

CDN

CONTENT DISTRIBUTION

NETWORKS

Trabajo escrito de PI

Natalia Dai
Clàudia Giró Figueras

ÍNDICE

1. INTRODUCCIÓN	2
2. QUÉ ES UNA CDN	4
3. LATENCIA	5
4. EVOLUCIÓN	6
5. FUNCIONAMIENTO	8
5.1. DISTRIBUTION SYSTEM	9
5.2. REQUEST ROUTING SYSTEM	10
6. EJEMPLOS DE CDNs	13
6.1 AKAMAI	13
7. CONCLUSIÓN	16
8. REFERENCIAS	17

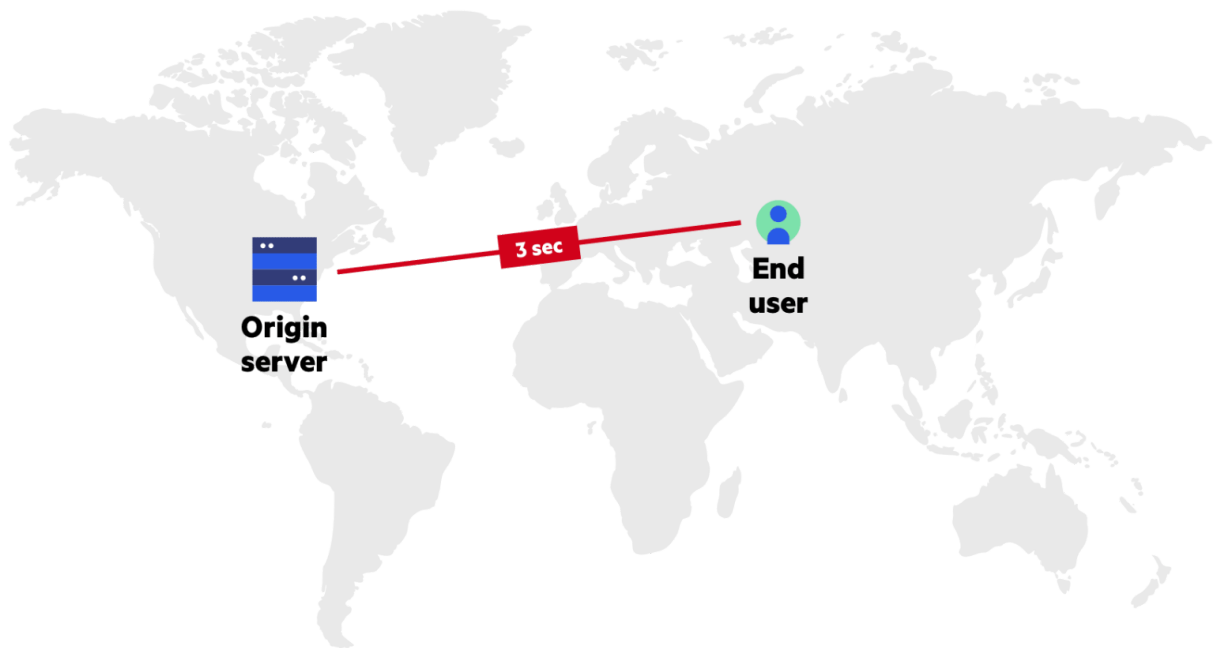
1. INTRODUCCIÓN

En el día de hoy la mayoría de negocios están en Internet, ya sea de manera total o parcial. Cada día pueden recibir centenares, miles o incluso millones de peticiones de usuarios que quieren acceder a los servicios que ofrecen. Desde la creación de Internet los tiempos de carga a los servicios web han mejorado de una forma muy drástica, y seguirán mejorando de forma constante con los años. De esta forma conseguimos navegar por Internet en cuestión de microsegundos.

