to a stream on the Internet.
PoP	Points of Presence are local caching servers in a CDN network that can be distributed throughout a region/world
QoE	Quality of Experience data, normally retrieved from online media player states (like start up time or buffer underrun), indicating the actual perceived performance of video playout.

8. Bibliography

Apple-HLS Redundant Streams in M3U	Apple Developer page describing the creation of Multivariant Playlists
Apple-HLS Specification	Developer page of Apple describing HLS specification and deployment scenarios
CMAF	Reference to the ISO-standard
DASH Industry Forum (DASH-IF)	Industry forum providing guidelines, test vectors and reference tools to implement MPEG DASH
DASH-IF, MPEG DASH multiple BaseURL support in MPD:	'Guidelines for Implementation: DASH-IF Interoperability Points'
DASH-IF Live Media Ingest Protocol:	Specification of the DASH-IF Live Media Ingest Protocol
MPEG-DASH:	Home page of the MPEG-DASH standard
Streaming Video Alliance Open Caching:	Project page