Estos resultados tan pequeños nos podrían hacer pensar que todos los usuarios están satisfechos, ya que podemos acceder extremadamente rápido a donde queremos, pero la realidad es que esto no sucede. Hay estudios que muestran que mientras menos latencia haya, más satisfecho está el usuario y que por tanto más conexiones habrá: impacta directamente con la rentabilidad de las empresas.

Se han inventado muchos mecanismos para poder reducir la latencia y así dar una mejor experiencia a los usuarios. Uno de estos fueron los CDN. Se basan en replicar el contenido a distintos servidores distribuidos por todo el globo y redirigir las peticiones entrantes. Se usan algoritmos para poder detectar de dónde vienen esas peticiones y se mandan al servidor más cercano geográficamente.

Without CDN



With CDN

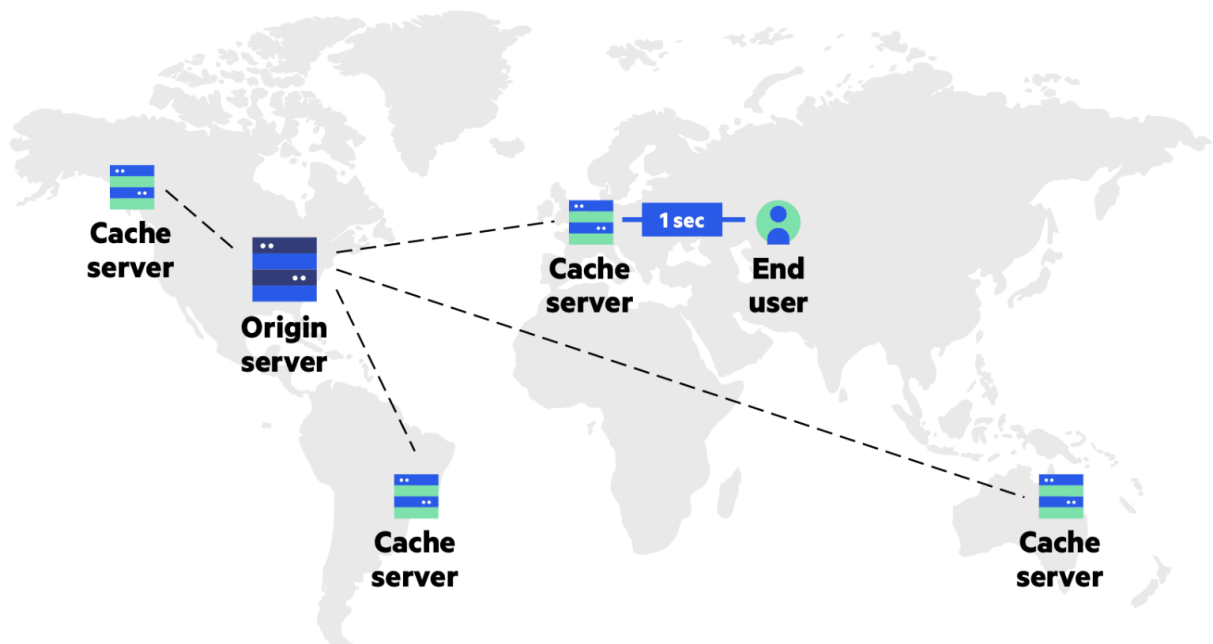


Figura 1 - La utilidad de tener CDN

2. QUÉ ES UNA CDN

CDN son las siglas de *Content Delivery Network*. Está formada por un grupo de servidores distribuidos geográficamente que facilitarán y agilizarán la entrega de contenido de internet. Permite la transferencia rápida de la información necesaria para cargar el contenido de Internet: HTML, archivos Javascript, hojas de estilo, imágenes y vídeo. Los centros de datos de todo el mundo utilizan cachés para almacenar allí copias de los archivos para que se pueda acceder a ellos de forma más rápida a través de un servidor cercano. Además, una CDN configurada de forma correcta también puede ayudar con la protección de sitios web contra ataques maliciosos, como DDoS.

Hay que tener en cuenta que una CDN no es igual a un servidor de sitios web. Se dedica a almacenar en caché el contenido en el perímetro de la red, mejorando el rendimiento y la disponibilidad de la página web. Reducen el ancho de banda del servidor, previenen interrupciones en el servicio y mejoran la seguridad.

Se podrían resumir los beneficios de utilizar una CDN en la siguiente lista:

1. Mejorar los tiempos de carga de un sitio web.
2. Reducir los costes de ancho de banda.
3. Aumentar la disponibilidad del contenido y la redundancia.
4. Aumentar la seguridad del sitio web.

3. LATENCIA

La finalidad principal de una CDN es reducir la latencia, y se puede disminuir el tiempo de carga de distintas formas:

- Una de ellas es reducir la distancia entre el usuario y el servidor donde pide los datos. Para eso se colocan distintos *Points of Presence* (PoP) en múltiples localizaciones geográficas que contienen un número de servidores usados como cachés responsables de entregar contenido a los visitantes más próximos.
- Otra sería hacer optimizaciones de software y de hardware, que con un equilibrio de carga eficiente y un uso de dispositivos de estado sólido (SSD), puede mejorar el rendimiento de la infraestructura del lado del servidor, haciendo que los datos lleguen más rápidos al usuario.
- También se hacen reducciones de transferencias de datos, empleando técnicas como la minificación (proceso que consiste en eliminar los caracteres innecesarios que compone un bloque de código) y la compresión de archivos (por ejemplo con GZip, que es un método que muchos proveedores de CDN tienen predeterminados). Con esta ayuda, la cantidad total de datos transferidos disminuye, reduciendo también la latencia y el ancho de banda necesario. Todo ello contribuye a la aceleración de las cargas iniciales de la página.

```
1 // This function takes in name as a parameter
2 // and logs a string which greets that name
3 // using the information passed
4 function sayHi(name) {
5     console.log("Hi" + name + ", nice to meet you.")
6 }
7
8 sayHi("Sam");
```

```
1 function sayHi(o){console.log ("Hi"+o+", nice to meet you.") }sayHi("Sam");
```

Figuras 2, 3 - Ejemplo de minificación (antes y después de su aplicación)

4. EVOLUCIÓN

Los CDN comerciales han existido desde los años 90. Como cualquier otra tecnología con décadas de antigüedad, pasaron por múltiples etapas evolutivas, con la ayuda de avances en la conectividad (como la fibra óptica) y nuevas tendencias en el consumo de contenido.

Por lo general, se podría considerar que la evolución de las CDNs se pueden dividir en 3 generaciones, cada una con nuevas características y tecnologías:

- CDN estático (1998)

El contenido servido es el estático, que significa que el tipo de información que es igual para cada usuario, aquel material que no suele variar en el tiempo y no se tiene en cuenta los datos personales del usuario. Para este caso, replicar los servidores no es ningún problema ya que sabemos que estos recursos no cambian. Por otro lado, como primera generación de CDNs, se centraba principalmente en el rendimiento y era muy caro, por lo que lo solía usar solamente el sector corporativo.

- CDN dinámico (2001)

En este caso, el contenido servido es tanto estático como dinámico, siendo dinámico el tipo de información que se personaliza para cada usuario basada en sus datos, que puede variar en función de parámetros como el nombre del usuario, la hora o preferencias fijadas por el usuario, gracias a la recopilación de información legítima de los sitios web (o ilegítima que se vende en mercados negros). Con la muestra de esta información, tenemos de resultado unos datos más útiles y más pensados para nosotros. El inconveniente para los CDNs en este caso es que, como no sabemos quiénes querrán el contenido adicional, no sabemos dónde replicarlos.

Como no queremos que sea ineficiente, tenemos que encontrar un balance entre replicar el contenido dinámico en servidores y tener suficientes para evitar cuellos de botella que generan ineficiencia. Una de las soluciones sería usar servidores HTTP con tecnología *push*, que permiten enviar datos desde un servidor Web a un navegador Web anticipándose a la solicitud del cliente.

Por otro lado, como segunda generación de CDNs, se centraba principalmente en el rendimiento y la disponibilidad, se había bajado un poco el precio, y lo solía usar el sector empresarial.

- CDN multipropósito (2010)

En esta tercera generación de CDNs se siguen sirviendo contenidos estáticos y dinámicos. Además de centrarse principalmente en el rendimiento y la disponibilidad, también se enfoca a la seguridad. En este caso, el precio ya es mucho más asequible, por lo que lo puede usar cualquier persona con una página Web.

5. FUNCIONAMIENTO

Por lo general las CDN's siguen el esquema mostrado a continuación. Son 7 componentes, pero se pueden añadir más para obtener las funcionalidades deseadas.

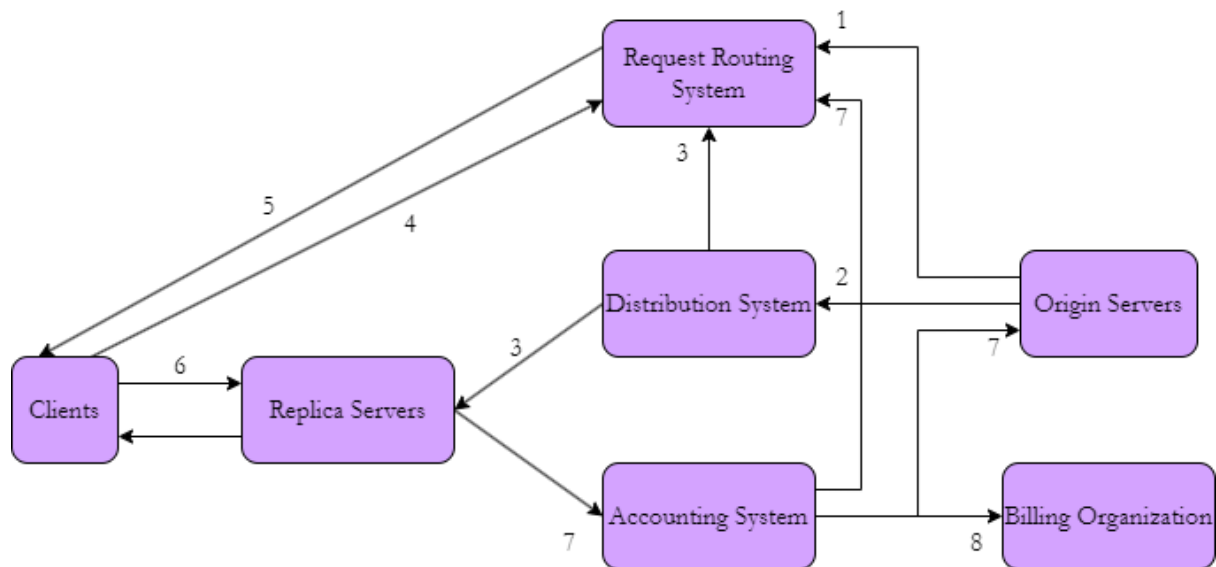


Figura 4 - Esquema del funcionamiento de una CDN

El funcionamiento es el siguiente:

1. El *Origin Server* contrata un *Distribution System* y delega el contenido que desea y sus *Uniform Resource Identifier* (URI) a la empresa.
2. La CDN distribuye los contenidos que le ha dado el *Origin Server* por la red en los servidores que tiene distribuidos alrededor del planeta (*Replica Server*). Determina las rutas que se deberán usar para peticiones futuras.
3. Gracias a los URI cedidos a la CDN y a la delegación (1), ahora cuando un end user quiera acceder a un contenido, esta petición irá al *Request Routing System*, que con DNS y otros algoritmos se redirigirá a un *Replica Server*.
4. Una vez llegado al *Replica Server*, este le proporciona el contenido exigido y guarda los datos de la conexión posteriormente en el *Accounting System*.

5. Ambos *Accounting System* y *Billing Organization* son los encargados de recopilar datos para poder servirle al cliente como servicios de valor añadido.

5.1. **DISTRIBUTION SYSTEM**

Es muy importante decidir bien cuál será la ubicación de los servidores réplica. Se debe tener en cuenta cuántas réplicas vamos a tener y dónde las queremos colocar, pues queremos conseguir reducir la latencia en la mayor medida posible.

Para poder distribuir el contenido en los distintos servidores réplica se usan principalmente dos métodos: por Internet o por *broadcast* por satélite. El primero hace que el resultado sea impredecible, mientras que el segundo otorga más calidad y eficiencia pero es mucho más complejo de implementar.

Los problemas generados por la localización de los CDNs se estudiaron a nivel teórico y se diseñaron unos algoritmos subóptimos que tienen en cuenta todas las variables que influyen en el momento de decidir cómo diseñar uno de estos servidores. Algunos de estos algoritmos son:

- k-HST
- Minimun K-center
- Heuristic Solutions
- Greedy Algorithm
- Topology-informed Placement Strategy

5.2. REQUEST ROUTING SYSTEM

Las CDNs tienen que tener un sistema que nos indique qué servidor escoger por cada petición que se recibe. El aumento en el tiempo de respuesta está directamente relacionado con la pérdida de rendimiento, la congestión y una gran cantidad de usuarios que recargan el sitio web, lo que empeora el acceso al sitio web. Según un estudio realizado por Brian D. Davison, profesor y presidente del Departamento de Informática e Ingeniería en la Universidad de Lehigh, si el tiempo de respuesta de una solicitud web supera los 8 segundos, alrededor del 30% de los usuarios abandonan la solicitud.

En general, el criterio que se sigue es seleccionar el CDN más cercano geográficamente usando RTTs o *Hop-Count* (cantidad total de dispositivos intermedios entre el origen y el destino por donde pasan los datos). Sin embargo, puede no ser lo mejor para dirigir la solicitud del cliente con el fin de mantener una alta QoS ya que al desconocer el estado de la red puede resultar ineficiente.

En su lugar, se podría considerar un conjunto de parámetros durante este proceso de selección, como la distancia, la velocidad, el ancho de banda disponible y la carga del servidor sustituto. Una vez conocida toda esta información, ya se podría redirigir al cliente al servidor óptimo.

Algunos de los métodos que se usan para el *Request Routing System* teniendo en cuenta lo explicado anteriormente son:

- *DNS-based*

Siendo el mecanismo más aplicado, se retorna mediante DNS la dirección IP del servidor que ha sido seleccionado como la mejor opción una vez procesada la solicitud o petición. Ofrece transparencia, simplicidad y flexibilidad.

Sin embargo, con el uso cada vez mayor de DNS remoto, las CDN que usan este tipo de sistemas enfrentan un gran desafío en cuanto a la degradación del rendimiento.

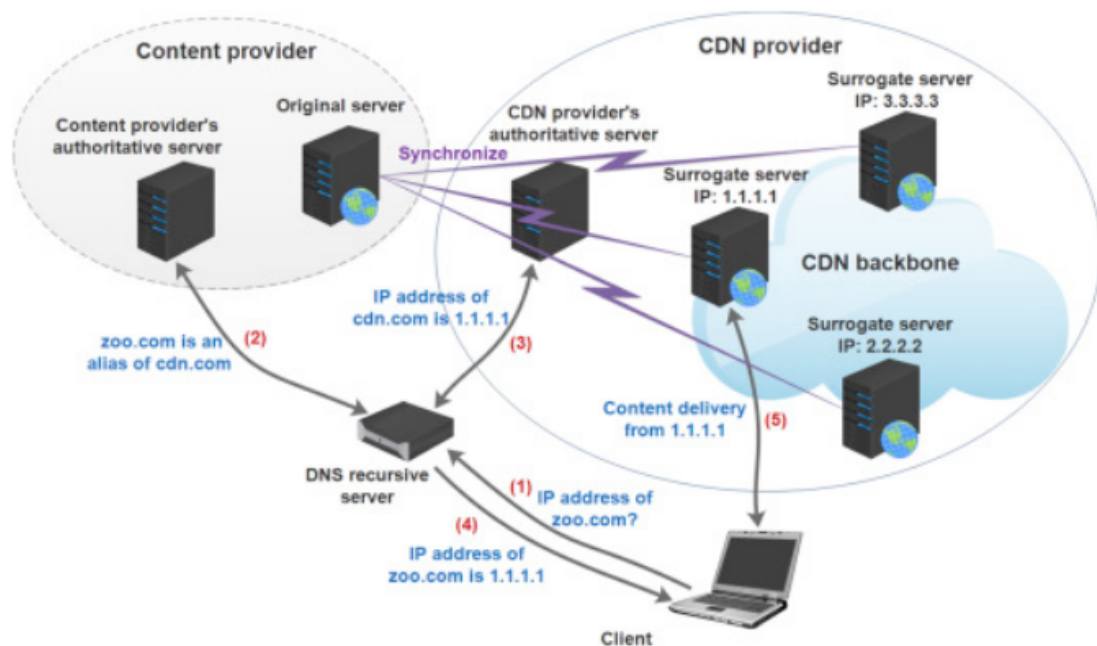


Figure 5 - Ejemplo del funcionamiento de DNS en CDN

- *HTTP redirection*

Este mecanismo permite que un servidor Web propague el resultado de la selección del servidor al usuario a través de encabezados HTTP, así para que el cliente pueda ser redirigido al servidor óptimo después de que la petición haya viajado hasta el servidor origen. Esto nos puede generar un cuello de botella y

por tanto, tener ineficiencia, además de que en cada sesión HTTP se produce un retraso de ida y vuelta (más otros gastos generales).

Por otra parte, una ventaja que presenta es la capacidad de gestionar las redirecciones con una granularidad muy baja, por lo que se puede redirigir a páginas Web y no a la Web entera, que es lo que ocurre en otros métodos.

- Anycasting

En este método, la misma dirección IP es asignada a un grupo de servidores sustitutos ubicados distributivamente. Cuando el cliente envía solicitudes a la dirección IP, mediante *anycast*, estas peticiones se enrutarán a todos los servidores, de los cuales uno le servirá la solicitud, que seguramente sea el servidor sustituto más cercano definido por la política de enrutamiento.

Ofrece transparencia, tanto para los proveedores de contenido como para los usuarios finales. Por otra parte, las fluctuaciones de enrutamiento de Internet pueden afectar negativamente la estabilidad de IP *anycast*.

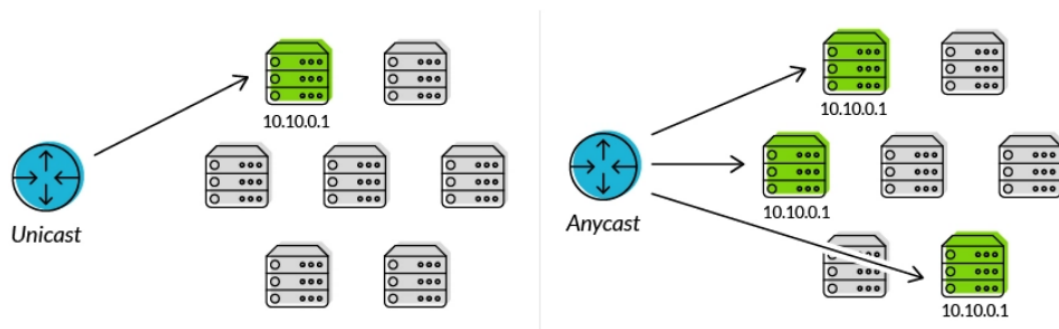


Figura 6 - Comparación *Unicast* y *Anycast*

6. EJEMPLOS DE CDNs

Existen muchas empresas que ofrecen el servicio de CDNs, y las más conocidas o populares son:

- Akamai
- CloudFlare
- Amazon CloudFront (AWS)
- Microsoft Azure
- BelugaCDN
- KeyCDN
- CMS Hub
- Limelight
- Swarmify
- ...

6.1 AKAMAI

Akamai, una de las listadas en el apartado anterior, además de ser una de las CDNs más importantes del mercado fue la primera en inventarse. Fue fundada en el año 1998 por el estudiante graduado del MIT, Daniel Lewin, ayudado por el profesor de Matemáticas Aplicadas de la misma institución, Tom Leighton, dándole a esta corporación exitosa un nombre de origen hawaiano que significa “inteligente”.

Algunas multinacionales que tiene como clientes son Google, Apple y Facebook, entre otros, que confían en las soluciones y la experiencia de Akamai de proteger su información.

Tiene más de 1400 redes y 365.000 servidores que abarcan 135 países, con una capacidad de red global de más de 300 Tbps. Con estos aspectos, son capaces de evitar atascos con agilidad y de ofrecer experiencias digitales increíbles a cada usuario, independientemente de la ubicación geográfica, el dispositivo o la red.

¿CÓMO FUNCIONA AKAMAI?

Su función básica es la entrega de información de un sitio web o aplicación mediante Akamai Edge Network (la CDN) para que así Akamai pueda “hostear” el contenido.

Para atender el tráfico en la red de Akamai se necesitan al menos tres componentes:

- Un archivo ARL (Akamai Resource Locators) o de configuración para indicar a los servidores cómo responder a las solicitudes. Este se consigue mediante Akamaizer, una herramienta propia de Akamai, que convierte las URLs/URIs de los clientes en ARLs, para lograr la distribución en la red de Akamai.

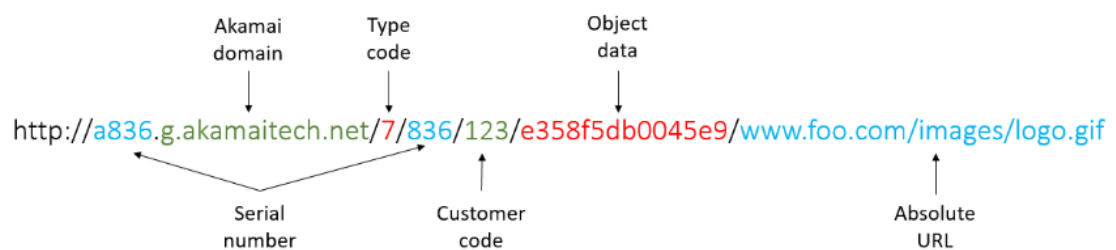


Figura x - Ejemplo de ARL

Los campos que presenta tiene las siguientes funcionalidades:

- Serial number: aparece dos veces en el ARL y sirve para equilibrar la distribución de contenido en la red de Akamai.

- Akamai domain: Akamai tiene creada su propia red de DNS, que con este campo y otros métodos, se elige un *Replica Server* encargado de servir la información al usuario y de resolver el nombre con la dirección IP del servidor determinado. También se tiene en cuenta la IP origen de la petición DNS, el estado de la red y de los servidores.
Puede ser con HTTP (contenido no seguro, a<serial#>.g.akamai.net) o HTTPS (contenido seguro, a248.e.akamai.net).
- Type code: indica el tipo de mecanismo de coherencia a utilizar para cada objeto. Suele ser '7', que indica que el campo de Object data contiene un ID de objeto (en este caso del gif), o 'f', que indica que contiene un valor TTL.
- Customer code: identifica el proveedor de contenido.
- Object data: sirve como un mecanismo de coherencia del objeto. En caso de que el objeto se modifique, se actualiza el ARL.
- Absolute URL: Se especifica el origen de la URL del objeto que quiere acceder el cliente. Luego se copia el contenido en los *Replica Server*.
- Un código de proveedor de contenido (código CP) para etiquetar nuestro contenido como nuestro en sus servidores.
- Un nombre de host edge para que sus servidores puedan actuar como proveedores de contenido de nuestro sitio web.

Además de necesitar más tipos de certificados edge (para por ejemplo asegurar el tráfico), el funcionamiento de la CDN es el generalmente descrito del apartado 4 con el método DNS (apartado 5.2).

7. CONCLUSIÓN

Con todas las características y ventajas (tanto para los distribuidores de contenido como para nosotros, los usuarios finales) que presentan las CDNs después de un estudio sobre estas, se podría decir que son imprescindibles en nuestro día a día, ya que se hace un uso (incremental) del Internet cada día, y eso significa a tener toda una CDN detrás distribuyendo servidores por todo el globo.

Desde la creación de las CDNs se ha visto una mejora muy grande respecto a el tiempo de respuesta de una página web como de la satisfacción de los usuarios con la descarga de contenido. Fue un invento que revolucionó la forma en la que funcionaba Internet y a día de hoy sigue dando solo ventajas.

Con todo el desarrollo de tecnologías que está ocurriendo actualmente es obvio pensar que habrá más mejoras y más generaciones de CDNs por venir, y con ello, un mejor Internet para todos. Siempre se va a buscar dar la mejor calidad al usuario final.

8. REFERENCIAS

What is a CDN? | Cloudflare

<https://www.cloudflare.com/es-es/learning/cdn/what-is-a-cdn/>

What is a CDN? How does a CDN Work? | Imperva. (n.d.). Learning Center.

<https://www.imperva.com/learn/performance/what-is-cdn-how-it-works/>

¿Qué es un CDN y qué plataformas utilizar en tu proyecto? (2020, September 13).

<https://josefacchin.com/que-es-un-cdn/>

Performance | Cloudflare

<https://www.cloudflare.com/es-es/learning/cdn/performance/>

Akamai. (2022, September 2). Wikipedia.

<https://es.wikipedia.org/wiki/Akamai>

¿Qué es una CDN? (2021). Akamai.com.

<https://www.akamai.com/es/our-thinking/cdn/what-is-a-cdn>

How Akamai works. (2022). Get Started.

<https://techdocs.akamai.com/start/docs/how-akamai-works>

ARL syntax. (2022). Edge Diagnostics.

<https://techdocs.akamai.com/edge-diagnostics/docs/arl-syntax>

Omotunde, Ayokunle & Izang, Aaron. (2015). Request-Routing for Content Delivery Networks (CDN). International Journal of Advanced Research in Computer Science. 6. 10-16.

https://www.researchgate.net/publication/313797759_Request-Routing_for_Content_Delivery_Networks_CDN

Sarkar, Darothi & Rakesh, Nitin & Mishra, Krishn. (2015). Problems in Replica Server Placement (RSP) over Content Delivery Networks (CDN). 10.1145/2818567.2818600.

https://www.researchgate.net/publication/315783203_Problems_in_Replica_Server_Placement_RSP_over_Content_Delivery_Networks_CDN

Sandeep Kath. (2015, January 6). *Request routing in CDN*. <https://www.slideshare.net/sankath/request-routing-in-cdn>

Wang, Z., Huang, J., & Rose, S. (2018). Evolution and challenges of DNS-based CDNs. *Digital Communications and Networks*, 4(4), 235–243. <https://doi.org/10.1016/j.dcan.2017.07.005>

de, C. (2009, April 6). *tecnología de envío de mensajes del servidor al cliente*. Wikipedia.org; Wikimedia Foundation, Inc. https://es.wikipedia.org/wiki/Tecnolog%C3%AD_a_push